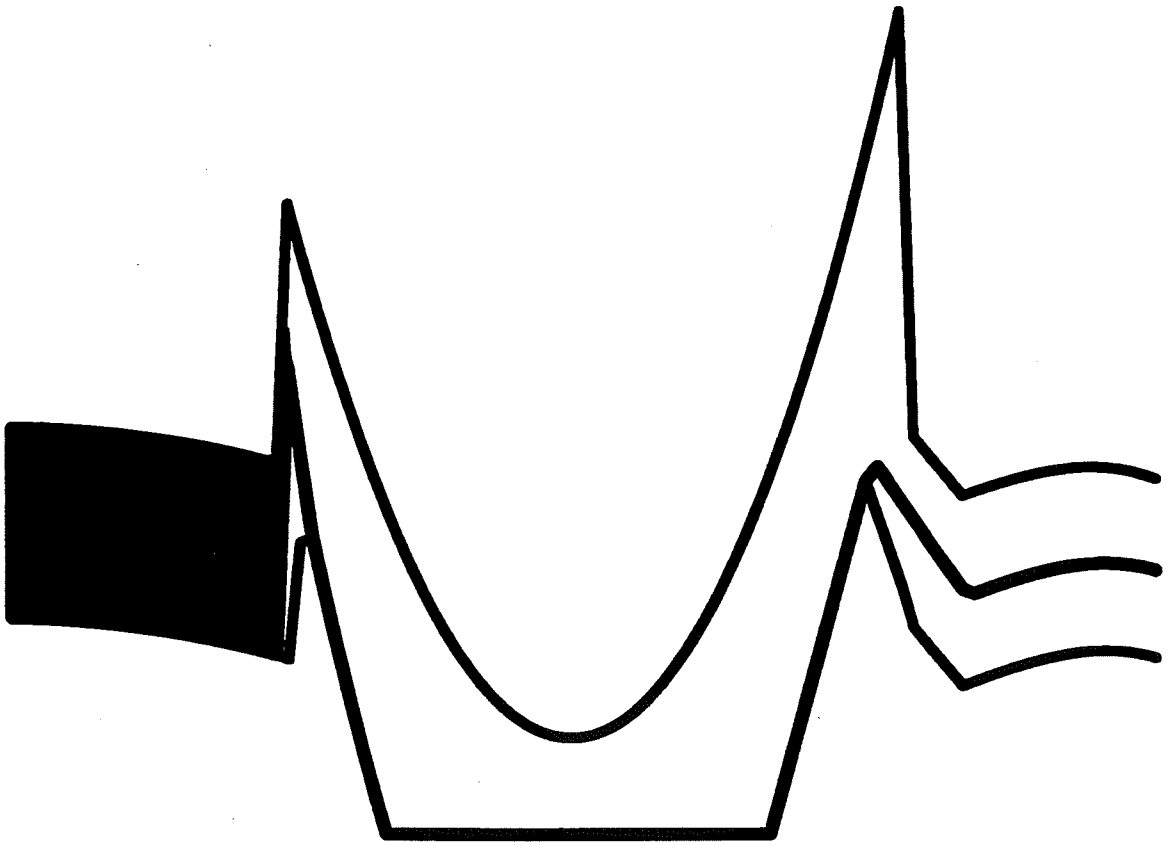
Discrete Time Optimal Control

Hans F. Ravn



1999

Denne afhandling er af Danmarks Tekniske Universitet antaget til forsvar for den tekniske doktorgrad.

Antagelsen er sket efter bedømmelse af foreliggende afhandling.

Lyngby, den 9. juni 1999
Hans Peter Jensen
Rektor

/Anne Grete Holmsgaard
Universitetsdirektør

This thesis has been accepted by the Technical University of Denmark for public defence in fulfilment of the requirements for the degree of Doctor Technices.

The acceptance is based on this thesis.

Lyngby, June 9, 1999

Hans Peter Jensen
Rector

/Anne Grete Holmsgaard
University Director

Foreword

The thesis presented here is a result of work that has mainly taken place during my activities up to 1994 at the Institute of Mathematical Modelling, previously the Institute of Mathematical Statistics and Operations Research, at the Technical University of Denmark. The publication synthesizes part of my work with mathematical modeling and optimization, where in particular the discrete time optimal control problem formulation was found appropriate and challenging in many applications.

I owe much to Dr. Tech. R.V.V.Vidal at the Institute of Mathematical Modelling for introduction to the subject and for continuous inspiration and friendship over the years. In similar ways, Dr. Tech. Z.Nahorski at the Systems Research Institute of the Polish Academy of Sciences has been with me for long times.

Also I am grateful to other good colleagues that contributed towards my understanding of the subject, in particular Mogens Pedersen, Claus Stefan Nielsen, José Ferreira, Ólafur P. Pálsson, Jens Moberg Rygaard, Claus Jørgensen, Pedro Borges and Halldór Pálsson.

Introduction

Experience has shown that a number of relevant problems within engineering, economics and energy systems analysis may be formulated and modeled as discrete time optimal control problems, or more generally as mathematical optimization problems.

Mathematical optimization models serve several purposes. Sometimes they are used to gain insight into the nature of matter, as is for example the case with the modeling of various physical phenomena as optimization problems (reflected as e.g. least force principles). Sometimes the purpose is to give a quantitative answer or support to a decision problem that is well understood in qualitative terms. Very often the purpose is somewhere between these: both to gain insight into the nature of a problem and to contribute towards quantification and decision making.

In this light the purpose is to clarify a problem by modeling, and in relation to optimization this is done by characterizing an optimal solution, by interpreting it and by finding it. As indicated, it may very well be relevant to focus on one or two of these aspects only.

The relevance of treating discrete time optimal control problems specifically is due to the fact that this formulation seems natural for a number of practical problems; and the more adequately the problem is formulated, the better may the other purposes of modeling and optimization be fulfilled.

The distinctive element in the formulation of the discrete time optimal control problem (OCP) - relative to the general mathematical programming formulation - is the stagewise perspective. The OCP is seen as consisting of a sequence of stages, where each stage is linked only to its two neighbor stages. Typically, all stages display the same structure. Therefore the relationships between the stages are central in the analysis. The possibility of choice between different paths, or states, through the stages is represented by the control or decision variables, also given stagewise.

The OCP was identified as a separate topic around 1960. At that time, continuous time optimal control was emerging as a modern version of the calculus of variations. The maximum principle of Pontryagin (1962) was formulated and proved powerful for theoretical and practical purposes, and attempts to formulate similar optimality conditions for the OCP were made, without full understanding of the intricate differences between continuous and discrete time problems. Further, Bellman's (1957) principle of optimality was formulated and applied in dynamic programming algorithms for solving the OCP.

The new field of mathematical programming emerged simultaneously and soon the similarities in the principles of modeling real systems as optimization problems became apparent. Around 1970 attempts were made to unify the discrete time optimal control theory and mathematical programming. In this, the OCP was seen as a special case of a mathematical programming problem, and the particular structure of the OCP was interpreted as a case of sparcity. With the advances of mathematical programming the interest shifted to some extent away from the specificity of the OCP.

However, the stagewise structure of the OCP permits specific characterizations of the solutions in terms of necessary and sufficient optimality conditions and interpretations of the solution. The classical maximum principle, for instance, is formulated as a mixture of conditions of stationarity and optimality, which is not natural for the general mathematical programming problem. Further, the distinction between state and control variables in the OCP is obvious in relation to the real system being modeled, and the so called costate vector, which is essential in the formulation of the maximum principle, has its natural interpretations. Thus, for instance, the state vector, which often represents physical or technical variables, may also be interpreted as representing the memory in the system, thus expressing that the stages are not only similar elements, but also represent a

specific sequence unilaterally forwards through the stages. Similarly, the costate vector may be interpreted as a transfer price between consecutive stages, and in particular also as a price signal transmitted backwards through the stages. These interpretations in turn admit construction of transparent solution algorithms.

Another strong reason to insist on the OCP formulation is that the discrete time optimal control problem in many cases is formulated as a model of real system that is inherently developing in continuous time. Two words are significant here. First, the continuous versus discrete time modeling is not merely a mathematical topic but is related also to e.g. measurement and control, which in turn have both theoretical and practical aspect. Second, when the parameter or index in the control formulation represents real time this has important implications, in particular for the information structure. Thus, if the model contains stochastic elements the uncertainty may be eliminated gradually over time, and this should be reflected in the control strategy.

The approach that interprets the OCP as a specific case of the more general mathematical programming problem has its obvious strengths, permitting immediate transfer of result from the general to the specific. However, it is obvious that it also has weaknesses, besides the control theoretic aspect just mentioned. Precisely because the OCP has a particular structure, stronger results may be derived, and even types of results that need not be applicable to the general mathematical programming problem. In this perspective the way forward is to exploit that the OCP in fact is a special case of the mathematical programming formulation, recognizing that the OCP is specific and that therefore further results and perspectives hold for this.

That this is not straightforward is illustrated by the fact that even within the field of discrete time optimal control itself a thorough understanding of the relations between the two prototype approaches, the maximum principle and dynamic programming, is lacking. Although they are both specific to the OCP and both apply a stagewise approach they are seldomly connected, neither theoretically nor in applications. Notable early exceptions are Krotov (1967) and Dreyfus (1976), the former largely neglected in the West. Attempts to highlight and analyze the interplay between the two traditions of mathematical programming and control theory were made in Luenberger (1972), Boltyanskii (1978), Ferreira and Vidal (1986), Nahorski and Ravn (1988).

The thesis

In the present thesis we continue this ambition of benefiting from the mathematical programming advances for the OCP, and at the same time recognizing the specificity of the OCP, with the ultimate aim of improving operations research application of the OCP.

The analysis naturally falls into two categories, that of characterizing an optimal solution and that of finding it.

Essential elements in the characterization is to derive the necessary and sufficient optimality conditions. In relation to solution structure this in particular means the analysis of stability and sensitivity of the OCP. Also, classical distinctions within the field of mathematical programming, such as e.g. between primal and dual decomposition, are relevant in this context. An equally important aspect of the characterization of the optimal solution is the interpretation of it. Although this will be specific to the problem and to the circumstances in which the problem and the analysis are situated, general characteristics like shadow prices may be derived.

The task of finding the optimal solution is derivation, implementation and application of algorithms. An essential element behind any algorithm is the description of the properties of points, that the algorithm will accept as solution. As this is typically necessary and/or sufficient optimality conditions, various algorithms may be derived, differing between them according to which

optimality conditions they are based on. Another essential element in the analysis of algorithms, to be dealt with here, is derivation of global as well as local convergence properties.

While methods and results of mathematical programming will be exploited, the approach taken will be to insist on the stagewise perspective in the analysis such that in this way the specificity of the OCP is best exploited and illustrated.

The thesis has the form of a book in order to present a synthesis on this large, diverse and scattered subject. Since previous attempts of comprehensive treatments appeared - see in particular Canon, Cullum and Polak (1970), Polak (1971), Tabak and Kuo (1971), Boltyanskii (1978) - significant developments have taken place in mathematical programming, numerical analysis, computing theory and practice and in the acceptance of mathematical modeling in relation to real world problem solving. It is therefore appropriate to present a coherent framework for the exposition and development of discrete time optimal control. Within this framework the relevant perspective may be unfolded and related and particular results may be suitably appreciated.

The thesis relies on results and traditions of a number of scientific fields, in particular mathematical programming, control theory and numerical analysis. Although related, these fields have separate traditions for appreciation and presentation of results and therefore a suitable balance must be obtained. The selection and presentation of the material here has been done with the aim of ultimately providing tools for operations research applications.

While a main perspective in the thesis is applications this aspect is not heavily represented, in fact only the first chapter introduces application examples. The reason is not that the application perspective is not taken seriously; the reason is, to the contrary, that applications are considered so important that they cannot be properly incorporated if the present volume should not be excessively large. Rather, the reader is referred to the survey of engineering application of discrete time optimal control in Ravn and Vidal (1990), and more specifically to the author's applications oriented publications, see Chapter 11.

The thesis consists of ten chapters. In Chapter 1 the discrete time optimal control problem (OCP) is introduced. We give examples of application, analysis and solution, and we review approaches towards the analysis of the OCP. In particular we review the development of the discrete time maximum principle and relate it to dynamic programming. We review the mathematical programming problem approach towards the OCP. The purpose here is to see the similarities and differences in the optimality conditions in the approaches and to point out parallels in their historical development. In particular it is seen that it is very easy to formulate optimality conditions in the form of a number of maximum principles. The chapter also serves the purpose of presenting main concepts from mathematical programming that will be applied to the OCP in later chapters.

Chapter 2 defines and deals with the upper boundaries, in mathematical programming known as optimal value functions or perturbation functions, in relation to dynamic programming known as e.g. cost-to-go functions. Conditions for smoothness, concavity and other properties of the upper boundaries are derived. The basis for this treatment is the development of results on sensitivity of the mathematical programming problem, which are here transferred to the OCP. The main topic of the chapter is the clarification of the possibility of applying a stagewise approach towards the analysis and solution of the OCP. Central to this is the question of stability and sensitivity. It is shown that in general the stagewise approach requires stronger assumptions than a mathematical programming approach does. In particular this may be seen in the conditions of constraint qualifications, which need not hold stagewise, even if they hold for the OCP, considered as a mathematical programming problem.

In Chapter 3 it is the demonstrated that a number of optimality conditions may be presented and interpreted in relation to the upper boundaries. It follows that sufficient conditions guaran-

teeing that they hold may be conveniently expressed in relation to the upper boundaries, as these are defined and analyzed in Chapter 2. Also, the relationships between the different optimality conditions are more clearly seen; in particular, maximum principles and dynamic programming conditions may be related. Once this is clear it is straightforward to formulate versions of the maximum principle with nonlinear or nonsmooth auxiliary (price) functions. In relation to mathematical programming decomposition this corresponds to decomposition by nonlinear price functions, and it is further shown how this fits into a duality framework based on stagewise decomposition and nonlinear price functions.

In Chapter 4 the analysis of algorithms is initiated. Specifically, this chapter deals with dynamic programming. The emphasis of the chapter is on situations where the upper boundaries may be represented explicitly; this in part is motivated by the fact that the popular approach of discretization has poor approximation features. The major case where the upper boundaries may be explicitly represented is the quadratic-linear problem. We present classical results about this for the problem with equality constraints. We also treat the problem with inequality constraints; it is shown that it is possible to apply dynamic programming also in this case provided the state vector is of dimension 1.

Also Chapter 5 deals with application of the upper boundaries in algorithms, now the smaller ones. This may be seen as resource (or primal) decomposition with respect to stages. Rate of convergence result is derived in the case of general non-quadratic/linear problems. The approach may be attractive if parallel computations can be applied. It is shown that with non-smooth upper boundaries (present if e.g. the solution or the Lagrange multipliers are not unique, as is in general the case with inequality constraints present) this approach will hardly be viable.

Chapter 6 deals with maximum principle algorithms. Although the tradition for formulating and applying algorithms based on the maximum principle is long, few theoretical results have appeared. The chapter unites the application of the classical and the generalized maximum principles in the sense that the generalized maximum principle may be seen as a way of controlling the stepsize in a classical maximum principle algorithm. Further, it is shown that the concept of feedback strategy emerges naturally as a response to the difficulties encountered in case of state dependent constraints. The generalized maximum principle is derived and it is shown that convergence is intimately linked to approximation of the upper boundaries. A weakness of these maximum principle algorithms is that they achieve at most a linear rate of convergence.

Chapter 7 deals with algorithms that traditionally took inspiration from dynamic programming, exploiting smoothness of the problem functions. It is shown that this may alternatively be seen as a generalized maximum principle idea. Conditions for global and local convergence are analyzed. Global convergence is based on an active set strategy applied within a dynamic programming solution of a quadratic-linear problem, and it is also based on the absolute value penalty function. Local convergence results are based on two different applications of Newton iterations, and on the differential dynamic programming (DDP) approach. They are all based on recursive formulation and solution of quadratic-linear approximations, and they all provide quadratic rate of local convergence. The implementations differ between the methods, in particular with respect to the possibility of maintaining feasibility throughout the iterations with respect to nonlinear constraints.

Chapter 8 deals with algorithms based on price, or dual, decomposition with respect to stages. Lagrangian relaxation is the classical application of this, however it is shown that it is also possible to derive convergence results with the application of nonlinear price functions. In particular it is shown that when a weighting of given price functions is applied then the dual function is convex with respect to the weights. This permits derivation of subgradient and gradient algorithms. Also

Lagrangian relaxation, i.e. application of linear price functions, is treated, and it is shown how the complicated situations with linearity in the state variables (implying non-unique solutions to the subproblems and consequently a nonsmooth dual function) may be treated.

Chapter 9 treats forwards algorithms. The idea here is to parameterize optimal trajectories emanating from the initial point, and then select the parameter values such that a feasible - and hence optimal - solution is found. The method may be seen as application of Lagrangian relaxation and is as such based on sufficient optimality conditions. The forwards algorithms also detect and exploit decision and forecast horizons; these are expedient for the algorithmic efficiency but also for the interpretation of the solution's characteristics. It is shown that monotonicity relations are essential for the algorithms. Simultaneously, problems that have these characteristics display interesting solution structures and properties. The forwards algorithms therefore illustrate elegantly the transparency obtained in the optimal control approach.

Chapter 10 contains conclusions and suggestions for further research.

The author's contributions to the thesis fall into two categories. One is the synthesis of discrete time optimal control. Thus, the thesis presents a framework for the understanding, application and further development of this discipline such that different general perspectives as well as particular results may be interpreted and appreciated. The synthesis presents modern results from mathematical programming, selected and applied according to the discrete time optimal control problem's intrinsic characteristics. The perspective that the dynamic aspects are fundamental i.e., the stagewise or sequential character of the problem, is maintained in focus. Hence essentially all the relevant contributions from mathematical programming theory have been reformulated to fit into the stagewise perspective. Further, the extensive literature of discrete time optimal control has been searched and what is considered to be major contributions are included with suitable reformulations according to the context. Thus, a comprehensive presentation of discrete time optimal control is given.

The second category of the author's contributions consists of a number of specific results. The major ones may be outlined as follows. In Chapter 1 the applications in Section 1.5 and Section 1.6 are new, and the algorithms and computational complexity results on the isotone regression problem, Section 1.7, are new. Parts of the background material included in Chapter 2 was developed in Nahorski, Ravn and Vidal (1987), but the organization, formulations and many details are new; in particular the central observation on the distinctive differences between the stagewise and total (mathematical programming) constraint qualifications, Section 2.5, is new. Details of Proposition 2.8.1 are new. Part of the results in Chapter 3 were given or are implicit in Nahorski, Ravn and Vidal (1983). The nonsmooth generalizations of the maximum principle, Proposition 3.4.4, Proposition 3.4.5, Proposition 3.4.6 are new, although in part foreseen in Outrata (1984). The results on elimination of singularities in linear problems, Proposition 3.4.8, is new. The sufficiency result of the extended maximum principle, Proposition 3.5.2, was developed independently, although previously given in Krotov (1967); the remaining results on the extended maximum principle, Propositions 3.5.4, 3.5.5, 3.5.7 are new. The application of duality with nonlinear price functions to the optimal control problem, Section 3.6, is new, and so is the demonstration that this maintains decomposability. In Chapter 4 the derivation of the approximation error in relation to dynamic programming and discretization, Proposition 4.1.1, is new. The treatment of the QLEI problem with $n = 1$, Proposition 4.4.1, the computational complexity result of Proposition 4.5.1, the general result for problems with $n = 1$, Proposition 4.6.1, and the computational complexity result on forwards dynamic programming, Proposition 4.7.1, are new. In Chapter 5 the computational complexity result of Proposition 5.2.1 and the rate of convergence result of Proposition 5.2.2 are new. The treatment of the case of primal decomposition with nonsmooth upper boundaries,

Propositions 5.3.1 - 5.3.3, is new. In Chapter 6 the convergence result for the various maximum principle algorithms, Propositions 6.1.3, 6.1.4, 6.1.5, 6.3.1, 6.4.1, 6.4.2, are believed to be new. In Chapter 7 the idea of combining dynamic programming with an active set idea to the QLEI problem is obvious, but not straightforward, and the convergence result of Proposition 7.1.1 is new. The application of the linearization method and the absolute value merit function is obvious; however, the identification of the different ways of doing so is new. The results on global convergence of the two versions of the differential dynamic programming method with nonlinear local constraints, Proposition 7.2.3, Proposition 7.2.4, are new. The rate of convergence result for differential dynamic programming on the same problem, Proposition 7.4.1, is new. In Chapter 8 the demonstration that a weighted sum of price functions provide a convex dual function, Proposition 8.1.1, is new, and so is the convergence result, Proposition 8.1.2. The Lagrangian relaxation result and the application to linearity in the state variables are relatively straightforward although new; the convergence result on the forwards-backwards sequential projection algorithm, Proposition 8.3.1, is new but the algorithm is adopted from Ravn (1990). In Chapter 9 the convergence and computational complexity result of Proposition 9.1.1 are new. The results on the Ansgar algorithm are adopted from Ravn (1987), however, Proposition 9.2.5 is new. The treatment of the linear problems and the computational complexity results of Proposition 9.3.1 and Proposition 9.3.2 are new. Planning horizon results are usually given with $n = 1$, and therefore Proposition 9.4.1 with $n \geq 1$ and Proposition 9.4.2 with $n = 2$ are new.

Contents

Notation:

i	stage index, usually runs from $i = 0$ through $i = N$
x_i	state vector, column vector with n components
u_i	control vectors, column vector with m components
r_i	criterion function at stage i, $i = 0, \ldots, N-1$; $r_i : R^{n+m} \to R$
r_N	criterion function at stage N; $r_N : R^n \to R$
f_i	dynamic column vector function at stage i, $i = 0, \ldots, N-1$; $f_i : R^{n+m} \to R^n$
V_i	local constraint set at stage i, $i = 0, \ldots, N-1$; $V_i \in R^{n+m}$
V_N	local constraint set at stage N; $V_N \in R^n$
g_i	local inequality constraint vector function at stage i; $g_i : R^{n+m} \to R^k$
g_N	local inequality constraint vector function at stage N; $g_N : R^n \to R^k$
h_i	local equality constraint vector function at stage i; $h_i : R^{n+m} \to R^\ell$
h_N	local equality constraint vector function at stage N; $h_i : R^n \to R^\ell$
p_i	costate row vector at stage i, $i = 0, \ldots, N$, with n components
λ_i, μ_i	Lagrange multipliers, row or column vectors of dimensions k and ℓ, respectively
x_i^j, u_i^j, p_i^j, λ_i^j, μ_i^j	component j in x_i, u_i, p_i, λ_i, μ_i, respectively
r_i^j, f_i^j, g_i^j, h_i^j	component j in r_i, f_i, g_i, h_i, respectively
x, u, p, λ, μ	vectors $(x_0', x_1', \ldots, x_N')'$, $(u_0', u_1', \ldots, u_{N-1}')'$, $(p_0, \ldots, p_N)$ etc.
$\nabla_u r_i(x_i, u_i)$	partial derivative of r_i with respect to u_i, a row vector with m components
$\nabla_x r_i(x_i, u_i)$	partial derivative of f_i with respect to x_i, a row vector with n components
$\nabla^2_{xx} r_i(x_i, u_i)$	the second partial derivative of r_i with respect to x_i, x_i, a $n \times n$ matrix
$\nabla^2_{xu} r_i(x_i, u_i)$	the second partial derivative of r_i with respect to x_i, u_i, a $m \times n$ matrix
$\nabla^2_{uu} r_i(x_i, u_i)$	the second derivative of r_i with respect to u_i, u_i, a $m \times m$ matrix
$\nabla_u f_i(x_i u_i)$	partial derivative of f_i with respect to u_i, a $n \times m$ matrix
$\nabla_x f_i(x_i u_i)$	partial derivative of f_i with respect to x_i, a $n \times n$ matrix
$\underline{x}_i$, $\overline{u}_i$, $\overline{g}_i$, $\underline{h}_i$, $\underline{f}_i$, etc.	vectors with same dimensions as x_i, u_i, etc.
H_i^x, H_i^u, G_i^x, G_i^u, F_i^x, F_i^u	$\ell \times n$, $\ell \times m$, $k \times n$, $k \times m$, $n \times n$ and $n \times m$ matrices, respectively
ub_i^{i+1}	smaller upper boundary function at stages $i, i+1$; $ub_i^{i+1} : R^{2n} \to R$
UB_i	greater upper boundary function at stage i, forwards; $UB_i : R^n \to R$
RUB_i	greater upper boundary function at stage i, backwards; $RUB_i : R^n \to R$
$u_i()$, $u_i^*()$	$u_i : R^n \to R^m$, $u_i^* : R^n \to R^m$ depend on x_i (or x_{i+1})
A, B_i	matrices where the dimensions are implicit in the context
A_i'	the matrix which is the transpose of the matrix A_i
$A < 0$, $A \leq 0$	the matrix A is negative definite, negative semidefinite
$\{(x_j, u_j)\}_{j=a}^b$	the vector $(x_a', u_a', x_{a+1}', u_{a+1}', \ldots, x_b', u_b')'$
$\{(x_j, u_j)\}$	$\{(x_j, u_j)\}_{j=a}^b$ where a and b are implicit in the context
$z'A$	the scalar product of the row vector z' and the matrix A
$(x_i^j)^2$	the square of x_i^j
$\| x_i \|$	$\sqrt{(x_i^1)^2 + (x_i^2)^2 + \ldots + (x_i^n)^2}$
$\| \alpha \|$	the absolute value of the scalar α
$\min\{\alpha_1, \alpha_2, \ldots, \alpha_s\}$	the smallest of the scalars α_1, α_2, ... ,α_s
$\max\{\alpha_1, \alpha_2, \ldots, \alpha_s\}$	the biggest of the scalars α_1, α_2, ... ,α_s
$\max_u[r(x, u)]$	maximize the function $r(x, u)$ with respect to the variable u
$\text{argmax}_u[\,]$	the value of u that maximizes the expression in [ ]

Chapter 1

Discrete Time Optimal Control

In this chapter we introduce the discrete time optimal control problem (OCP) and approaches towards its analysis.

Thus, in Section 1.1 we give a general formulation of the OCP and in Section 1.2 we present a few examples of problems that are naturally formulated as discrete time optimal control problems.

We then review in Section 1.3 the development of the discrete time maximum principle and relate it to dynamic programming. This motivates the attempts to see the interrelationships between those two sets of optimality conditions and implied solution techniques. We also review briefly the continuous time optimal control problem, from where the discrete time maximum principle took its inspiration.

We review in Section 1.4 the mathematical programming problem approach towards the OCP. The purpose here as in Section 1.3 is to see the similarities and differences in the optimality conditions in the approaches and point out parallels in their historical development. In particular we shall see that it is very easy to formulate optimality conditions in the form of a number of maximum principles. Section 1.4 also serves the purpose of presenting main concepts from mathematical programming that will be applied to the OCP in later chapters.

We then illustrate how the application of the optimal control stagewise perspective may be used in relation to the characterization of the optimal solution (Section 1.5), to the interpretation of this (Section 1.6) and to the development of algorithms (Section 1.7).

1.1 Problem Statement

We consider the following general form which we refer to as the optimal control problem (OCP):

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{1.1}$$

$$x_{i+1} = f_i(x_i, u_i), \ i = 0, \ldots, N-1 \tag{1.2}$$

$$(x_i, u_i) \in V_i, \qquad i = 0, \ldots, N-1 \tag{1.3}$$

$$x_N \in V_N \tag{1.4}$$

Here $x_i \in R^n$, $u_i \in R^m$, $r_i : R^{n+m} \to R$, $f_i : R^{n+m} \to R^n$, $i = 0, \ldots, N-1$, $r_N : R^n \to R$, $V_i \subset R^{n+m}$, $i = 0, \ldots, N-1$, and $V_N \subset R^n$. N will be assumed finite.

In this formulation x_i is the *state* and u_i is the *control* (at *stage* i). We let $x = (x_0', x_1', \ldots, x_N')'$ and $u = (u_0', u_1', \ldots, u_{N-1}')'$. We call (1.1) the *criterion*, (1.2) is the *dynamic equation*, or the *dynamics*, f_i is the *transformation function*. The restrictions (1.3) - (1.4) are called the *local restrictions* or *local constraints*. Similarly, r_i is the *local criterion*.

The condition $x_N \in V_N$ is referred to as the *end point* or *final restriction*, and $x_N \in V_N$ and r_N are together called *end conditions*. If in particular V_N contains only one point $\underline{x}_N$ then this is called the *end point*, and we may write the condition as $x_N = \underline{x}_N$. If $V_N = R^n$ then x_N is *free*. Often the *initial condition* $(x_0, u_0) \in V_0$ implies $x_0 = \underline{x}_0$. In this case $\underline{x}_0$ is the *initial point*.

In many cases (but not for instance in Example 3 in Section 1.2), the local restrictions (1.3) and the end point restriction (1.4) can be written in the more structured form

$$g_i(x_i, u_i) \leq 0, \qquad g_i : R^{n+m} \to R^k \tag{1.5}$$

$$h_i(x_i, u_i) = 0, \qquad h_i : R^{n+m} \to R^\ell \tag{1.6}$$

$$g_N(x_N) \leq 0, \qquad g_N : R^n \to R^k \tag{1.7}$$

$$h_N(x_N) = 0, \qquad h_N : R^n \to R^\ell \tag{1.8}$$

And again in many cases this may further specialize. For instance g_i and h_i may be separable or additively separable in x_i and u_i, i.e. they may be written in the form $g_i(x_i, u_i) = (g_i^x(x_i)', g_i^u(u_i)')'$ or $g_i(x_i, u_i) = g_i^x(x_i) + g_i^u(u_i)$. Often $V_i = X_i \times U_i$; and further often $X_i = R^n$ or $X_i = \{x_i \mid \underline{x}_i \leq x_i \leq \overline{x}_i\}$, and/or $U_i = R^m$ or $U_i = \{u_i \mid \underline{u}_i \leq u_i \leq \overline{u}_i\}$

Similarly, many problems have simple dynamics, for instance because f_i can be written in the additively separable form $f_i^x + f_i^u$, where further f_i^x and/or f_i^u may be linear. In particular we may have $f_i^x(x_i, u_i) = x_i$. Similar observations often apply to r_i as well.

In general the criterion function (1.1) will contain contributions from all stages $i = 0, \ldots, N$. Sometimes there is no term r_N while in other cases r_N is the only term. These three variants are in the continuous time tradition referred to as the *Bolza* problem (cf. (1.1)), the *Lagrange* problem (cf. (1.9)) and the *Mayer* problem (cf. (1.22)), respectively.

An important class of problems is the so called *two-point boundary value problem*. This problem is characterized by having local contraints that depend only on the control u_i, except for $i = 0$ and possibly also for $i = N$. Other specific problems are *linear problems* and *quadratic-linear problems*. A *stationary problem* has r_i, f_i that are independent of i for $i = 0, \ldots, N-1$, and V_i independent of i for $i = 1, \ldots, N-1$.

Any problem of the form (1.1) - (1.4) can be manipulated such that it has a fixed initial point. This may be required for some algorithms. To see this, increment the stage indexes on all functions and sets by 1, and also increment N by 1. Then define a new transformation function $f_0(x_0, u_0) = x_0 + u_0$, a new local criterion $r_0(x_0, u_0) \equiv 0$ and define $V_0 = \underline{x}_0 \times R^m$, where $\underline{x}_0$ is any point in R^n. The new problem is seen to be essentially equivalent to the original one.

Any problem of the form (1.1) - (1.4) can be transformed such that has a fixed end point, i.e., V_N contains only one point. To see this, define a new transformation function $f_N(x_N, u_N) = x_N + u_N$, redefine $V_N = V_N \times R^m$, redefine $r_N(x_N, u_N) = r_N(x_N)$. Define $r_{N+1}(x_{N+1}) = 0$. Define $V_{N+1} = R^n$. Finally increment N by 1. Such transformation may be convenient or required for some algorithms.

The optimal solution of the problem remains the same by such transformations, however other characteristics of the problem (e.g. strict concavity of the criterion function or sensitivity information in the form of shadow prices) may change.

The dimensions of x_i and u_i need not be the same for all i. Similarly the restrictions (1.3) may have varying dimensions, depending on i. This can be handled by appropriate adjustment in the

additional variables and parameters, we shall use (Lagrange multipliers, for instance). We shall disregard this refinement, in order to keep notation as simple as possible.

In many cases the stage index i corresponds to "time". It is not a requirement for a practical case that this is so. The index i may very well represent any other one-dimensional parameter, which takes a sequence of integer values. We shall in any case refer to the index i as the *stage* index.

As an example where the stage index does not correspond to "time" we may take the knapsack problem. In the knapsack problem the purpose is to select which of N items shall be included in a knapsack of limited volume, such that the total value of the items selected is as high as possible. In this problem, i corresponds to the number of the item which is considered to be included in the knapsack. This problem can also be formulated in the form (1.1) - (1.4). In this case u_i is a zero/one variable, indicating whether item i is to be included ($u_i = 1$) or not ($u_i = 0$). We introduce additionally the one-dimensional state vector x_i, which represents the volume used up in the knapsack by the items with indexes $j = 0$ to $j = (i-1)$. The dynamical equation is then $x_{i+1} = x_i + c_i u_i$, where c_i is the volume of item i. Further, the only restriction on x_i are $x_0 = 0$ and the final restriction $x_N \leq \underline{x}_N$ where $\underline{x}_N$ is the volume of the knapsack. The purpose is to maximize $[\sum_{i=0}^{N-1} a_i u_i]$, where a_i is the value of item i. We see that this is a variant of the two points boundary value problem.

Thus many different problems may be represented by (1.1) - (1.4). The essential point is the mathematical structure as expressed in this problem.

Clearly, the structure of the optimal solution, the optimality conditions and their interpretation as well as the methods which can be used to solve the problem will depend on the way the functions and restrictions are specified. The more structure that is present, the more structure of the optimal solution, and the more possibilities of solution. In particular, the nature of the local restrictions are very important, for instance if we are restricted to a discrete set of points, or if we are dealing with a continuous or even smooth problem. Thus for instance the first and second examples in the next section, that of the heat accumulator, and that of the water network, will in many particular formulations admit solution by the maximum principle and interpretation of the solution in terms of shadow prices. The third example given below, that of the electrical power plant, could typically be solved by dynamic programming; and the concept of shadow price does not readily apply.

In summary we see that the discrete time optimal control problem that we shall work with may be characterized as a finite dimensional optimization problem with the following specific features:

- the objective function (1.1) is additively separable between the stages
- the dynamics (1.2) links only stages i and $i+1$ and uniquely specifies x_{i+1} from (x_i, u_i)
- the local constraints (1.3) - (1.4) are separable between the stages
- typically N is large (but finite) relative to n and m.

The stagewise nature is not limited to the formulation of the problem. Indeed it applies also to the characterization of the optimal solution and to the interpretation of the solution, and it may further be exploited in algorithms that are formulated stagewise. This in fact is the major point in the present work.

1.2 Examples of Discrete Time Optimal Control Problems

We now give a few examples that illustrate, how practical problems may be formulated as discrete time optimal control problems in the form (1.1) - (1.4). For a survey see Ravn and Vidal (1990).

Example 1: Maximum Benefit from a Heat Accumulator

In a combined heat and power (CHP) plant electrical power is distributed via the electrical network and heat is distributed to a district heating system. There is also a heat accumulator in the form of a water tank, which can store heat as hot water.

The purpose of the heat accumulator is to permit a deviation between heat production and demand in any given hour, thus permitting the shift of production towards those hours of the day, where production is most profitable or least costly. Further, the accumulator can be used to cover the demand during peak hours, where the capacity of the CHP unit may be insufficient. The electrical power produced simultaneously with the heat is assumed sold to the electrical network at prices which vary over the day.

The problem is to find production levels for the heat produced, in order that the demand for heat be satisfied, and in the most economical way.

Let the following be given: the demand for heat in the district heating system, expressed as the quantity d_i to be sent from the CHP plant (including the accumulator) in hour i, $i = 0, \ldots, 167$, during one week; the heat production lower and upper limits at hour i, $\underline{q}_i$ and $\overline{q}_i$, respectively; the time dependent profit (income from sale of electrical power minus cost of production) $r_i(q_i)$, as a function of the production level q_i; the lower and upper limits at hour i, $\underline{x}_i$ and $\overline{x}_i$, respectively, of the accumulator; the loss in the accumulator, represented by the constants α and β, where $0 < \alpha \leq 1$ and $0 \leq \beta$; and the inital and final required contents in the accumulator, $\underline{x}_0$ and $\underline{x}_{168}$, respectively.

We can then formulate the problem as follows

$$\max[\sum_{i=0}^{167} r_i(q_i)] \tag{1.9}$$

$$x_{i+1} = \alpha x_i + q_i - d_i - \beta \tag{1.10}$$

$$\underline{q}_i \leq q_i \leq \overline{q}_i \tag{1.11}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{1.12}$$

$$x_0 = \underline{x}_0 \tag{1.13}$$

$$x_{168} = \underline{x}_{168} \tag{1.14}$$

If we do not require a specific final level $\underline{x}_{168}$ in the accumulator we may substitute (1.14) by a constraint of the type (1.12) for $i = N$. If we prefer some final levels to others, we may indicate this by adding to the criterion (1.9) a term $r_{168}(x_{168})$.

Example 2: Control of a Water Network

Consider a water network consisting of the following 4 types of elements:

- reservoirs v^j with indexes $j = 1, \ldots, n$ where water can be stored
- connecting flows q^j with indexes $j = 1, \ldots, m$ where flow number j transports water from reservoir v^a to reservoir v^b

- outflows s^j where flow number j takes water from reservoir v^c
- inflows d^j where flow number j takes water to reservoir v^e

Thus, the volume in reservoir j at time $(i+1)$ can be seen to be given by the following dynamics relating the volumes and flows at time i to the volumes at time $(i+1)$:

$$v_{i+1}^j = v_i^j + \sum_{k \in D_j} d_i^k - \sum_{k \in S_j} s_i^k + \sum_{k \in Q_j^+} q_i^k - \sum_{k \in Q_j^-} q_i^k \tag{1.15}$$

Here subindexes i and $(i+1)$ refer to time periods; D_j is the set of indexes, for which inflow d^k goes to reservoir number j; S_j is the set of indexes, for which outflow s^k comes from reservoir number j; Q_j^+ is the set of indexes for which connecting flow q^k goes to reservoir j; and Q_j^- is the set of indexes for which q^k goes from reservoir j.

Individual volumes and flows are subject to lower and upper bounds in all time periods i:

$$\underline{v}^j \leq v_i^j \leq \overline{v}^j \tag{1.16}$$

$$\underline{d}^j \leq d_i^j \leq \overline{d}^j \tag{1.17}$$

$$\underline{q}^j \leq q_i^j \leq \overline{q}^j \tag{1.18}$$

$$\underline{s}^j \leq s_i^j \leq \overline{s}^j \tag{1.19}$$

The flows that are controllable and the purpose of controlling the flows (and thus the criterion function) depend on the circumstances.

If the network is for supply of use water we may consider s_i^j as given demands for water. Then the problem is to find inflows d_i^j, connecting flows q_i^j and reservoir levels v_i^j such that the cost of pumping the water is minimized. Let $v_i = (v_i^1, \ldots, v_i^j, \ldots, v_i^n)'$, where n is the number of reservoirs. Similarly define d_i, q_i and s_i of appropriate dimensions. We denote the cost of pumping at time i by $r_i(v_i, q_i, s_i)$. If we consider the time periods to be the hours of one day, we then want to solve

$$\min[\sum_{i=0}^{23} r_i(v_i, q_i, s_i)] \tag{1.20}$$

subject to the restrictions on flow and reservoir levels, and subject to the dynamics of the network. To obtain a criterion formulated with maximization rather than minimization we multiply by -1 to obtain

$$\max[-\sum_{i=0}^{23} r_i(v_i, q_i, s_i)] \tag{1.21}$$

The network may also represent a sewer network. In this case, d_i^j is considered a given inflow of sewer and rain water, and s_i^j is the amount of water, which leaves the network.

The purpose of the control of the network may in this case be to end up with a small amount of water in the reservoirs at final time $i = 24$. This may be obtained by solving the problem

$$\max[-\sum_{j=1}^{n} w_{24}^j (v_{24}^j)^2] \tag{1.22}$$

where w_{24}^j are positive weight scalars, J is the set of indexes j on s, and n is the number of reservoirs.

Finally the network may serve the purpose of production of hydro power. The cost of hydro production may be set to zero, while the alternative or supplementary power sources (thermal units for instance) may be represented by the cost functions $r_i : R \to R$. Thus, the cost of satisfying the demand a_i for power during period i by thermal units will be $r_i(a_i)$. Hydro power may be produced by (part of) the flows s_i^j and q_i^k, corresponding to index sets J and K, respectively. The cost of production during period i is then

$$r_i(a_i - \sum_{j \in J} s_i^j - \sum_{k \in K} q_i^k) \tag{1.23}$$

Let further the functions $r_N^j : R \to R$ represent the value of the final volume of storage j. Then the problem of scheduling the production over the 52 week of the year may be formulated with the criterion

$$\max[-\sum_{i=0}^{51} r_i(a_i - \sum_{j \in J} s_i^j - \sum_{k \in K} q_i^k) + \sum_{j=1}^{n} r_{52}^j(v_{52}^j)] \tag{1.24}$$

Obviously, in all the interpretations here stochastic elements are present in the problem considered, typically the inflows. It may therefore be relevant to model this as well.

Example 3: Unit Commitment of an Electrical Power Plant

Consider an electrical power plant which is to be operated optimally over time periods from $i = 0$ to $i = 167$, representing the hours of one week. At time i, the plant can produce u_i, where $u_i \in U_i = \{u_i \in R \mid u_i = 0 \vee \underline{u}_i \leq u_i \leq \overline{u}_i\}$, with $0 < \underline{u}_i$.

The production costs (fuel, manpower cost, etc.) are $C_i^p(u_i)$. The produced amount, u_i, is assumed sold to the electrical network at the unit price μ_i resulting in the income $\mu_i u_i$ at time i.

Apart from the production cost, there may be startup costs. If the unit was producing in period $(i-1)$ (i.e., $\underline{u}_{i-1} \leq u_{i-1}$) then the startup cost is zero. Otherwise, there is a positive startup cost, which depends on how long time the unit has been off. We denote by $C_i^s(x_i)$ the startup cost at time i, provided the unit has been out of operation for x_i consecutive hours at the beginning of period i. We thus have $C_i^s(0) = 0$ and $C_i^s(x_i) > 0$ if $x_i > 0$.

To calculate the startup costs it is therefore necessary to keep track of the on-off sequence of the unit. This sequence is called the *unit commitment*. We can represent this as follows. Let x_i denote the number of hours the unit has been off at the beginning of period i. Then we have that

$$x_{i+1} = f_i(x_i, u_i) = \begin{cases} x_i + 1 & \text{if } u_i = 0 \\ 0 & \text{if } \underline{u}_i \leq u_i \end{cases} \tag{1.25}$$

It may be required that the unit cannot start at time i if it has not been off for a certain number of periods, say, $\underline{x}$. We can indicate this by adding the constraints $u_i \in U_i(x_i) = \{u_i \in R \mid u_i \in U_i \text{ and } (u_i = 0 \text{ if } 0 < x_i < \underline{x})\}$ or similarly $(x_i, u_i) \in V_i = \{(x_i, u_i) \mid x_i \in Z^+, u_i \in U_i(x_i)\}$, where Z^+ is the set of nonnegative integers. Also other restrictions on the unit commitment may be relevant.

On the other hand a simple version of the problem does not have these additional requirements and only distinguishes between on and off states, such that (1.25) simplifies to

$$x_{i+1} = f_i(x_i, u_i) = \begin{cases} 1 & \text{if } u_i = 0 \\ 0 & \text{if } \underline{u}_i \leq u_i \end{cases} \tag{1.26}$$

Now the problem of the optimal weekly operation of the power plant can be stated as

$$\max[\sum_{i=0}^{167}(\mu_i u_i - C_i^p(u_i) - C_i^s(x_i))] \tag{1.27}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{1.28}$$

$$(x_i, u_i) \in V_i \tag{1.29}$$

$$x_0 = \underline{x}_0 \tag{1.30}$$

$$x_{168} \in R \tag{1.31}$$

1.3 Development of the Maximum Principle

In this section we review the historical development of the discrete time maximum principle and give a feeling of "what" and "why". We shall also relate the principle to dynamic programming ideas.

The classical discrete time maximum principle is inspired by the maximum principle for continuous time problems, developed by Pontryagin and coworkers (1962) (see later in this section). In the beginning it was formulated in relation to the following two point boundary value problem with free end point, cf. the problem definition in the previous section. All functions are assumed to be continuously differentiable.

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{1.32}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{1.33}$$

$$u_i \in U_i \tag{1.34}$$

$$x_0 = \underline{x}_0 \tag{1.35}$$

$$x_N \in R^n \tag{1.36}$$

The maximum principle is formulated by introducing the *Hamiltonian* at stages $i = 0, \ldots, N-1$ as

$$H_i(x_i, u_i, p_{i+1}) = \begin{cases} r_N(x_N) & \text{for } i = N \\ r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) & \text{for } 0 \leq i < N \end{cases} \tag{1.37}$$

where $p_{i+1} \in R^n$ is a row vector, called the *costate* or *adjoint vector*.

The *classical maximum principle* asserts that there exist p_{i+1}^* such that at an optimal solution (x^*, u^*) the following two conditions hold for $i = 0, \ldots, N-1$:

$$u_i^* \text{ maximizes } H_i(x_i^*, u_i, p_{i+1}^*) \text{ over } u_i \in U_i \tag{1.38}$$

$$p_i^* = \nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*), \text{ with } p_N^* = \nabla r_N(x_N^*) \tag{1.39}$$

We see that the maximum principle is a sort of decomposition principle. The "maximum" part (1.38), i.e. the maximization with respect to u_i, is "local", i.e. stagewise. But u_i^* is linked to stage $(i+1)$ by the inclusion of p_{i+1} in the definition of the Hamiltonian at stage i. The "adjoint equations part" (1.39) links adjoining stages and thus account for the "global" aspects. Observe that the adjoint equations have a simple backwards recursive form which parallels the forwards recursive form of the dynamic equation (1.33).

We may formulate the maximum principle as follows:

Proposition 1.3.1 *Consider the OCP (1.32) - (1.36) and assume that it has an optimal solution* (x^*, u^*). *Assume that all functions are continuously differentiable. Assume that for all* i H_i *is concave with respect to* u_i *for any* (x_i, p_{i+1}) *and* U_i *is convex. Then there exists a* p^* *such that (1.38) - (1.39) hold. If in addition* r_i *are concave and* f_i *are linear then any* (x^*, u^*) *satisfying (1.38) - (1.39) and (1.33) - (1.35) is optimal.*

Proof. See e.g. Propositions 3.4.3 and 3.4.7. □

To see why something like the maximum principle could be valid, we can relate it to dynamic programming ideas. The principle of optimality was formulated in Bellman (1957). This principle immediately suggests a computational technique, dynamic programming. This technique exploits the stagewise nature of the problem, by calculating RUB_i in (1.40) in a recursive way. However, it has inherent practical difficulties in representing and calculating the function RUB_i in (1.40). Typically a discretization is used, representing RUB_i in a finite (small) set of grid points. The weakness is that the number of points needed to represent RUB_i with acceptable accuracy typically grows rapidly with n. This is referred to as the curse of dimensionality. The idea of DP may be indicated as follows.

Let us define $RUB_i(x_i)$ as the optimal value of the criterion for the partial problem that starts at stage i with state x_i. That is,

$$RUB_i(x_i) = \max[\sum_{j=i}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{1.40}$$

subject to (1.33) - (1.36) and x_i fixed. This is also what is called the optimal value function in dynamic programming. (See Chapter 3 for more on this). We then have the following

Proposition 1.3.2 u_i^* *maximizes* $[r_i(x_i^*, u_i) + RUB_{i+1}(f_i(x_i^*, u_i))]$ *over* $u_i \in U_i$.

Proof. See Proposition 3.2.1. Essentially the argumentation states that u_i^* is maximizing because if not, u_i^* could not be optimal, due to the definition of RUB_{i+1}. It is implicitly assumed that at a point where RUB_{i+1} is not defined we have $RUB_{i+1} = -\infty$. □

Observe that this is formulated without smoothness assumptions. This is one of the characteristics of the dynamic programming approach.

Making smoothness assumptions we have the following link between the optimality conditions of dynamic programming and the maximum principle.

Proposition 1.3.3 *Assume that* r_i *and* f_i *are continuously differentiable and that the optimal solution is unique. Then*

$$\nabla RUB_i(x_i^*) = \nabla_x(r_i(x_i^*, u_i^*) + RUB_{i+1}(f_i(x_i^*, u_i^*)))$$

Proof. See Proposition 2.7.3. Essentially the argumentation may be interpreted to say that if not the relation holds, it would be possible to find a point (x_i, u_i) near (x_i^*, u_i^*), such that $r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i)) > RUB_i(x_i)$, contradicting the definition of RUB_i. □

These two results motivate that the maximum principle could hold. Clearly one should try to select p_i^*, such that $p_i^* = \nabla RUB_i(x_i^*)$; provided, of course, that RUB_i is smooth at this point. It

is in fact not trivial to guarantee this crucial assumption. It is, however, fulfilled if there is no local constraint that links u_i to x_i and if x_N is free. This indicates why the problems with state constraints were analyzed much later than those with pure control constraints. This also justifies, that with the problem being defined as (1.32) - (1.36), the differentiability assumptions (in other words, the adjoint equations) were not the main topic in the early treatments of the discrete time maximum principle.

The main topic was the establishment of conditions to assure that the maximum part was true. In continuous time the maximum part holds under relatively mild assumptions, see below. Fan and Wang (1964) concluded incorrectly that this is also the case for discrete time systems. Rozonoer (1959) concluded correctly that the maximum principle is not generally valid for discrete time systems, although it is valid for the linear system (i.e., linear r_i and f_i and $U_i = R^m$). Chang (1960, 1961), Butkovski (1963, 1965), Horn and Jackson (1965), Jackson and Horn (1965), Gabasov(1968) further discussed this.

A *weak maximum principle* (Katz (1962a, 1962b)) was formulated. It states that the Hamiltonian takes a *stationary* value with respect to u_i at the optimal point. It may be formulated as follows, assuming that $U_i = R^m$:

There exist p^*_{i+1} such that for $i = 0, \dots, N-1$ there holds

$$\nabla_u H_i(x^*_i, u^*_i, p^*_{i+1}) = 0 \tag{1.41}$$

$$p^*_i = \nabla_x H_i(x^*_i, u^*_i, p^*_{i+1}), \qquad \text{with } p^*_N = \nabla r_N(x^*_N) \tag{1.42}$$

Assuming that the set U_i is given as $\{u_i \in R^m \mid g_i(u_i) \leq 0, h_i(u_i) = 0\}$ we may formulate a similar stationarity result in the familiar terms of the Karush-Kuhn-Tucker (KKT) conditions. The constraint qualifications mentioned are further discussed in Section 2.5.

Proposition 1.3.4 *Let the OCP (1.32) - (1.36) be given with* $U_i = \{u_i \in R^m \mid g_i(u_i) \leq 0, h_i(u_i) = 0\}$ *and assume that it has an optimal solution* (x^*, u^*). *Assume that all functions are continuously differentiable. If a constraint qualification holds, then there exist* p^* *and* (λ^*, μ^*) *such that*

- $\nabla_u(H_i(x^*_i, u^*_i, p^*_{i+1}) - \lambda^*_i g_i(u^*_i) - \mu^*_i h_i(u^*_i)) = 0,$
- $\lambda^*_i \geq 0,\ \lambda^*_i g_i(u^*_i) = 0$
- $p^*_i = \nabla_x H_i(x^*_i, u^*_i, p^*_{i+1}),$

If r_i *are concave,* f_i *are linear and* U_i *are convex, then any feasible* (x^*, u^*) *satisfying this is optimal.*

Proof. See Proposition 1.4.6. By observing that the OCP can be formulated as a mathematical programming problem, see Section 1.4, this result is (now!) a standard result. □

We observe at this point that our historical account is not truly chronological. The formulation above of the weak maximum principle uses the KKT (Karush-Kuhn-Tucker) multipliers (λ_i, μ_i). This was not the case when the development of the maximum principles took place. In fact, the development of the optimal control theory and mathematical programming took place as separate developments (as it has, essentially, done since).

The existence of p^*_{i+1} such that u^*_i maximizes H_i over U_i was discussed in Halkin (1964, 1966), and Propoi (1964, 1965) in geometric terms. The essential condition was that the set of reachable extended states $\hat{Y}_{i+1}$ was convex; $\hat{Y}_{i+1}$ was defined as the set of $\hat{x}_{i+1} = (x^o_{i+1}, x'_{i+1})'$, with $x^o_{i+1} =$

$\sum_{j=0}^{i} r_j(x_j, u_j)$ and (1.33) - (1.35) satisfied. This condition is in fact very restrictive. By observing that $\hat{x}_{i+1}$ would only be located at the upper (in the direction of x_{i+1}^o) boundary of $\hat{Y}_{i+1}$, the convexity requirement should only concern this part. This led to the definition of *directional convexity*, discussed in Holtzman (1966a, 1966b), Holtzman and Halkin (1966). The fact that $\hat{x}_{i+1}$ is located at the upper boundary was called *the principle of optimal evolution* by Halkin (1964).

Sufficient conditions to guarantee directional convexity can be formulated such that they are also sufficient conditions for optimality by a point satisfying the maximum principle. The second part of the following was formulated in Vidal (1987) as the sufficient maximum principle.

Proposition 1.3.5 *Assume that r_i and f_i are continuously differentiable, r_i concave and f_i linear, and that U_i are convex. Assume that an optimal solution (x^*, u^*) exists. If a constraint qualification holds, then there exist p_{i+1}^* such that the maximum principle holds. If a (x^*, u^*) satisfying (1.33) and a p^* exist such that the maximum principle holds, then (x^*, u^*) is optimal.*

Proof. See 1.4.1. As above this also follows after formulating the OCP as a mathematical programming problem, see Section 1.4. □

In an attempt to extend the applicability of the maximum principle, new formulations were used. Thus Butkovski (1963) suggested a *local maximum principle*, where the content is clear from the title. Gabasov and Kirillova (1966) proposed a *quasimaximum principle*. This states that if u_i^* does not maximize $H_i(x_i^*, u_i, p_{i+1}^*)$ at least the difference between the maximum value of the Hamiltonian and the value attained at u_i^* is bounded. They gave bounds for the deviation by analyzing second derivatives of the functions involved. Further results in this direction were obtained in Gabasov and Tarasenko (1971) and Aschchepkov and Gabasov (1972) and called there *higher-order necessary conditions*. These results have not been taken any further, nor have they been applied in practical problem solving. The assumptions of smoothness were weakened in Doležal (1982, 1988), Outrata (1984), Vinter (1988).

A different line can be followed by redefining the Hamiltonian. In fact, Proposition 1.3.2 shows that the maximum part of the maximum principle can always be made to hold, provided we use another Hamiltonian. Introduce a function $\pi_{i+1} : R^n \to R$ and let

$$H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i)) \tag{1.43}$$

then clearly, with $\pi_{i+1} = RUB_{i+1}$, the maximum part of the maximum principle holds if RUB_{i+1} is defined at the point $f_i(x_i^*, u_i^*)$.

This approach poses two difficulties. One is, that RUB_{i+1} is not known; if it were, then there were no need to use the maximum principle. The other is, that it is not obvious that π_{i+1} can be selected, such that the adjoint equation holds.

The following *nonlinear*, or *generalized maximum principle* uses the upper boundary in the forwards direction, or, in dynamic programming terms, the optimal value function calculated forwards, UB_{i+1}, see Section 2.1. It was given in Nahorski, Ravn and Vidal (1983). A similar result was derived in Yakovlev (1978).

Proposition 1.3.6 *Assume that r_i and f_i are continuously differentiable, and that UB_i are continuously differentiable at x_i^*. Then there exist continuously differentiable π_{i+1}^* such that at the optimal solution (x_i^*, u_i^*) there hold*

- *u_i^* maximizes $H_i(x_i^*, u_i, \pi_{i+1}^*)$ over $u_i \in U_i$*

- $\nabla\pi_i^*(x_i^*) = \nabla_x H_i(x_i^*, u_i^*, \pi_{i+1}^*)$, *with* $\nabla\pi_N^*(x_N^*) = \nabla r_N(x_N^*)$

Proof: See the discussion around Proposition 3.4.6. □

We have in this survey shown how the maximum principle has been developed in relation to discrete time optimal control. The problem (1.32) - (1.36) discussed here is typical in the sense that there are no state dependent constraints. It is not incidental that this version has been central in the development, because this limitation permits easier analysis. As will be seen in subsequent chapters, the introduction of state dependent constraints may, partially at least, obstruct the stagewise analysis.

It is interesting to observe, that investigation of the linkage between the maximum principle and dynamic programming was never seriously on the agenda. Exceptions are Krotov (1967) and Dreyfus (1976). Thus our account here, which used the dynamic programming concepts UB_i and RUB_i is in this respect historically misleading. However, it provides insight. Also it serves as an ouverture to the considerations which follow.

The inspiration came from the continuous time optimal control, Pontryagin et al. (1962), which we present now.

Continuous Time Optimal Control

Let us consider the following continuous time optimal control problem:

$$\max[\int_0^1 r(x(t), u(t), t)\mathrm{d}t] \tag{1.44}$$

$$\dot{x}(t) = f(x(t), u(t), t) \tag{1.45}$$

$$u(t) \in U \tag{1.46}$$

$$x(0) = \underline{x}_0 \tag{1.47}$$

Here $t \in R$, $x(t) \in R^n$, $u(t) \in R^m$, $U \subseteq R^m$, $r : R^{n+m+1} \to R$, $f : R^{n+m+1} \to R^n$.

Observe that we have no state constraints other than a given initial point, and that the restriction (1.46) on the control is independent of time. This is in line with the early theoretical developments, but not necessary although it simplifies the analysis considerably. See e.g. Feichtinger and Hartl (1986), and see Hartl, Sethi and Vickson (1995) for a survey of the continuous time maximum principle in relation to state constraints.

An important variant has free final time (rather than ending at the specified time $t = 1$) such that determination of the optimal final time is part of the problem. Also problems with infinite time horizon are common.

It is quite obvious that by discretization of time t and considering only controls, that are constant on each time interval, we get something similar to the discrete time optimal control problem defined in Section 1.1 (although the question is subtle, cf. e.g. Sage and White (1977), p. 136, Cullum (1972)). The similarity between the two problems may be more easily seen if we define the dynamic equation in discrete time as

$$x_{i+1} - x_i = f_i(x_i, u_i) \tag{1.48}$$

rather than as in (1.33); with this new definition, the adjoint equation will become

$$p_i - p_{i+1} = \nabla_x H_i(x_i, u_i, p_{i+1}) \tag{1.49}$$

We introduce in continuous time the costate row vector $p(t) \in R^n$ and define the Hamiltonian

$$H(x(t),u(t),p(t),t) = r(x(t),u(t),t) + p(t)f(x(t),u(t),t) \tag{1.50}$$

The maximum principle, formulated by Pontryagin and co-workers (1962) can be stated as the following necessary optimality conditions:

Proposition 1.3.7 *Assume that r, f and $\partial f/\partial x$ are continuous with respect to (x,u,t). Let $\{u^*(t)\}$ be a piecewise continuous optimal control and $\{x^*(t)\}$ the corresponding trajectory. Then there is a continuous and piecewise continuously differentiable function $p^*(t) \in R^n$ such that*

- $\dot{x}^*(t) = \partial H(x(^*t),u^*(t),p^*(t),t)/\partial p$
- $\dot{p}^*(t) = -\partial H(x^*(t),u(^*t),p^*(t),t)/\partial x$ *for all t except where $\{u^*(t)\}$ is discontinuous*
- $u^*(t)$ *maximizes $H(x^*(t),u(t),p^*(t),t)$ over $u(t) \in U$.*

Proof. See Seierstad and Sydsæter (1987) p. 85, p. 182. □

The similarity to the discrete time maximum principle with the relations (1.48) - (1.49) is obvious.

An interesting observation is that there is no requirements of concavity, linearity or convexity involved in the necessary conditions of the above proposition. From this it can *not* be concluded that these assumptions are not needed for the validity of the discrete time analogue of the maximum principle. In fact, it took about ten years before the questions concerning this were settled, cf. the above discussion. Assumptions in line with the constraint qualification discussed above must be introduced if the local constraints involve state variables, or end point constraints are present.

Sufficient conditions for optimality may be expressed in the following form. Observe, that again the assumptions on concavity, linearity or convexity are somewhat weaker that in the discrete time case, cf. e.g. Proposition 1.3.1.

Proposition 1.3.8 *Make the same assumptions as in Proposition 1.3.7, except that $\{u^*(t)\}$ need not be optimal. Then $\{u^*(t)\}$ is indeed optimal under either of the following sets of assumptions.*

- *f is continuously differentiable with respect to u, U is convex and $H(x,u,p^*(t))$ is concave in (x,u) for all t.*
- *$\hat{H}(x,p^*(t),t) \equiv \max_{u(t)\in U}[H(x,u,p^*(t))]$ exists and is concave in x for all t.*

Proof. These are easy modifications from Seierstad and Sydsæter (1987) pp. 105 - 108. □

A second important result in continuous time is the Hamilton-Jacoby equation (1.51), also called the Hamilton-Jacoby-Bellman equation. Define $\hat{R}(x(t),t)$ as the optimal value in that part of problem (1.44) - (1.47) which starts at time t with the state $x(t)$ and ends at time $t = 1$. That is, $\hat{R}$ is the dynamic programming optimal value function (its discrete time analogue was called RUB above, cf. (1.40)).

Consider the following partial differential equation

$$\frac{-\partial F(x(t),t)}{\partial t} = \max_{u(t)\in U}[r(x(t),u(t),t) + (\frac{\partial F(x(t),u(t),t)}{\partial x})f(x(t),u(t),t)] \tag{1.51}$$

with boundary condition $F(x(1),1) = 0$.

The following result on sufficient optimality conditions links these concepts.

Proposition 1.3.9 *Assume that r, f and $\partial f/\partial x$ are continuous with respect to (x,u,t). Suppose that there exists a continuously differentiable function $F : R^n \times [0,1] \to R$ which satisfies (1.51) and the boundary condition. If $\{u^*(t)\}$ is piecewise continuous and $\{x^*(t)\}$ is the corresponding solution to (1.45) and if*

$$\frac{-\partial F(x^*(t),t)}{\partial t} = r(x^*(t),u^*(t),t) + (\frac{\partial F(x^*(t),u^*(t),t)}{\partial x})f(x^*(t),u^*(t),t)$$

then $\{u^*(t)\}$ is optimal in (1.44) - (1.47), and F is identical to the optimal value function $\hat{R}$ in this problem.

Proof. See Fleming and Rishel (1975) p. 87. □

The right hand side of (1.51) can be interpreted as the Hamiltonian (1.50), with $p = \partial F/\partial x$. Thus, the Hamilton-Jacoby equation links beautifully the maximum principle with dynamic programming.

It is impossible to derive a discrete time analogue to this result, although the following somehow similar results can be given. Consider the following relation which is easily derived:

$$\begin{aligned} &-(RUB_{i+1}(x_i^*) - RUB_i(x_i^*)) \\ &= r_i(x_i^*,u_i^*) + \nabla RUB_{i+1}(x_{i+1}^*)f_i(x_i^*,u_i^*) + o(\| x_i^* - x_{i+1}^* \|) \end{aligned} \tag{1.52}$$

The relation may be seen as derived under the assumption that RUB_{i+1} is smooth at x_{i+1}^* such that $o(\| x_i - x_{i+1}^* \|)/(\| x_i - x_{i+1}^* \|) \to 0$ as $\| x_i - x_{i+1}^* \| \to 0$. The relation can be seen as a discrete time analogue of (1.51), using the dynamics (1.48) and assuming that u_i^* is maximizing the Hamiltonian with $p_{i+1} = \nabla RUB_{i+1}(x_{i+1}^*)$. In general it can not be expected that $\| x_i^* - x_{i+1}^* \|$ is small so that o vanishes, and an equality like in (1.51) can not be attained in discrete time.

If RUB_{i+1} is concave then o may be omitted in the above if the equality is substituted by an inequality:

$$-(RUB_{i+1}(x_i^*) - RUB_i(x_i^*)) \geq r_i(x_i^*,u_i^*) + \nabla RUB_{i+1}(x_{i+1}^*)f_i(x_i^*,u_i^*) \tag{1.53}$$

(and something similar even if RUB_{i+1} is not smooth) but again this is not quite (1.51).

We see that the main impetus from continuous time to discrete time optimal control are two. First, the formulation of optimal control problems in the two traditions are clearly similar, with the difference being whether "time" is considered continuous or discrete. Second, the maximum principle of continuous time optimal control theory was taken as an inspiration for the similar formulation in discrete time. However, it proved to be difficult to analyze thoroughly the validity of the principle in discrete time, in particular the necessary assumptions of convexity were an obstacle. - The Hamilton-Jacoby equation, on the other hand, has not had any influence on the development of discrete time optimal control theory. Only the maximum principle was carried over.

1.4 A Mathematical Programming Approach

Around 1970 a number of works (Canon, Cullum and Polak (1970), Polak (1971), Tabak and Kuo (1971)) attempted to unify the discrete time optimal control theory and mathematical programming. In this tradition the OCP was seen as a special case of a mathematical programming

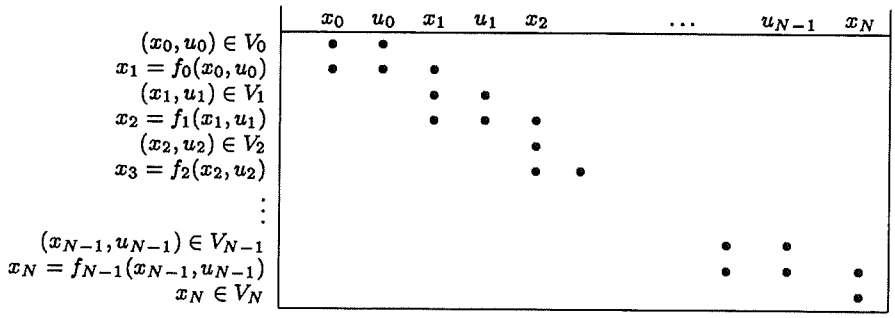


Figure 1.1: The constraint "matrix" structure, direct formulation

problem. The structure of the OCP was exploited or interpreted as a case of sparcity. With the advances of mathematical programming the interest in the specific structure of the OCP diminished, and the emphasis shifted towards the new advances of mathematical programming, as for instance duality and algorithms.

In this section we shall discuss the discrete time optimal control problem as a finite dimensional mathematical programming problem.

In order to assess the structure of the problem we consider it as a special case of the mathematical programming problem:

$$\max[r(v)] \tag{1.54}$$

$$g(v) \leq 0 \tag{1.55}$$

$$h(v) = 0 \tag{1.56}$$

where $r : R^{N(n+m)+n} \to R$, $g : R^{N(n+m)+n} \to R^{(N+1)k}$, $h : R^{N(n+m)+n} \to R^{(N+1)\ell}$ and $v = (x_0', u_0', x_1', \ldots, u_{N-1}', x_N')'$.

Consider the constraints. If they are linear, then the constraint matrix looks like illustrated on Figure 1.1. There and in the following figures the dots indicate the positions where the coefficients are not zero. If the constraints are not linear, the positions with dots in a particular row indicate the variable that enters the constraint function which that row represents; and the positions with dots in a particular column indicate the constraint functions in which the variable of that column enters.

We see from Figure 1.1 that the constraint "matrix" is highly structured, with non-zero elements only close to the main diagonal.

If we assume that r_i are twice continuously differentiable we may calculate the Hessian matrix of the criterion function (1.54). As seen on Figure 1.2, this has a similar diagonal structure.

Since x_{i+1} is uniquely given from x_i and u_i through the dynamic equation $x_{i+1} = f_i(x_i, u_i)$, the x's may be eliminated (except for x_0 which may be assumed to take a fixed value $\underline{x}_0$):

$$x_1 = f_0(\underline{x}_0, u_0) \tag{1.57}$$

$$x_2 = f_1(f_0(\underline{x}_0, u_0), u_1) \tag{1.58}$$

$$x_3 = f_2(f_1(f_0(\underline{x}_0, u_0), u_1), u_2) \tag{1.59}$$

$$\vdots \tag{1.60}$$

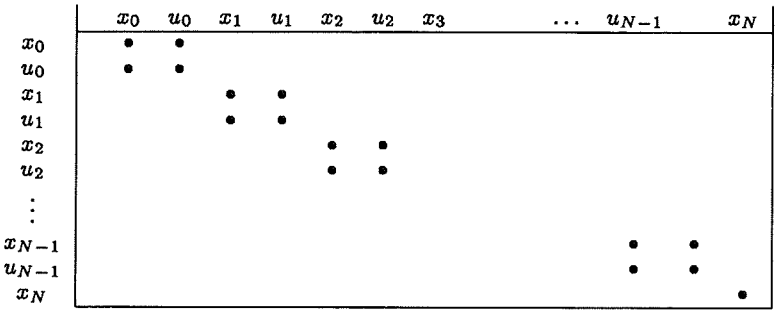


Figure 1.2: The structure of the Hessian, direct formulation

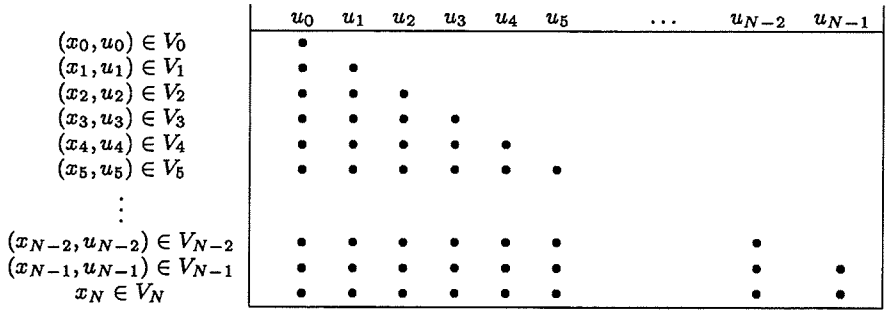


Figure 1.3: The constraint "matrix" structure, state variables eliminated

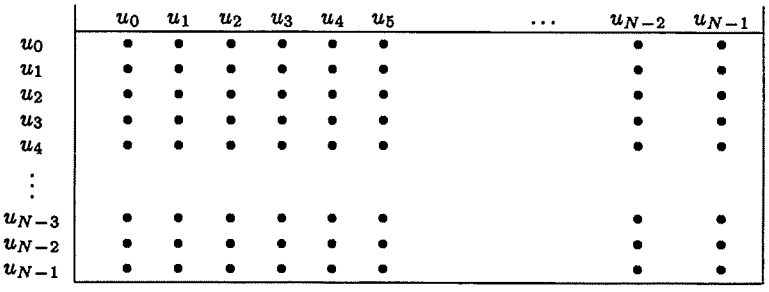


Figure 1.4: The structure of the Hessian, state variables eliminated

$$x_N = \tag{1.61}$$
$$f_{N-1}(f_{N-2}(\ldots(f_2(f_1(f_0(\underline{x}_0,u_0),u_1),u_2)\ldots),u_{N-2}),u_{N-1})$$

Then the criterion function is

$$r(u) = r_0(\underline{x}_0,u_0) \tag{1.62}$$
$$+\sum_{i=1}^{N-1} r_i(f_{i-1}(f_{i-2}(\ldots),u_{i-1}),u_i) + r_N(f_{N-1}(f_{N-2}(\ldots),u_{N-1}))$$

In this case the constraints and Hessian matrices look like illustrated on Figures 1.3 and 1.4, respectively. We see that the constraint matrix now is lower triangular, while the Hessian matrix does not display any structure. Thus the reduced number of variables reduces the structure of the matrices.

Lagrangian Relaxation

Let us now consider the possibility to exploit easily the structure of the optimal control problem by mathematical programming techniques. The two classical schemes are dual (price) decomposition (or Lagrangian relaxation) and resource (primal) decomposition. We consider here only decomposition with respect to the dynamic equation.

To use Lagrangian relaxation of the dynamic equation we introduce the Lagrange multiplier row vector $p = (p_1, p_2, \ldots, p_N) \in R^{Nn}$ and define $L^*(p)$ as

$$L^*(p) \equiv \tag{1.63}$$
$$\max_{(x,u)}[\sum_{i=0}^{N-1} r(x_i,u_i) + p_{i+1}(f_i(x_i,u_i) - x_{i+1}) + r_N(x_N)]$$
$$(x_i,u_i) \in V_i \tag{1.64}$$
$$x_N \in V_N \tag{1.65}$$

We see that this problem is additively separable with respect to the index i. Therefore the solution to the problem can be found by solving the $(N+1)$ problems

$$\max_{x_0,u_0}[r_0(x_0,u_0) + p_1 f_0(x_0,u_0)] \tag{1.66}$$
$$\max_{x_i,u_i}[r_i(x_i,u_i) + p_{i+1}f_i(x_i,u_i) - p_i x_i],\ i = 1,\ldots,N-1 \tag{1.67}$$
$$(x_i,u_i) \in V_i,\ i = 0,\ldots,N-1 \tag{1.68}$$
$$\max_{x_N}[r_N(x_N) - p_N x_N] \tag{1.69}$$
$$x_N \in V_N \tag{1.70}$$

We immediately observe that the criterion function (1.67) may be written in terms of the familiar Hamiltonian as

$$H_i(x_i,u_i,p_{i+1}) - p_i x_i \tag{1.71}$$

We have the following sufficient conditions for optimality

Proposition 1.4.1 *Suppose that (x^*, u^*) solves (1.63 - (1.65) for $p = p^*$ and that for this (x^*, u^*) the dynamic equation is satisfied. Then (x^*, u^*) solves OCP.*

Proof. This is a classical Lagrangian relaxation result, Everett (1963), applied to the OCP. □

This result was formulated in Vidal (1987) as the sufficient maximum principle.

It is interesting to observe, that this result is independent of the nature of the functions r_i and f_i and the constraint sets V_i. The only requirement is that the optimal solution (x^*, u^*) to (1.63) - (1.65) exists. Thus for instance also Example 3 on page 18 in Section 1.2 might be attempted solved this way (but not necessarily successfully).

In the remaining part of this section we shall work with structured constraint sets of the following form:

$$g_i(x_i, u_i) \leq 0, \qquad i = 0, \ldots, N-1 \tag{1.72}$$

$$h_i(x_i, u_i) = 0, \qquad i = 0, \ldots, N-1 \tag{1.73}$$

$$g_N(x_N) \leq 0 \tag{1.74}$$

$$h_N(x_N) = 0 \tag{1.75}$$

Conditions to ensure that there actually exist p^* such that Proposition 1.4.1 applies are as follows:

Proposition 1.4.2 *Assume that for all i r_i is concave, f_i linear, g_i convex, h_i linear. Assume further that a constraint qualification holds. If an optimal solution to the OCP exists then there exists a p^* such that Proposition 1.4.1 applies.*

Proof. Apply the result of e.g. Bazaraa and Shetty (1979) pp. 183 - 184 to the OCP. □

From these two result we can formulate a variety of maximum principles. We give some of the most obvious versions in the next Proposition. We use the Hamiltonian defined as

$$H_i(x_i, u_i, p_{i+1}) = \begin{cases} r_N(x_N) & \text{for } i = N \\ r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) & \text{for } 0 \leq i < N \end{cases} \tag{1.76}$$

Proposition 1.4.3 *Assume as in Proposition 1.4.2 and in addition that r_i, f_i, g_i and h_i are continuously differentiable. Then there exist p^*, λ^* and μ^* such that at an optimal solution there holds for $i = 0, \ldots, N-1$*

1. *(x_i^*, u_i^*) maximizes $[H_i(x_i, u_i, p_{i+1}^*) - p_i^* x_i^*]$ subject to $g_i(x_i, u_i) \leq 0$ and $h_i(x_i, u_i) = 0$, and x_N^* maximizes $[r_N(x_N) - p_N^* x_N]$ subject to $g_N(x_N) \leq 0$ and $h_N(x_N) = 0$.*
2. *u_i^* maximizes $H_i(x_i^*, u_i, p_{i+1}^*)$ subject to $g_i(x_i^*, u_i) \leq 0$ and $h_i(x_i^*, u_i) = 0$*
3. *$\nabla_u (H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)) = 0$*
4. *$\lambda_i^* \geq 0$, $\lambda_i^* g_i(x_i^*, u_i^*) = 0$ and $\lambda_N^* \geq 0$, $\lambda_N^* g_N(x_N^*, u_N^*) = 0$*
5. *$p_i^* = \nabla_x (H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*))$ and $p_N^* = \nabla (r_N(x_N^*) - \lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*))$*

If, on the other hand at a feasible (x^*, u^*) (1) holds or (3), (4) and (5) hold, then (x^*, u^*) is optimal in OCP.

Proof. The first part follows from Proposition 1.4.2. The second part is, under the assumptions of differentiability, concavity, constraint qualifications, linearity and convexity, the necessary and sufficient KKT conditions for optimality of the OCP. See e.g. Bazaraa and Shetty (1979) pp. 147 - 148. □

Often the local constraints are simple and the conditions simplify accordingly. In particular the last condition (5) of Proposition 1.4.3 may be formulated without reference to (λ, μ).

Thus, assume as in Proposition 1.4.3 and in addition that the local constraints can be written as $g_i^u(u_i) \leq 0$, $h_i^u(u_i) = 0$, $\underline{x}_{i+1} \leq x_{i+1} \leq \overline{x}_{i+1}$, $i = 0, \ldots, N-1$, $x_0 = \underline{x}_0$. Then the condition (5) of Proposition 1.4.3 may be formulated

- $p_i^{j*} = (\nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*))^j$ if $\underline{x}_i^j < x_i^{j*} < \overline{x}_i^j$
- $p_i^{j*} \geq (\nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*))^j$ if $\underline{x}_i^j = x_i^{j*}$
- $p_i^{j*} \leq (\nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*))^j$ if $\overline{x}_i^j = x_i^{j*}$
- $p_N^{j*} = (\nabla r_N(x_N^*))^j$ if $\underline{x}_N^j < x_N^{j*} < \overline{x}_N^j$
- $p_N^{j*} \geq (\nabla r_N(x_N^*))^j$ if $\underline{x}_N^j = x_N^{j*}$
- $p_N^{j*} \leq (\nabla r_N(x_N^*))^j$ if $\overline{x}_N^j = x_N^{j*}$

If further g_i and h_i are independent of x_i for all or some i then the conditions (5) may for these i be formulated as

- $p_i^* = \nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*)$

As seen, Lagrangian relaxation of the dynamic equation leads in a straightforward way to the formulation of many specialized maximum principles.

Search for the good value p^* of p (in the sense of Proposition 1.4.1) can be guided by duality theory, and the search for saddle points; this implies unconstrained minimization of the convex function $L^*()$ in the variables p. Subgradients to $L^*()$ may be readily calculated. Under some further assumptions first and second derivatives of $L^*()$ can be calculated. So also in this respect we are well served by the mathematical programming theory.

In practice it may be more difficult, though. Some of the most frequently found optimal control problems are linear in x, see for instance Examples 1 and 2 in Section 1.2. This means that the maximization of the Lagrangian will not yield a unique x^*. This in turn implies that the dual function $L^*()$ is not differentiable. If we therefore want to apply Lagrangian relaxation algorithms to such problems, special care must be taken upon implementation of the algorithm.

This illustrates that although the maximum principle (in this case, the sufficient maximum principle) may be convenient for characterization and interpretation of an optimal solution, it need not be readily applicable in the search for this solution.

In primal decomposition with respect to the dynamic equation we would consider a two level procedure. At the lower level, all states are assumed known, and the controls u_i are found by optimization with the states fixed at those values. This decomposes into N independent optimizations. Denoting the optimal u_i as $u_i^*(x_i, x_{i+1})$ the upper level consists of finding appropriate values for x by maximization of $\sum_{i=0}^{N-1} r_i(x_i, u_i^*(x_i, x_{i+1})) + r_N(x_N)$. See Chapter 5 for more on this.

Augmented Lagrangians

If the assumptions of concavity, linearity and convexity of Proposition 1.4.2 are not fulfilled, Lagrangian relaxation breaks down as far as necessary conditions are concerned. In this case the *augmented Lagrangian* may be applied.

We define the augmented Lagrangian AL in relation to the criterion (1.62) and the local constraints (1.72) - (1.75) as

$$\begin{aligned} AL(p,\lambda,\mu,c) \equiv \quad & \sum_{i=0}^{N-1} r_i(x_i,u_i) + r_N(x_N) \\ & + \sum_{i=0}^{N-1} p_{i+1}(f_i(x_i,u_i) - x_{i+1})) - \tfrac{1}{2}c \sum_{i=0}^{N-1} \|f_i(x_i,u_i) - x_{i+1}\|^2 \\ & -\tfrac{1}{2}c \sum_{i=0}^{N-1} \sum_{j=1}^{k} ((\max\{0, \lambda_i^j + c g_i^j(x_i,u_i)\})^2 - (\lambda_i^j)^2) \\ & -\tfrac{1}{2}c \sum_{j=1}^{k} ((\max\{0, \lambda_N^j + c g_N^j(x_N)\})^2 - (\lambda_N^j)^2) \\ & - \sum_{i=0}^{N-1} (\mu_i h_i(x_i,u_i)) - \mu_N h_N(x_N) - \tfrac{1}{2}c \sum_{i=0}^{N-1} \|h_i(x_i,u_i)\|^2 \\ & -\tfrac{1}{2}c \sum_{j=1}^{l} \|h_N(x_N)\|^2 \end{aligned} \tag{1.77}$$

As before, p, $\lambda \geq 0$ and μ are Lagrange multipliers, and $c > 0$ is a scalar. If r_i, f_i, g_i, and h_i are continuously differentiable then so is the AL.

We have the following basic result concerning the relation between the solution to the OCP and the unconstrained maximum of AL.

Proposition 1.4.4 *Assume that r_i, f_i, g_i, and h_i are twice continuously differentiable. Let (x^*,u^*) be a solution to (1.62), (1.72) - (1.75). Assume that a constraint qualification is fulfilled at (x^*,u^*) with Lagrange multipliers (p^*,λ^*,μ^*). Assume that the standard second order sufficient conditions are fulfilled at $(x^*,u^*,p^*,\lambda^*,\mu^*)$. Then there is a $\underline{c} \geq 0$, such that for all $c \geq \underline{c}$ the augmented Lagrangian has a unique local unconstrained maximum at (x^*,u^*).*

Proof. See e.g. Bertsekas (1982) pp. 108, 158-161, Luenberger (1989) pp. 409-416. See (2.69) - (2.73) for a discussion of second order sufficient conditions. □

From this it seems natural to define the *augmented Hamiltonian*, cf. (1.43) as

$$\begin{aligned} H_i(x_i,u_i,p_{i+1},c) = \\ r_i(x_i,u_i) + p_{i+1}f_i(x_i,u_i) - \tfrac{1}{2}c\|f_i(x_i,u_i) - x_{i+1}\|^2 \end{aligned} \tag{1.78}$$

This may be seen as an example of the nonlinear Hamiltonian (1.43) in relation to Proposition 1.3.6.

From the above Proposition we derive the following result, which we call the *augmented Hamiltonian maximum principle*:

Proposition 1.4.5 *Assume as in Proposition 1.4.4 and in addition that the sets* $\{u_i \in R^m \mid g_i(x_i^*, u_i) \leq 0, h_i(x_i^*, u_i) = 0\}$ *are compact. Then there is a* $c^* \geq 0$ *such that*

- u_i^* *maximizes* $H_i(x_i^*, u_i, p_{i+1}^*, c^*)$ *subject to* $g_i(x_i^*, u_i) \leq 0$, *and* $h_i(x_i^*, u_i) = 0$
- $\nabla_u(H_i(x_i^*, u_i^*, p_{i+1}^*, c^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)) = 0$
- $p_i^* = \nabla_x(H_i(x_i^*, u_i^*, p_{i+1}^*, c^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*))$, *and* $p_N^* = \nabla_x(r_N(x_N^*) - \lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*))$
- $\lambda_i^* \geq 0, \lambda_i^* g_i(x_i^*, u_i^*) = 0$, *i=0,...,N-1*, $\lambda_N^* \geq 0, \lambda_N^* g_N(x_N^*) = 0$

Proof. Consider the maximization of AL. Clearly the terms involving $h_i(x_i, u_i)$ can be omitted from the criterion, if we reintroduce the constraint $h_i(x_i, u_i) = 0$, since the omitted terms are identically zero for all (x_i, u_i) satisfying $h_i(x_i, u_i) = 0$. We can also omit from the criterion the terms involving $g_i(x_i, u_i)$ if we reintroduce $g_i(x_i, u_i) \leq 0$. The reason is that the contribution to AL of the terms involving $g_i(x_i, u_i)$ for all (x_i, u_i) with $g_i(x_i, u_i) \leq 0$ are greater than or equal to the contribution to AL of the terms involving $g_i(x_i^*, u_i^*)$ (cf. Bertsekas (1982) p. 160, Luenberger (1989) p. 415). Therefore (x^*, u^*) remains a local maximum to AL, if we omit from AL the discussed terms and reintroduce the corresponding local restrictions.

Since (x^*, u^*) is a unique local maximum, u_i^* is a unique local maximum with all other variables held fixed at their optimal values. The assumption of compactness ensures that u_i^* is also a unique global maximum, if c^* is sufficiently big (possibly bigger than $\underline{c}$ in Proposition 1.4.4). Finally, comparing AL and the augmented Hamiltonians we see that the first item is proved.

The remaining parts follow from the continuous differentiability of the functions and the fact that (x^*, u^*) is a local unconstrained maximum of AL.□

Algorithms based on the augmented Lagrangian proceed much like Lagrangian relaxation based methods. Basically, c is kept fixed, and duality theory is used to guide the search for (p, λ, μ) towards the minimum of the maximized AL (i.e., a local saddlepoint). The value of c may be increased, if convergence is not attained.

We see that with the augmented Lagrangian the stagewise decomposition property is lost, if we go for maximization with respect to (x, u); the reason is the terms involving $\|f_i(x_i, u_i) - x_{i+1}\|^2$. As optimality conditions we do have stagewise decomposition as shown in Proposition 1.4.5. This may be used for construction of algorithms, as we shall later show.

The Karush-Kuhn-Tucker Conditions

From the Karush-Kuhn-Tucker (KKT) conditions we have the following result, which we call the *weak maximum principle* ("weak" because we do not maximize with respect to u but only find a stationary value):

Proposition 1.4.6 *Assume that* r_i, f_i, g_i *and* h_i *are continuously differentiable, and that there is an optimal solution* (x^*, u^*). *Then there hold:*

- *If a constraint qualification holds at* (x^*, u^*) *then there exist* (p^*, λ^*, μ^*) *such that*
  1. $\nabla_u(H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i, {}^* u_i^*) - \mu_i h_i(x_i^*, u_i^*)) = 0$
  2. $\lambda_i^* \geq 0$, $\lambda_i^* g_i(x_i^*, u_i^*) = 0$, $\lambda_N^* \geq 0$ *and* $\lambda_N^* g_N(x_N^*) = 0$

3. $p_i^* = \nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)$ *with* $p_N^* = \nabla(r_N(x_N^*) - \lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*))$

- *If* $\sum_{i=0}^N r_i$ *is pseudoconcave,* f_i^j *and* h_i^j *are both quasiconvex and quasiconcave, and* g_i^j *are quasiconvex, and if the three conditions above hold, then* (x^*, u^*) *is optimal.*

Proof. Apply the results of e.g. Bazaraa and Shetty (1979) pp. 146 - 148 to the OCP. □

The requirement that $\sum_{i=0}^N r_i$ be pseudoconcave is fulfilled if all r_i are concave. On the other hand, if $\sum_{i=0}^N r_i$ is pseudoconcave, then in fact for all i, except at most one, r_i is concave (on this see Debreu and Koopmans (1982)) so that the assumption of pseudoconcavity is not much milder that the assumption of concavity. The requirement that f_i^j and h_i^j be both quasiconvex and quasiconcave will be fulfilled if these functions are linear. The requirement may be slightly weakened since it is only required that f_i^j be quasiconvex if p_i^j is positive and quasiconcave if p_i^j is negative; and similarly that h_i^j be quasiconvex if μ_i^j is positive and quasiconcave if μ_i^j is negative (see Bazaraa and Shetty (1979) pp. 147 - 148). Observe, though, that only after p_i^j and μ_i^j have been found do we know if they are positive or negative.

Even if the conditions of Proposition 1.4.6 are sufficient for optimality this does not imply that a maximum principle holds in a "strong" (maximizing) form. Thus, for instance, the assumption of concavity of r_i cannot be substituted by an assumption of pseudoconcavity. To see this, consider the following example: Let $n = 1$, $m = 1$, $N = 1$, $l = 0$, $k = 0$, $r_0(x_0, u_0) = e^{-(u_0)^2}$, $f_0(x_0, u_0) = x_0 + u_0$, $r_1(x_1) = -(x_1 - c)^2$ with $c = \sqrt{2}(1 + e^{-1/2})/2$, $\underline{x}_0 = 0$. For this problem the assumptions of the second part of Proposition 1.4.6 are fulfilled. We find the following unique solution to the KKT conditions: $u_0^* = \sqrt{2}/2$, $x_1^* = \sqrt{2}/2$, $p_1^* = \sqrt{2}e^{-1/2}$. From Proposition 1.4.6 it then follows that this is the unique solution to the problem. However, u_0^* is not maximizing $H_0(\underline{x}_0, u_0, p_1^*)$. In fact, $\sup_u[H_0(\underline{x}_0, u_0, p_1^*)] = \infty$.

The KKT conditions are essential in relation to necessary and/or sufficient optimality condition, for sensitivity analysis and also in a large number of algorithms for solution of the mathematical programming problem (1.54) - (1.56).

Sensitivity

The KKT conditions are also essential for sensitivity analysis. This is concerned with the analysis of the optimal value function $F : R^a \to R$, defined in relation to a perturbed version of (1.54) - (1.56), i.e., the following problem, where $z \in R^a$ is a parameter:

$$F(z) = \max_v [r(z, v)] \tag{1.79}$$

$$g(z, v) \leq 0 \tag{1.80}$$

$$h(z, v) = 0 \tag{1.81}$$

It is immediately clear that the function F is an upper boundary function in a sense very close to UB_i and RUB_i defined in Section 1.3 in relation to the OCP. In the mathematical programming literature this function is called the *perturbation function* or the *optimal value function.* The properties of F have been intensively studied over the last 15 years. Of particular interest are differentiability, Lipschitz continuity, continuity and concavity of F. We shall in the next chapter analyze the upper boundaries of the OCP along these lines. We may indicate the following classical result.

Proposition 1.4.7 *Assume that r, g and h are continuously differentiable and consider a given $\underline{z}$. Assume that there is a compact set V such that for all z near $\underline{z}$, $\{v \mid g(z,v) \leq 0, h(z,v) = 0\} \subset V$. Assume that for $z = \underline{z}$ the optimal solution v^* is unique and that the KKT multipliers λ^* and μ^* to (1.80) - (1.81) are unique. Then F defined in (1.79) is continuously differentiable at $\underline{z}$ and*

$$\nabla F(\underline{z}) = \nabla_v(r(\underline{z},v^*) - \lambda^* g(\underline{z},v^*) - \mu^* h(\underline{z},v^*))$$

Proof. See Gauvin and Debeau (1982). □

Duality

Duality is one of the most fruitful concepts in mathematical programing, with implications for problem formulation, statement and interpretation of optimality conditions and algorithms. Many of the above results may be restated in duality terms, or the duality perspective may permit similar results.

We shall here only present the classical duality results for linear programming applied to the OCP.

Consider the primal optimal control problem

$$\max_{x,u}[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N] \tag{1.82}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{1.83}$$

$$G_i^x x_i + G_i^u u_i - \overline{g}_i \leq 0 \tag{1.84}$$

$$u_i \geq 0 \tag{1.85}$$

$$x_0 = \underline{x}_0 \tag{1.86}$$

Here, R_i^x, R_i^u, F_i^x, F_i^u, $\overline{f}_i$, G_i^x, G_i^u, $\overline{g}_i$, and $\underline{x}_0$ are given matrices of appropriate dimensions. We introduce the row vectors $p_0 \in R^n$, $p_i \in R^n$, $i = 1,\ldots,N$ and $\lambda_i \in R^k$, $i = 0,\ldots,N-1$ which are dual variables to the constraints (1.86), (1.83) and (1.84), respectively. In the context of optimal control, p is referred to as the costate or dual state vector, and λ is referred to as the cocontrol or dual control vector.

The dual optimal control problem is the following one:

$$\min_{p,\lambda}[p_0\underline{x}_0 + \sum_{i=0}^{N-1} p_{i+1}\overline{f}_i + \lambda_i\overline{g}_i] \tag{1.87}$$

$$p_i = R_i^x + p_{i+1}F_i^x - \lambda_i G_i^x \tag{1.88}$$

$$R_i^u + p_{i+1}F_i^u - \lambda_i G_i^u \leq 0 \tag{1.89}$$

$$\lambda_i \geq 0 \tag{1.90}$$

$$p_N = R_N^x \tag{1.91}$$

All matrices R_i^x, R_i^u, F_i^x, F_i^u, $\overline{f}_i$, G_i^x, G_i^u, $\overline{g}_i$, and $\underline{x}_0$ are the same as in the primal problem. It is seen that the structure of the dual problem (1.87) - (1.91) is the same as that of the primal one (1.87) - (1.91), except that for the primal problem the initial state is given as $\underline{x}_0$ and the final state is free while for the dual problem the final value of the costate is given as R_N^x and the initial value p_0 is free. Alternatively, we could say that the difference is that the primal problem runs

forwards in time over indexes $i = 0, \ldots, N$ while the dual problem runs backwards in time over indexes $i = N, \ldots, 0$; this interpretation is consistent with the dual dynamics (1.88) which specifies p_i uniquely from p_{i+1} and λ_i.

The results on duality state among other things that a for solution (x, u, p, λ), where (x, u) is feasible in the primal problem and (p, λ) is feasible in the dual problem the value P^o of the primal criterion function (1.82) and the value D^o of the dual criterion function (1.87) satisfy $P^o \leq D^o$ (weak duality). Further, the two feasible solutions are optimal in the primal and the dual problems respectively, if and only if $P^o = D^o$. One of the problems has an unbounded solution if and only if the other problem is infeasible (see e.g. Luenberger (1989) p. 89).

We now restate these results in stagewise terms. We define for $i = 0, \ldots, N-1$ the primal Hamiltonian as

$$H_i^P(x_i, u_i, p_{i+1}) = R_i^x x_i + R_i^u u_i + p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) \tag{1.92}$$

which is to be maximized subject to the constraints (1.84) - (1.85). Similarly the dual Hamiltonian is defined for $i = 0, \ldots, N-1$ as

$$H_i^D(p_i, \lambda_i, x_i) = p_{i+1}\overline{f}_i + \lambda_i \overline{g}_i + (R_i^x + p_{i+1}F_i^x - \lambda_i G_i^x)x_i \tag{1.93}$$

which is to be minimized subject to the constraints (1.89) - (1.90).

It is obvious that we get the same optimal values for u_i and λ_i if we omit those parts of H_i^P and H_i^D that do not depend on u_i and λ_i, i.e substitute (1.92) - (1.93) by

$$H_i^{Pu}(u_i, p_{i+1}) = R_i^u u_i + p_{i+1}F_i^u u_i \tag{1.94}$$

$$H_i^{D\lambda}(\lambda_i, x_i) = \lambda_i \overline{g}_i - \lambda_i G_i^x x_i \tag{1.95}$$

Moreover, we observe (by adding $R_i^x x_i + p_{i+1}F_i^x x_i + p_{i+1}\overline{f}_i$ on both sides and rearranging) that

$$H_i^{Pu}(u_i, p_{i+1}) = R_i^u u_i + p_{i+1}F_u^u u_i = \lambda_i \overline{g}_i - \lambda_i G_i^x x_i = H_i^{D\lambda}(\lambda_i, x_i) \tag{1.96}$$

if and only if

$$H_i^P(x_i, u_i, p_{i+1}) = R_i^x x_i + R_i^u u_i + p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) \tag{1.97}$$

$$= p_{i+1}\overline{f}_i + \lambda_i \overline{g}_i + (R_i^x + p_{i+1}F_i^x - \lambda_i G_i^x)x_i = H_i^D(p_i, \lambda_i, x_i) \tag{1.98}$$

That is, it is not important whether one uses in proposition 1.4.8 the "full" Hamiltonians (1.92) - (1.93) (as in MacRae (1969)) or only the control dependent parts (1.94) - (1.95) (as in Propoi (1981)).

The following is as stagewise version of the duality result for linear optimal control problems.

Proposition 1.4.8 *Assume that (x, u) is feasible in the primal problem and (p, λ) is feasible in the dual problem. Then $P^o \leq D^o$. The two feasible solutions are optimal in the primal and the dual problems, respectively, if and only if $H_i^P(x_i, u_i, p_{i+1}) = H_i^D(p_i, \lambda_i, x_i)$ for all i.*

Proof. The first part is immediate from linear programming duality, see e.g. Luenberger (1989) p. 89. For the second part we observe that for values (x, u, p, λ) that are feasible in the primal and dual problems we have

$$\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N = p_0 \underline{x}_0 + \sum_{i=0}^{N-1} p_{i+1}\overline{f}_i + \lambda_i \overline{g}_i$$

if and only if (use (1.83) and (1.88))

$$\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) - p_{i+1}x_{i+1} + R_N^x x_N$$
$$= p_0 \underline{x}_0 + \sum_{i=0}^{N-1} p_{i+1}\overline{f}_i + \lambda_i \overline{g}_i + (R_i^x + p_{i+1}F_i^x - \lambda_i G_i^x)x_i - p_i x_i$$

if and only if (reduce and use (1.86), (1.91))

$$\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i)$$
$$= \sum_{i=0}^{N-1} p_{i+1}\overline{f}_i + \lambda_i \overline{g}_i + (R_i^x + p_{i+1}F_i^x - \lambda_i G_i^x)x_i$$

if and only if (use (1.92) - (1.93))

$$\sum_{i=0}^{N-1} H_i^P(x_i, u_i, p_{i+1}) = \sum_{i=0}^{N-1} H_i^D(p_i, \lambda_i, x_i)$$

Therefore $\sum_{i=0}^{N-1} H_i^P(x_i, u_i, p_{i+1}) = \sum_{i=0}^{N-1} H_i^D(p_i, \lambda_i, x_i)$ if and only if $P^o = D^o$ and by linear programming duality, $P^o = D^o$ if and only if the solution found is primal and dual optimal .

If $H_i^P(x_i, u_i, p_{i+1}) = H_i^D(p_i, \lambda_i, x_i)$ for all i then $\sum_{i=0}^{N-1} H_i^P(x_i, u_i, p_{i+1}) = \sum_{i=0}^{N-1} H_i^D(p_i, \lambda_i, x_i)$. We now show the implication the other way to complete the proof. Consider the problem similar to (1.82) - (1.86) except that it starts at stage $i = 1$ at the optimal x_1. For this problem also $\sum_{i=1}^{N-1} H_i^P(x_i, u_i, p_{i+1}) = \sum_{i=1}^{N-1} H_i^D(p_i, \lambda_i, x_i)$; therefore $H_0^P(x_0, u_0, p_1) = H_0^D(p_0, \lambda_0, x_0)$ as well. Continuing this way we see that $\sum_{i=0}^{N-1} H_i^P(x_i, u_i, p_{i+1}) = \sum_{i=0}^{N-1} H_i^D(p_i, \lambda_i, x_i)$ implies $H_i^P(x_i, u_i, p_{i+1}) = H_i^D(p_i, \lambda_i, x_i)$ for all i. □

We observe that by the way of presentation here, duality and the maximum principle are closely related. However, also dynamic programming may be brought into play, see e.g. White (1975) where linear programming duality is derived by dynamic programming argumentation.

Summary

In summary we see that the mathematical programming literature has developed a theoretical foundation involving many fruitful concepts and results. These have not been integrated in the optimal control literature. In particular we note that the concepts of duality and sensitivity analysis would have potential interest. Moreover, there has evolved an extensive body of theoretical results and practical experiences with algorithms and practical problem solving, which should be transferable to the OCP.

The result from mathematical programming may be used directly on the OCP, as shown. However, the question arises whether better (in some sense) results may be obtained if a stagewise approach is applied. This will be investigated in subsequent chapters.

1.5 Example: Structure of the Optimal Solution

In this and the following two sections we illustrate by examples how the discrete time maximum principle can be used. The main point is that the optimal control problem has a specific structure, and that this structure has implications for the structure of the optimal solution, for the interpretation of the optimal solution, and for possible solution procedures.

As an example of the application of optimality conditions for the interpretation of the solution, we consider now a version of Example 1 on page 16 in Section 1.2, the CHP plant. The model may represent many other production planning cases. Thus we have the following problem statement:

$$\max[\sum_{i=0}^{N-1} r_i(u_i)] \tag{1.99}$$

$$x_{i+1} = \alpha_i x_i + u_i - d_i - \beta_i \tag{1.100}$$

$$\underline{u}_i \leq u_i \leq \overline{u}_i \tag{1.101}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{1.102}$$

$$x_0 = \underline{x}_0 \tag{1.103}$$

$$x_N = \underline{x}_N \tag{1.104}$$

We assume $n = 1$, $m = 1$ and without further mentioning that the criterion function r_i is strictly concave. (Many similar results as below can be derived if r_i is only concave; in this case the presentation will be more difficult, since we have to handle all the time the case of non-unique solutions). Further it is assumed that r_i *is continuously differentiable.* For this problem the maximum principle as formulated in e.g. Proposition 1.4.3 holds as necessary and sufficient optimality conditions.

In the example, u_i is the production and x_i is the contents of the storage, d_i is the demand, α_i, $0 < \alpha_i \leq 1$ and β_i represent loss from the storage. In such a system much is dependent on e.g. the temperature levels and prices of electrical power, and since these may vary over the planning period, all coefficients are made stage dependent, as indicated by the subscript i. We assume throughout that there is a feasible and hence also unique optimal solution (x^*, u^*).

For this problem the Hamiltonian is

$$H_i(x_i, u_i, p_{i+1}) = r_i(u_i) + p_{i+1}(\alpha_i x_i + u_i - d_i - \beta_i) \tag{1.105}$$

which according to the maximum principle is to be maximized subject to the constraint (1.101). Therefore for given (x_i, p_{i+1}) we may find u_i as

$$u_i = \arg\max_{u_i}[H_i(x_i, u_i, p_{i+1})] \tag{1.106}$$

subject to (1.101). If in particular (x_i, p_{i+1}) is optimal then so is u_i from (1.106).

In the sequel we discuss the problem under various assumptions on some of the parameters.

1. No intermediate state constraints and $\alpha = 1$.

We first analyze the situation assuming that there are no intermediate state constraints, i.e. $\underline{x}_i = -\infty$ and $\overline{x}_i = \infty$ for $i = 1, \ldots, N-1$. Further the loss proportional to the contents of the storage is zero, i.e. $\alpha_i = 1$ for $i = 0, \ldots, N-1$.

In this case the adjoint equation takes the form

$$p_i = \nabla_x H_i(x_i, u_i, p_{i+1}) = p_{i+1} \tag{1.107}$$

In other words there is only one optimal adjoint vector p^* in the problem:

$$p_i^* = p^* \text{ for all } i. \tag{1.108}$$

The u_i^* which maximizes the Hamiltonian is easily found as the u_i^* which satisfy the following conditions:

- if $\underline{u}_i < u_i^* < \overline{u}_i$ then $\nabla r_i(u_i^*) = -p^*$
- if $\underline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \leq -p^*$
- if $\overline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \geq -p^*$

Denote $\nabla r_i(u_i^*)$ as the *marginal gain* (this could be denoted marginal cost in a minimization problem). We see that the effect of the storage is to make marginal gains equal in all periods, as far as this is permitted by the constraints on the control. In other words, we see a redistribution of production from "expensive" periods to "cheap" periods. Further we see that p^* can be interpreted as a "price" (marginal value) of heat produced and stored.

We can therefore from the optimality conditions of the maximum principle immediately see the economic significance of a storage. With time varying cost structure, we can get a leveling of the marginal gains:

> Assume $\underline{x}_i < x_i^* < \overline{x}_i$, $\underline{u}_{i-1} < u_{i-1}^* < \overline{u}_{i-1}$ and $\underline{u}_i < u_i^* < \overline{u}_i$. Then $\nabla r_{i-1}(u_{i-1}^*) = \nabla r_i(u_i^*)$

Also with time varying demands d_i (and loss β_i) we can get a leveling of production, according to the economic rationality in this.

Let us illustrate these observations with an example.

Example 1.5.1 *Let* $N = 24$, $\underline{x}_0 = \underline{x}_N = 25$, $\underline{x}_i = -\infty$ *and* $\overline{x}_i = \infty$, $i = 1, \ldots, 23$; $\underline{u}_i = 0$, $\overline{u}_i = 10$, i*=0,...,23;* $r_i(u_i) = -u_i^2 - \gamma_i u_i$, *where* $\gamma_i = 5$ *for* $0 \leq i \leq 23$; $\alpha_i = 1$, $\beta_i = 0$, $d_i = 5 + 2\sin(2\pi i/24)$, $i = 0, \ldots, 23$. □

It may be verified that $u_i^* = 5.0$ for all i. Thus, despite the demand is varying over the time periods i, we get smooth production, such that $\nabla r_i(u_i^*) = -15 = -p^*$ for all i.

This further has implications for the necessary capacity, i.e. the bounds $\underline{u}_i$ and $\overline{u}_i$. Without a storage, we have to require $\underline{u}_i \leq \min_i\{(d_i + \beta_i)\}$ and $\overline{u}_i \geq \max_i\{(d_i + \beta_i)\}$. The storage permits a redistribution between time periods. Thus, even with for instance $\overline{u}_i = 6$ we would in the above example be able to fulfill the peak demand which is 7 for $i = 6$.

The leveling of marginal prices also takes place if the local criterion r_i is time dependent.

Example 1.5.2 *Assume as in the above Example 1.5.1, but let* $\gamma_i = 1$ *for* $0 \leq i \leq 7$ *and* $18 \leq i \leq 23$, $\gamma_i = 5$ *for* $8 \leq i \leq 11$, *and* $\gamma_i = 20$ *for* $12 \leq i \leq 17$. □

The optimal solution may be verified to be $u_i^* = 7.11$ for $0 \leq i \leq 7$ and $18 \leq i \leq 23$, $u_i^* = 5.11$ for $8 \leq i \leq 11$ and $u_i^* = 0$ for $12 \leq i \leq 17$, with $p^* = 15.22$.

In Figure 1.5 we see that as γ_i follows the indicated curve, we get production following the other curve. As seen, we get a shift of production towards more "favorable" periods. We may verify that $\nabla r_i(u_i^*) = -p^* = -8.56$ except for $12 \leq i \leq 17$, where $\nabla r_i(u_i^*) = -20 \leq -p^*$, because

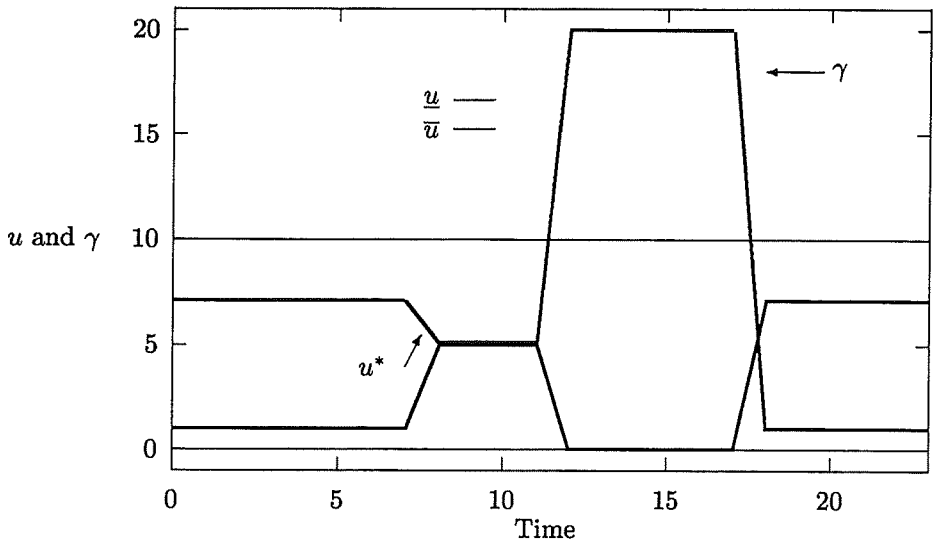


Figure 1.5: γ and the optimal control

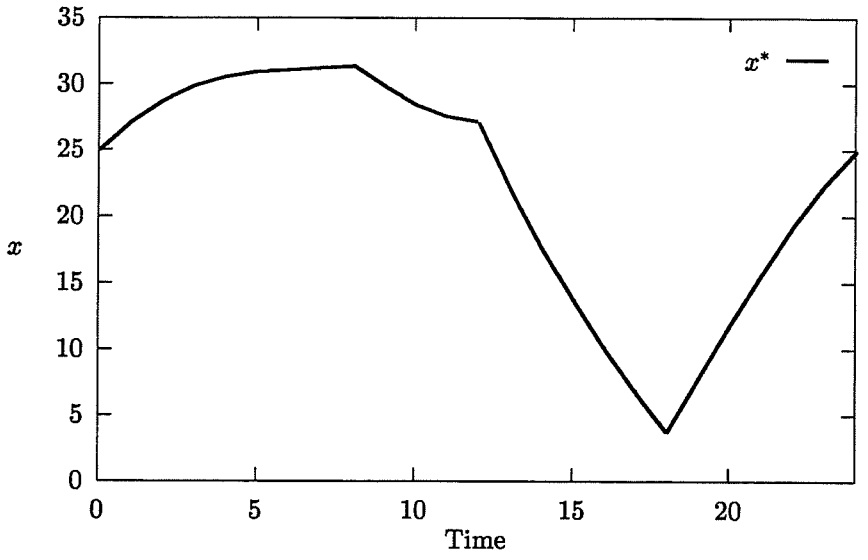


Figure 1.6: The optimal state

here $u_i^* = \underline{u}_i$. As seen on Figure 1.6 the variation in the storage, $(x_{i+1} - x_i)$, accounts for the difference between demand and production in period i.

If we also permit a certain freedom on the final stage, i.e. only require that $\underline{x}_N \leq x_N \leq \overline{x}_N$, rather than (1.104), then we have as terminal conditions on p_N^*, cf. Proposition 1.4.3, that

- if $\underline{x}_N < x_N^* < \overline{x}_N$ then $p^* = 0$
- if $\underline{x}_N = x_N^*$ then $p^* \geq 0$
- if $\overline{x}_N = x_N^*$ then $p^* \leq 0$

If we further modified the problem by adding the term $r_N(x_N)$ to the criterion (1.99) then the end condition on p^* would be

- if $\underline{x}_N < x_N^* < \overline{x}_N$ then $p^* = \nabla r_N(x_N^*)$
- if $\underline{x}_N = x_N^*$ then $p^* \geq \nabla r_N(x_N^*)$
- if $\overline{x}_N = x_N^*$ then $p^* \leq \nabla r_N(x_N^*)$

Example 1.5.3 *Assume as in Example 1.5.2 except that* $\underline{x}_N = -\infty$, $\overline{x}_N = \infty$, $r_N(x_N) = -5(x_N - 25)^2$. *The solution is now* $u_i^* = 7.03$ *for* $0 \leq i \leq 7$ *and* $18 \leq i \leq 23$, $u_i^* = 5.03$ *for* $8 \leq i \leq 11$ *and* $u_i^* = 0.0$ *for* $12 \leq i \leq 17$. *Also,* $x_{24}^* = 23.49$ *and* $p_N = 15.05 = \nabla r_{24}(x_{24}^*)$. □

Example 1.5.4 *Assume as in Example 1.5.3 except that* $\underline{x}_{24} = 25$. *Then* $p_{24} = 15$, $x_{24}^* = 25$ *and* $\nabla r_{24}(x_{24}^*) = 0 < p_{24}^*$. □

In the sequel we continue to use the condition $x_N = \underline{x}_N$.

2. Intermediate state constraints and $\alpha = 1$

Now introduce real lower and upper bounds (1.102) on the storage. In this case we can no longer be sure that $p_i^* = p_{i+1}^*$. There may be a discrepancy if (and only if) a state constraint is active:

- if $\underline{x}_i < x_i^* < \overline{x}_i$ then $p_i^* = p_{i+1}^*$
- if $\underline{x}_i = x_i^*$ then $p_i^* \geq p_{i+1}^*$
- if $\overline{x}_i = x_i^*$ then $p_i^* \leq p_{i+1}^*$

Hence we see that the leveling of the marginal costs takes place only as long as no state constraints are active.

Example 1.5.5 *Let us consider the previous example 1.5.2, but now modified such that* $\underline{x}_i = 20$ *and* $\overline{x}_i = 25$ *for* $i = 1, \ldots, 23$. □

In Figures 1.7 - 1.8 we see how the variation of the contents of the storage is interdependent with the variation of the costate vector p_i^*: only one of them can vary at a time, and if p varies with i, then the direction is depending on whether it is the lower or the upper state constraint that is active.

As seen, the redistribution from "expensive" periods to "cheap" periods is partially destroyed, as illustrated on Figure 1.9.

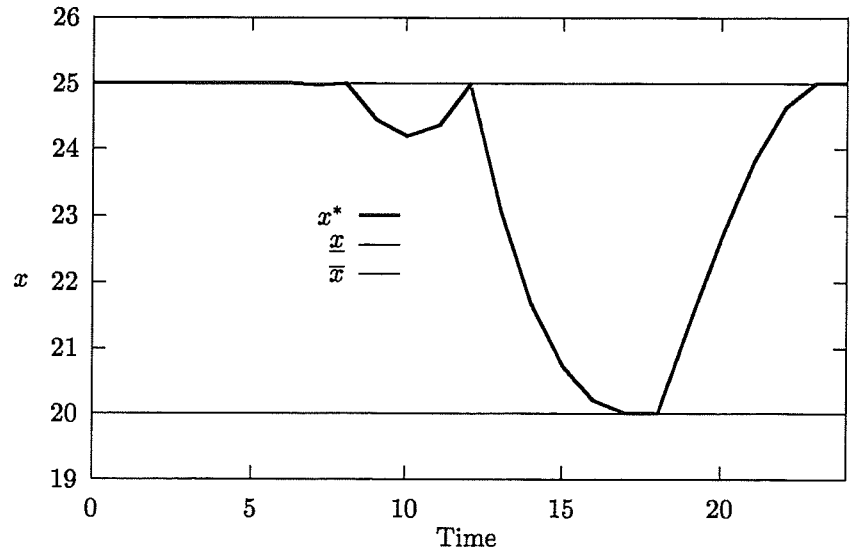


Figure 1.7: The optimal state

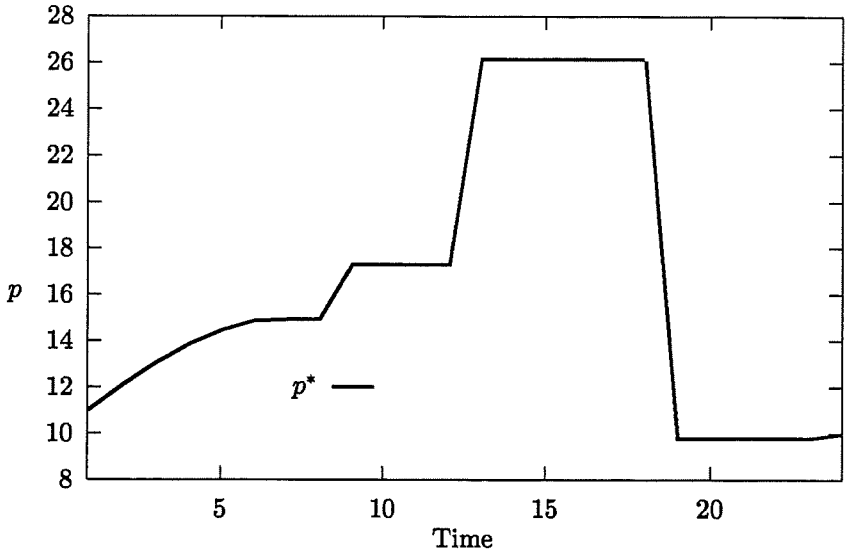


Figure 1.8: The optimal costate

Figure 5

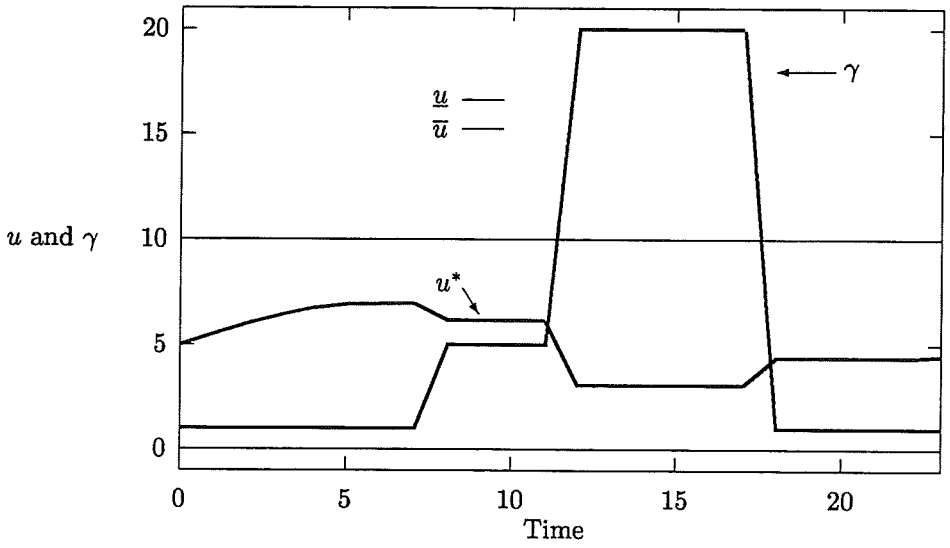


Figure 1.9: γ and the optimal control

3. Proportional loss: $\alpha < 1$

Now introduce a loss proportional with the content of the storage, i.e., $0 < \alpha_i < 1$. With this we get the following modification of the earlier result:

- if $\underline{x}_i < x_i < \overline{x}_i$ then $p_i^* = \alpha_i p_{i+1}^*$
- if $\underline{x}_i = x_i$ then $p_i^* \geq \alpha_i p_{i+1}^*$
- if $\overline{x}_i = x_i$ then $p_i^* \leq \alpha_i p_{i+1}^*$

The relationship between the marginal cost and the costate vector is therefore now:

- if $\underline{u}_i < u_i^* < \overline{u}_i$ then $\nabla r_i(u_i^*) = -p_{i+1}^*$
- if $\underline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \leq -p_{i+1}^*$
- if $\overline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \geq -p_{i+1}^*$

This has implication for the equal marginal gains principle, which can now be modified to

Assume $\underline{x}_i < x_i^* < \overline{x}_i$, $\underline{u}_{i-1} < u_{i-1}^* < \overline{u}_{i-1}$ and $\underline{u}_i < u_i^* < \overline{u}_i$. Then $\nabla r_{i-1}(u_{i-1}^*) = \alpha_i \nabla r_i(u_i^*)$

Example 1.5.6 *Take again the above Example 1.5.2, now modified such that* $\underline{x}_i = 20$, $\overline{x}_i = \infty$, *i=1,...,23, and* $\alpha_i = 0.9$, $i = 0, \ldots, 23$. □

In Figures 1.10 - 1.12 we see the optimal x^*, p^* and u^*.

Since the production in period i is a non-decreasing function of p_{i+1} we can see the implication for the distribution of production between periods. We shall tend to produce less in early periods and more in later periods, because some of the production is lost.

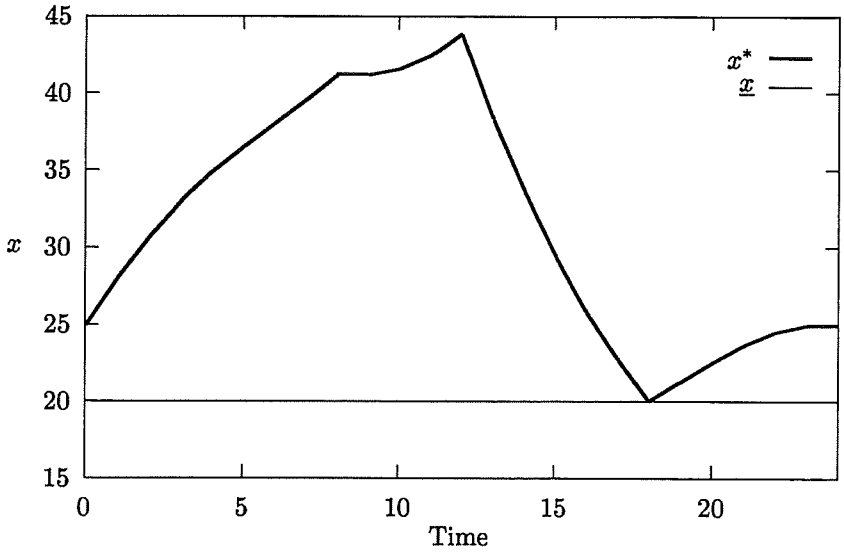


Figure 1.10: The optimal state

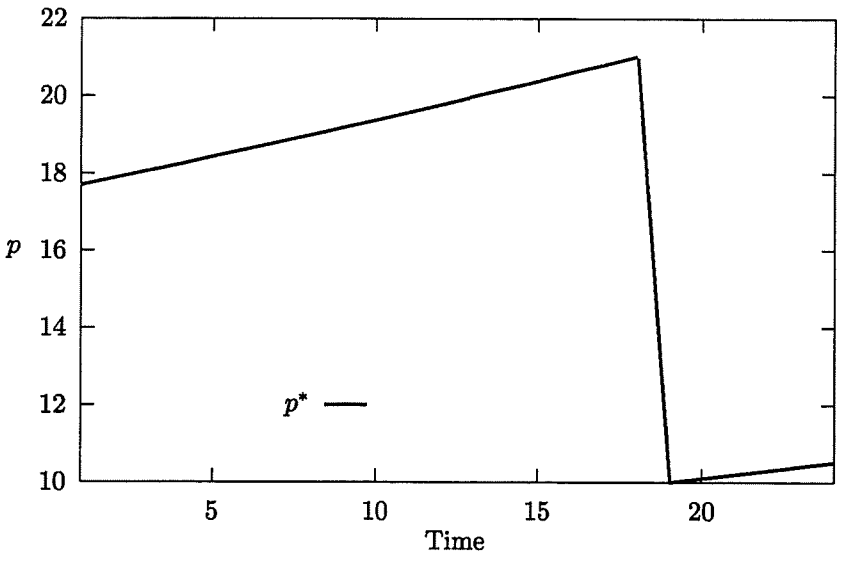


Figure 1.11: The optimal costate

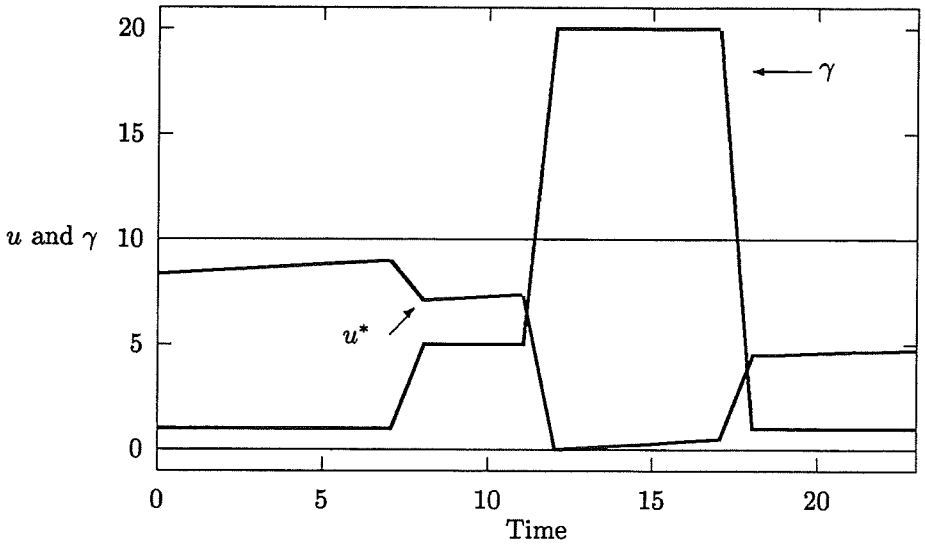


Figure 1.12: γ and the optimal control

4. Planning Horisons.

In a production planning situation as described above we have to make a decision "now", i.e. at stage $i = 0$. For all future stages there is some uncertainty involved. It would therefore be interesting to know how certain the data of the problem has to be, in order that the solution found remains optimal. We consider this in the no-loss situation for expositional simplicity.

> Let a p^o be given, and let $p_i = p^o$ for all i. Calculate then the unique u_i^o for all i according to (1.106) and x_{i+1}^o for all i according to (1.100). Assume the following holds true. There is a stage index i^d and a stage index i^f, with $0 \leq i^d < i^f \leq N$, such that $\underline{x}_i \leq x_i^o \leq \overline{x}_i$ for all $i < i^f$.
>
> Further there holds either (1) $x_i^o = \underline{x}_i$ for $i = i^d + 1$ and $x_i^o \geq \overline{x}_i$ for $i = i^f$, or (2) $x_i^o = \overline{x}_i$ for $i = i^d + 1$ and $x_i^o \leq \underline{x}_i$ for $i = i^f$.
>
> Then u_i^o are optimal for $0 \leq i \leq i^d$ and x_i^o are optimal for $0 \leq i \leq i^d + 1$.

We refer to this result as a planning horizon result. This result operates with two horizons, viz., the forecast horizon and the decision horizon. The forecast horizon i^f is the time period spanned by the indexes i, $0 \leq i \leq i^f$. These are the stage indexes for which we must know the data with certainty. For stages after i^f wee need not know anything at all (and, actually, we need not calculate u_i^o and x_{i+1}^o for these i in order to derive the planning horizon result).

The decision horizon i^d is the time spanned by the indexes i, $0 \leq i \leq i^d$. These are the indexes, for which we know the optimal decisions (controls), irrespective of data beyond the forecast horizon.

In addition to the obvious managerial implications of the planning horizon results, they may be exploited in algorithms as well. See Section 9.4.

5. Two storages.

We next assume a case with two storages, a and b. Production takes place on two production units. From one unit, the production u_i^1 is distributed in fixed proportions to each of the two storages. From the other unit only storage a is supplied by production u_i^2. Total production (including usage of the stored heat) is supposed to fulfill demand d_i. We have the following model:

$$\max[\sum_{i=0}^{N-1}(r_i^1(u_i^1)+r_i^2(u_i^2))] \tag{1.109}$$
$$x_{i+1}^a = \alpha^a x_i^a + \gamma_i^a u_i^1 + u_i^2 - u_i^3 - \beta_i^a \tag{1.110}$$
$$x_{i+1}^b = \alpha^b x_i^b + \gamma_i^b u_i^1 - u_i^4 - \beta_i^b \tag{1.111}$$
$$\underline{u}_i^1 \le u_i^1 \le \overline{u}_i^1 \tag{1.112}$$
$$\underline{u}_i^2 \le u_i^2 \le \overline{u}_i^2 \tag{1.113}$$
$$0 \le u_i^3 \le \kappa_i d_i \tag{1.114}$$
$$0 \le u_i^4 \tag{1.115}$$
$$d_i = u_i^3 + u_i^4 \tag{1.116}$$
$$\underline{x}_i^a \le x_i^a \le \overline{x}_i^a \tag{1.117}$$
$$\underline{x}_i^b \le x_i^b \le \overline{x}_i^b \tag{1.118}$$
$$x_0^a = \underline{x}_0^a \tag{1.119}$$
$$x_0^b = \underline{x}_0^b \tag{1.120}$$
$$x_N^a = \underline{x}_N^a \tag{1.121}$$
$$x_N^b = \underline{x}_N^b \tag{1.122}$$

The model is a simplified representation of a local cogenerating utility system, where u_i^1 represents the production on a gasmotor plus an oil/gas burner and u_i^2 represents the production on a heat pump. The controls u_i^3 and u_i^4 describe the use of heat from storages a and b, respectively.

The model may represent other production planning problems as well.

Most parameters have the same meaning as in the one-storage example above. γ_i^a and γ_i^b are nonnegative coefficients with $\gamma_i^a + \gamma_i^b \le 1$, which represent a distribution of production u_i^1 between the two storages. For simplicity we suppress the time dependency of α i.e. we assume that $\alpha_i^a = \alpha^a$ and $\alpha_i^b = \alpha^b$ for all i.

Storage a is a low temperature storage, supplied mainly from the heat pump; storage b is a high temperature storage, supplied mainly from the oil/gas burner. The demand for heat to e.g. a district heating system is represented by d_i. The restriction (1.114) represents that only a part (given by the fraction κ_i, $0 \le \kappa_i \le 1$) of the demand for heat that can be covered by the low temperature sources.

The criterion r_i^1 is the income from production on unit 1 (income from sale of electrical power to the electrical network minus cost of fuel), and the criterion r_i^2 is the negative of the cost of buying electrical power for the heat pump. We assume r_i^1 and r_i^2 to be strictly concave and differentiable.

With this model we have the Hamiltonian as follows:

$$H_i(x_i,u_i,p_{i+1}) = r_i^1(u_i^1) + r_i^2(u_i^2) \tag{1.123}$$
$$+p_{i+1}^a(\alpha^a x_i^a + \gamma_i^a u_i^1 + u_i^2 - u_i^3 - \beta_i^a) + p_{i+1}^b(\alpha^b x_i^b + \gamma_i^b u_i^1 - u_i^4 - \beta_i^b)$$

It is possible to analyze this as in the previous one-storage example, and arrive at conclusions similar to the previous results. We shall not repeat this. We go ahead to the additional point of interest, which is the dependency between the two storages.

Maximization of the Hamiltonian with respect to u_i is easy as far as u_i^1 and u_i^2 are concerned. These two variables' unique optimal values can be found independently of any other variables. Note that the optimal value of u_i^1 depends on both p_{i+1}^a and p_{i+1}^b since this part of the production is distributed between the two storages.

Now consider the maximization with respect to the two remaining control variables, u_i^3 and u_i^4. These two are interdependent, due to the restriction (1.116). The Hamiltonian is linear with respect to these variables, and so are the restrictions. The optimal solution to the maximization of the Hamiltonian with respect to (u_i^3, u_i^4) will therefore contain an extreme point and may be found by linear programming.

If $p_{i+1}^a \neq p_{i+1}^b$ then the solution is unique; it is seen that $(u_i^3, u_i^4) = (0, d_i)$ if $p_{i+1}^a > p_{i+1}^b$ and $(u_i^3, u_i^4) = (\kappa_i d_i, (1-\kappa_i)d_i)$ if $p_{i+1}^a < p_{i+1}^b$. If $p_{i+1}^a = p_{i+1}^b$ then the solution is not unique, and the solution is the line segment between the two extreme point solutions just given. Such difficulties are often referred to by saying that the Hamiltonian is *singular*, and the solution structure is often referred to as *bang-bang control*, cf. page 111.

If the solution is not unique, there is still an optimal extreme point. However, the solution to the whole problem (1.109) - (1.122) (and not only to the maximization of the Hamiltonian) may very well in this case be a point which is not an extreme point. It is not an exception, but rather the rule, that there will be at least one stage i, such that this is the case. We elaborate now on this observation.

We can interpret the functioning of the variables (u_i^3, u_i^4) as a redistribution between the two storages (although this is in an indirect way, since the two storages are not connected directly but only through (1.116)).

If the optimal solution is in fact unique and an extreme point, we shall say that *a maximal redistribution takes place.* The optimal solution with respect to (u_i^3, u_i^4) has the following interpretation, which is sound in the light of interpretation of the costate vectors as prices:

> If $p_{i+1}^a \neq p_{i+1}^b$ then a maximal redistribution takes place, from the storage with the lower p_{i+1} to the storage with the higher p_{i+1}.

We now assume that $\alpha^a \neq \alpha^b$. From the earlier investigation of the one storage case we derive the following argumentation: If $\underline{x}_i^a < x_i^{a*} < \overline{x}_i^a$ and $\underline{x}_i^b < x_i^{b*} < \overline{x}_i^b$ then $p_i^a = \alpha^a p_{i+1}^a$ and $p_i^b = \alpha^b p_{i+1}^b$. If $\alpha^a \neq \alpha^b$ this implies that either $p_i^a \neq p_i^b$ or $p_{i+1}^a \neq p_{i+1}^b$ (or both). We therefore find:

> Assume $\alpha^a \neq \alpha^b$. If there is a stage sequence, $i^1 \leq i \leq i^2$, such that for these i $\underline{x}_i^a < x_i^{a*} < \overline{x}_i^a$ and $\underline{x}_i^b < x_i^{b*} < \overline{x}_i^b$ then a maximal redistribution takes place in all these periods with at most one exception.

On the other hand: if $p_i^a = p_i^b$ and $p_{i+1}^a = p_{i+1}^b$ then either storage a or storage b is at a boundary. We therefore find:

> Assume $\alpha^a \neq \alpha^b$. If there is a sequence, $i^1 \leq i \leq i^2$, such that for these i a maximal redistribution does not take place then for all i, $i^1 < i \leq i^2$, at least one of the storages is at a boundary, viz.:
>
> - $x_i^a = \underline{x}^a$ if $p_i^a > \alpha^a p_{i+1}^a$

- $x_i^a = \overline{x}^a$ if $p_i^a < \alpha^a p_{i+1}^a$
- $x_i^b = \underline{x}^b$ if $p_i^b > \alpha^b p_{i+1}^b$
- $x_i^b = \overline{x}^b$ if $p_i^b < \alpha^b p_{i+1}^b$

If $\alpha^a = \alpha^b$ then we may have a sequence of i, such that $p_{i+1}^a = p_{i+1}^b$. This can be interpreted to mean that the prices of the contents in the storages are the same for these i. In this case the (u_i^3, u_i^4) that maximizes the Hamiltonian is not unique, illustrating that in this case it may not matter, in which of the two storages the heat is stored.

In summary we see that the optimal solution displays a structure which may be used to gain insight into the qualitative and quantitative nature of the problem.

1.6 Example: Interpretation of Optimality Conditions

In the previous section we analyzed two production planning problems with emphasis on the structure of the optimal solution. In this subsection we look at the same problems and give economic and organizational interpretations of the optimal costate vector p^*. Recall that in the problems we assumed r_i strictly concave and differentiable, and we shall also assume the existence of the unique optimal solution.

One Storage

We first consider the one storage example (1.99) - (1.104). As we saw in the preceding subsection the optimal p_{i+1}^* is intimately related to the *marginal production price* at stage i:

- if $\underline{u}_i < u_i^* < \overline{u}_i$ then $\nabla r_i(u_i^*) = -p_{i+1}^*$
- if $\underline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \leq -p_{i+1}^*$
- if $\overline{u}_i = u_i^*$ then $\nabla r_i(u_i^*) \geq -p_{i+1}^*$

We can also relate p^* to the value of the contents of the storage. An essential assumption for the results that follow is that p^* is unique. Recall from the previous subsection that p^* is supposed to satisfy the adjoint relations.

We let $RUB_i(x_i)$ denote the optimal value in the truncated problem which start at stage i with state x_i and ends at stage N. Thus, RUB_i is the backwards dynamic programming optimal value function. Also this truncated problem may be solved by the maximum principle, yielding p^*. We then have:

> Assume that all p_j^* in the truncated problem are unique. Then RUB_i is continuously differentiable at x_i and $\nabla RUB_i(x_i) = p_i^*$.

We therefore see that p_i^* is the *marginal price of the optimal contents of the storage at stage i, evaluated backwards.*

In this perspective the end condition $p_N = \nabla r_N(x_N)$ (cf. Section 1.3) is naturally interpreted when it is recalled that $RUB_N(x_N) = r_N(x_N)$.

At the initial stage we can use p_0^* to interpret the marginal value of the initial storage $\underline{x}_0$. By definition of RUB_i the value at $\underline{x}_0$ is $RUB_0(\underline{x}_0)$. We then get as before:

> Assume that all p_i^* are unique. Then RUB_0 is continuously differentiable at $\underline{x}_0$ and $\nabla RUB_0(\underline{x}_0) = p_0^*$.

As seen p^*_{i+1} has now two interpretations: as minus the marginal price of production at stage i (if $\underline{u}_i < u^*_i < \overline{u}_i$) and as the marginal value of the storage at stage $(i+1)$ evaluated backwards (if RUB_{i+1} is continuously differentiable at x^*_{i+1}). Clearly these two interpretations should agree at the optimal solution to the problem.

> Assume that RUB_{i+1} is continuously differentiable at x^*_{i+1} and that $\underline{u}_i < u^*_i < \overline{u}_i$. Then $\nabla r_i(u^*_i) + \nabla RUB_{i+1}(x^*_{i+1}) = 0$.

We can also give p^*_{i+1} a third interpretation as an evaluation of the production up to and including stage i. Thus consider this truncated problem, ending at stage $(i+1)$ at state x_{i+1}. Let $UB_{i+1}(x_{i+1})$ denote the optimal criterion value of this truncated problem. Let the truncated problem be solved by the maximum principle, yielding p^*. We then have:

> Assume that all p^*_j in the truncated problem are unique. Then UB_{i+1} is continuously differentiable at x_{i+1} and $\nabla UB_{i+1}(x_{i+1}) = -p^*_{i+1}$.

In other words p^*_{i+1} is the *marginal price of the optimal contents of the storage at stage (i+1), evaluated forwards.*

In particular this evaluates the marginal value of the final storage $\underline{x}_N$:

> Assume that all p^*_i are unique. Then UB_N is continuously differentiable at $\underline{x}_N$ and $\nabla UB_N(\underline{x}_N) = -p^*_N$.

Again we can relate the interpretation of p^*_{i+1} as a marginal production price and the interpretation of p^*_{i+1} as a marginal price of the contents of the storage, evaluated forwards. It is seen by simple argumentation that if $\underline{u}_i < u^*_i < \overline{u}_i$ and if UB_{i+1} is continuously differentiable at x^*_{i+1} then the two interpretations must agree, i.e.

> Assume that UB_{i+1} is continuously differentiable at x^*_{i+1} and that $\underline{u}_i < u^*_i < \overline{u}_i$. Then $\nabla r_i(u^*_i) = \nabla UB_{i+1}(x^*_{i+1})$.

We can also relate the two evaluations of the contents of the storage, viz. the forwards evaluation through UB_i and the backwards evaluation through RUB_i. Also these must agree:

> Consider the two truncated problems joined at x^*_i and assume that for both of these all p^*_j are unique. Then UB_i and RUB_i are continuously differentiable at x^*_i and $\nabla UB_i(x^*_i) + \nabla RUB_i(x^*_i) = 0$.

As seen, the costate vector has strong relationships to the optimal value of partial problems. It might therefore be expected that other aspects of the optimal solution can be interpreted in relation to p^*. This is indeed the case, as we shall now see.

Consider the parameters $\underline{x}_i$, $\overline{x}_i$, $\underline{u}_i$, $\overline{u}_i$, α_i, β_i and d_i. We can interpret the economic significance of these in relation to p^*.

Denote by $R(\delta)^*$, or R^* for short, the optimal criterion value, depending on the parameter δ; here δ represents $\{\underline{x}, \overline{x}, \underline{u}, \overline{u}, \alpha, \beta, d\}$. The marginal benefit from a change in, say, $\underline{x}_i$ is given by $\partial R(\delta)^*/\partial \underline{x}_i$. For the interpretation of $\underline{x}$ and $\overline{x}$ we have:

> Assume that all p^*_i are unique. Then the marginal gain by increasing the intermediate upper state bound $\overline{x}_i$ is $(\alpha_i p^*_{i+1} - p^*_i)$ if $x^*_i = \overline{x}_i$ (zero otherwise) and the marginal gain from a decrease in the lower state bound $\underline{x}_i$ is $(p^*_i - \alpha_i p^*_{i+1})$ if $x^*_i = \underline{x}_i$ (zero otherwise).

Thus we see again that p_i^* can be interpreted as prices. In the Example 1.5.2 above, relating to Figures 1.7 - 1.9, it may be verified that we get the unique values $x_3^* = \overline{x}_3$, $p_3^* = 13.00$, $p_4^* = 13.83$, and hence the gain from a small increase ϵ in $\overline{x}_3$ is approximately 0.83ϵ. A small change in $\underline{x}_3$ will not change $R(\delta)^*$.

Now assume that the intermediate state constraints are not stage dependent, i.e., $\overline{x}_i = \overline{x}$ and $\underline{x}_i = \underline{x}$ for $i = 1, \ldots, N-1$. Then we get the marginal gain from a change in the values $\overline{x}$ and $\underline{x}$ as follows:

> Assume that all p_i^* are unique. Then the marginal gain by increasing the intermediate upper state bound $\overline{x}$ is $\sum(\alpha_i p_{i+1}^* - p_i^*)$, and the marginal gain by decreasing the intermediate lower state bound $\underline{x}$ is $\sum(p_i^* - \alpha_i p_{i+1}^*)$ where the summations is to be taken over the state indexes i for which $x_i^* = \overline{x}_i$ and $x_i^* = \underline{x}_i$, respectively.

The economic evaluation of the initial and final storage is as follows:

> Assume that all p_i^* are unique. Then the marginal gain from an increase in $\underline{x}_0$ is p_0^*, and the marginal gain from an increase in $\underline{x}_N$ is $-p_N^*$.

Often a production planning problem has a cyclic structure, for instance with a cycle of 24 if each period represents one hour of the day. In such problems it often seems reasonable to assume that $\underline{x}_0 = \underline{x}_N$. This raises the question whether an optimal value for this common value can be found. In line with the above results we have the following:

> Assume that all p_i^* are unique. Then the common initial and final value of the storage is optimal if and only if $p_0^* + p_N^* = 0$.

For $\underline{u}_i$ and $\overline{u}_i$ we get:

> Assume that all p_i^* are unique. Then the marginal gain from an increase in $\overline{u}_i$ is $\nabla r_i(u_i^*) + p_{i+1}^*$ if $u_i^* = \overline{u}_i$ (zero otherwise), and the marginal gain from a decrease in $\underline{u}_i$ is $-(\nabla r_i(u_i^*) + p_{i+1}^*)$ if $u_i^* = \underline{u}_i$ (zero otherwise).

The economic significance of the magnitude of α and β can be expressed as follows:

> Assume that all p_i^* are unique. Then the marginal gain from an increase in α_i is $p_{i+1}^* x_i^*$ and the marginal gain from an increase in β_i is $-p_{i+1}^*$.

If for all i $\alpha_i = \alpha$ or $\beta_i = \beta$ we get as a parallel to the case with the intermediate state constraints the marginal gain from an increase in α as $\sum_{i=0}^{N-1} p_{i+1}^*$ and from an increase in β as $-\sum_{i=0}^{N-1} p_{i+1}^*$ if all p_i^* are unique.

For d we find

> Assume that all p_i^* are unique. Then the marginal gain from an increase in d_i is $-p_{i+1}^*$.

Two Storages

Now to the two storage example (1.109) - (1.122). This case admits essentially the same interpretations of p^* as has already been made for the one storage example. The argumentation shall not be repeated, and we shall now concentrate on the interplay between the two storages.

In the one storage example we used the interpretation of p^* as the marginal value of the contents of the storage when we analyzed the redistribution between the storages. This interpretation may be pursued also with two storages.

We assume $\underline{x}_i < \overline{x}_i$ and $\underline{u}_i < \overline{u}_i$. For this, we summarize the sensitivity of the value R^* of the optimal criterion with respect to parameter changes as follows:

Assume that the solution and all p_i^* are unique. Then the marginal gain relative to an increase in a parameter is as follows:

- $\partial R^*/\partial \overline{x}_i^a = \alpha_i^a p_{i+1}^{a*} - p_i^{a*}$ if $x_i^{a*} = \overline{x}_i^a$, zero otherwise
- $\partial R^*/\partial \underline{x}_i^a = p_i^{a*} - \alpha_i^a p_{i+1}^{a*}$ if $x_i^{a*} = \underline{x}_i^a$, zero otherwise
- $\partial R^*/\partial \overline{x}_i^b = \alpha_i^b p_{i+1}^{b*} - p_i^{b*}$ if $x_i^{b*} = \overline{x}_i^b$, zero otherwise
- $\partial R^*/\partial \underline{x}_i^b = p_i^{b*} - \alpha_i^b p_{i+1}^{b*}$ if $x_i^{b*} = \underline{x}_i^b$, zero otherwise
- $\partial R^*/\partial \overline{u}_i^1 = p_{i+1}^{a*}\gamma_i^a + p_{i+1}^{b*}\gamma_i^b + \nabla r^1$ if $u_i^* = \overline{u}_i^1$, zero otherwise
- $\partial R^*/\partial \underline{u}_i^1 = -(p_{i+1}^{a*}\gamma_i^a + p_{i+1}^{b*}\gamma_i^b + \nabla r^1)$ if $u_i^* = \underline{u}_i^1$, zero otherwise
- $\partial R^*/\partial \overline{u}_i^2 = p_{i+1}^{a*}\gamma_i^a + p_{i+1}^{b*}\gamma_i^b + \nabla r^2$ if $u_i^* = \overline{u}_i^2$, zero otherwise
- $\partial R^*/\partial \underline{u}_i^2 = -(p_{i+1}^{a*}\gamma_i^a + p_{i+1}^{b*}\gamma_i^b + \nabla r^2)$ if $u_i^* = \underline{u}_i^2$, zero otherwise
- $\partial R^*/\partial \alpha_i^a = p_{i+1}^* x_i^{a*}$
- $\partial R^*/\partial \alpha_i^b = p_{i+1}^* x_i^{b*}$
- $\partial R^*/\partial \beta_i^a = -p_{i+1}^{a*}$
- $\partial R^*/\partial \beta_i^b = -p_{i+1}^{b*}$
- $\partial R^*/\partial d_i = \begin{cases} -p_{i+1}^{a*} & \text{if } 0 < u_i^{3*} < \kappa_i d_i \\ -p_{i+1}^{b*} & \text{if } 0 < u_i^{4*} \end{cases}$
- $\partial R^*/\partial \kappa_i = -p_{i+1}^{a*} d_i$ if $u_i^{3*} = \kappa_i d_i$, zero otherwise
- $\partial R^*/\partial \gamma^a = (p_{i+1}^{a*} - p_{i+1}^{b*})u_i^{1*}$
- $\partial R^*/\partial \gamma^b = (p_{i+1}^{b*} - p_{i+1}^{a*})u_i^{1*}$

The above interpretations may be justified by reference to Proposition 1.4.7. We have reformulated the results to illuminate the role of the costate vector p. It is seen that virtually all sensitivity results are expressed in terms of p (in combination, of course, with the problem specific functions and parameters)

As p can, in one way, be interpreted in relation to the optimal storage content, forwards and backwards, this shows that the dynamic structure of the problem is also manifest in sensitivity results within each individual stage.

Organizational Interpretations

In the above examples the optimization with respect to the control is performed simultaneously for the whole problem over stages $i = 0$ through $i = N - 1$. However, in certain contexts it is relevant to interpret the decision making as a process distributed over the N stages. Therefore the question arises whether it is possible to make the right decision at stage i without explicit consideration of the decisions at the other stages (the right decision means here the decision that it optimal to the optimal control problem).

As a question of decomposition the two relevant approaches are primal and dual decomposition. In primal decomposition we may assume for the determination of u_i that x_i and x_{i+1} (the "resources") are known. If these two values are optimal with respect to the whole problem then u_i may be selected optimally, see Chapter 5 for more on this.

In dual decomposition we assume for the determination of (x_i, u_i) that p_i and p_{i+1} (the "prices") are known. In contrast to the case of primal decomposition it is not certain that prices exist such that (x_i, u_i) may be selected optimally, see Chapter 8 for more on this.

We may illustrate the decision situation at stage i in dual decomposition for the problem (1.99) - (1.104) as follows. The decision maker for this stage can buy resources carried over from the previous stage (decision maker) at the per unit price p_i; i.e., the expense of buying x_i units will be $p_i x_i$. Similarly, resources can be sold to the next stage (decision maker) at the per unit price p_{i+1}; i.e., the income from selling x_{i+1} units will be $p_{i+1}x_{i+1}$. The negative of the cost of production of u_i units is $r_i(u_i)$.

The decision maker at stage i must therefore consider the problem

$$\max_{x_i,u_i}[r_i(u_i) + p_{i+1}(\alpha_i x_i + u_i - d_i - \beta_i) - p_i x_i] \tag{1.124}$$

which expresses the desire to maximize profits by a combination of trade and production.

If not $\alpha_i p_{i+1} = p_i$ then the decision maker at stage i could make a profit simply by acting as intermediary between decision makers at stages $i-1$ and $i+1$. This trade would be possible if $\underline{x}_i < x_i < \overline{x}_i$ and $\underline{x}_{i+1} < x_{i+1} < \overline{x}_{i+1}$. However, if the last conditions hold then the optimality conditions precisely state that $\alpha_i p_{i+1} = p_i$, cf. above. If therefore the decision maker at stage i is given the prices that are optimal in the problem this implies that no profit can be make at stage i by trading between neighboring stages.

The interpretation may also be carried over to the situation where there are active bounds on x_i and/or x_{i+1}. If for instance $\underline{x}_i = x_i^*$ then $p_i \geq \alpha_i p_{i+1}$ and it could be profitable for the decision maker at stage i to buy resources from stage i; however, this would imply that x_i should be decreased, but this it not possible since $\underline{x}_i = x_i^*$.

For the determination of u_i the decision maker at stage i must again relate the production cost, indicated by r_i, to the alternatives of buying more from the previous stage and/or selling less to the next stage. If $\underline{u}_i < u_i < \overline{u}_i$ then the optimality conditions derived above prescribe that $\nabla r_i(u_i^*) = -p_{i+1}$ such that it it not possible to make a profit by trading with the decision maker at stage $i+1$. Similarly, the optimality implies that $\nabla r_i(u_i^*) = p_i/\alpha_i$ such that it is not possible to make a profit by trading with the decision maker at stage i either. The interpretation is easily extended to the case with a binding constraint $\underline{u}_i$ or $\overline{u}_i$ on the control.

In summary we see that the optimality conditions comply with an organizational interpretation of the problem where decisions are made stagewise, and where the decision makers at the individual stages interact with neighboring decision makers through buying and selling resources at given prices.

1.7 Example: Two Maximum Principle Algorithms

We shall in this section illustrate how the necessary and sufficient optimality conditions of the maximum principle can be used to construct an optimal solution to an optimal control problem. The kind of algorithms we wish to illustrate by this is the kind, where specific properties (besides the control structure) of the problem can be usefully exploited.

Thus we assume a "nice" problem in some way. The problem we here consider is "nice" first because it is of small dimensions ($n = 1$ and $m = 1$), second because it has a quadratic, strictly concave criterion, linear dynamics and a simple bound on the control variable.

The problem considered is

$$\min_x[\sum_{i=1}^{N} w_i(x_i - y_i)^2] \tag{1.125}$$

$$x_i \leq x_{i+1} \tag{1.126}$$

Here $x_i \in R$ are optimization variables and $w_{i+1} \in R$ and $y_{i+1} \in R$ are given parameters, with $w_{i+1} > 0$. The problem is known as the isotonic regression problem with respect to a complete ordering. The problem arises in e.g. statistics, production planning and inventory control, cf. Barlow and Bruuk (1972), Vidal (1994a). Although simple in structure, the problem is not trivial to solve, and it has attracted considerable attention. Several algorithms for solving the isotonic regression problem were discussed in Best and Chakravarti (1990). The computational complexities of these algorithms range from $O(N^4)$ to $O(N)$.

The problem (1.125) - (1.126) can be reformulated to an optimal control problem as

$$\max_{x,u}[-\sum_{i=0}^{N-1} w_{i+1}(x_i + u_i - y_{i+1})^2] \tag{1.127}$$

$$x_{i+1} = x_i + u_i \tag{1.128}$$

$$u_i \geq 0 \tag{1.129}$$

$$x_0 = \underline{x}_0 \tag{1.130}$$

Here $\underline{x}_0$ is arbitrary and (1.129) holds for $i = 1, \ldots, N-1$, while u_0 is unconstrained. Alternatively, $\underline{x}_0$ is chosen sufficiently small ($\underline{x}_0 \leq \min_i\{y_i\}$ is seen to be sufficient) and (1.129) holds for $i = 0, \ldots, N-1$. We shall here choose the latter option.

Optimality Conditions

We get the Hamiltonian

$$H_i(x_i, u_i, p_{i+1}) = -w_{i+1}(x_i + u_i - y_{i+1})^2 + p_{i+1}(x_i + u_i) \tag{1.131}$$

According to the maximum principle, the Hamiltonian is to be maximized by u_i, subject to $u_i \geq 0$. It is straightforward to find the unique u_i^* as

$$u_i^* = \begin{cases} y_{i+1} - x_i^* + p_{i+1}/(2w_{i+1}) & \text{if this is positive} \\ 0 & \text{otherwise} \end{cases} \tag{1.132}$$

The optimality condition (1.132) and the restriction (1.129) imply for $i = 0, \ldots, N-1$ that at the optimum there holds $\nabla_u H_i(x_i, u_i, p_{i+1}) \leq 0$ i.e.,

$$-2w_{i+1}(x_i^* + u_i^* - y_{i+1}) + p_{i+1}^* \leq 0 \tag{1.133}$$

The adjoint equation of the maximum principle gives us that for $i = 0, \ldots, N$

$$p_i^* = \nabla_x H_i(x_i^*, u_i^*, p_{i+1}^*) \tag{1.134}$$

Using (1.131) this is formulated as:

$$-2w_{i+1}(x_i^* + u_i^* - y_{i+1}) + p_{i+1}^* = p_i^* \tag{1.135}$$

From (1.133) and (1.135) we immediately have that for $i = 0, \ldots, N$:

$$p_i^* \leq 0 \tag{1.136}$$

From (1.132) and (1.133) we see that if $u_i^* > 0$ then $\nabla_u H_i(x_i^*, u_i^*, p_{i+1}^*) = p_i^* = 0$. On the other hand, we see from (1.132) that if $p_i^* = \nabla_u H_i(x_i^*, u_i^*, p_{i+1}^*) < 0$ then $u_i^* = 0$. Hence we have the following complementarity condition for $i = 0, \ldots, N-1$

$$p_i^* u_i^* = 0 \tag{1.137}$$

Since x_N is free we have the end condition

$$p_N^* = 0 \tag{1.138}$$

The criterion (1.127) is concave, (1.128) and (1.129) are linear, and therefore (1.128) - (1.138) are necessary as well as sufficient conditions for optimality, Proposition 1.3.5. As the criterion (1.125) is strictly concave and has an unconstrained maximum the optimal solution exists and is unique.

The problem illustrates the *two-point boundary value problem*: find the optimal solution that links the boundary values (x_0, p_0) and (x_N, p_N). The boundary conditions at $i = N$ are that either a fixed endpoint $\underline{x}_N$ is given and then p_N^* is free, or x_N^* is free, and then $p_N^* = 0$ (here we have the latter conditions). At $i = 0$, we have $x_0 = \underline{x}_0$ and p_0^* is free.

We have now investigated the optimality conditions. Let us see how we can use them to construct the optimal solution.

The conditions at the two boundaries indicate that we could work either in a forwards or a backwards direction. If we work in a forwards direction, we can start at x_0, which is known, and then, for given u_i, work forwards to calculate x_{i+1}, using the dynamic equation. If we work in the backwards direction, we can start at p_N^*, which is known to be zero, and then, for given x_i and u_i work backwards to calculate p_i, using the adjoint equation (1.135). We shall illustrate both procedures, starting in the forwards direction.

A Forwards Algorithm

Let the integers s and t be given and consider a sequence given by indexes i, $0 \leq s+1 \leq i \leq t \leq N-1$. Let x_{s+1} and p_{s+1} be given and assume that the corresponding sequences u_i, x_{i+1}, p_{i+1} satisfy (1.128), (1.132), (1.135) and (1.137) with $p_{i+1} < 0$. This implies by (1.137) that $u_{i+1} = 0$.

We change now x_{s+1} and/or p_{s+2} and consequently change all later u_i, x_{i+1} and p_{i+2}, $s+1 \leq i \leq t$, in such a way that (1.129), (1.130), (1.132) and (1.137) remain fulfilled. Considering the expressions involved we see that this in fact possible. Moreover we see the following result:

> Under the assumptions given p_{i+1} is a linear increasing function of p_{s+2} and x_{s+1}, as long as $p_{i+1} < 0$, $s \leq i \leq t$

Next consider a situation where we have a given x_s, $p_s = 0$ and $p_{s+1} < 0$. From (1.133) - (1.135) we see that $p_s = 0$ implies $\nabla_u H_s = 0$. Therefore we see from (1.132) that u_s is a linear increasing function of p_{s+1}, provided p_{s+1} is increased. From (1.128) we see that then x_{s+1} is linearly increasing with p_{s+1}, and from (1.135) we see that also p_{s+1} is linearly increasing with p_{s+1}. We therefore have:

> Under the assumptions given, p_{t+1} is a linearly increasing function of p_{s+1} as long as $p_{i+1} < 0$, $s \leq i \leq t$, provided p_{s+1} is increased.

Going through the algebra it may be verified that under the same assumptions as in the two results above we have for $i = s+2, \ldots, N$, with δp_i denoting increases:

$$\delta p_i = \delta p_{i-1} + w_i(\delta p_{s+1}/w_{s+1}) \tag{1.139}$$

We can now state the algorithm *Ostrava*$_2$ to solve the problem. We initialize by letting $u_i = 0$, $x_i = \underline{x}_0$ for $i = 0, \ldots, N-1$. Then we select p_i, $i = 1, \ldots, N$, consistent with (1.135) - (1.137), and such that for at least one index j we have $p_j = 0$.

This can be done by choosing a value $\underline{x}_0$ strictly smaller than $\min_i\{y_i\}$, which according to (1.135) - (1.137) forces $p_0 = 0$, and then update p_i, $i = 1, \ldots, N$, using (1.135).

Then we have the main body of the algorithm:

Step 1 Find the largest index s, such that $p_s = 0$. If $s = N$ then stop, else go to Step 2.

Step 2 Find using (1.139) the required increase δp_{s+1} in order to get $p_j = 0$ for an index j, $s+1 \leq j \leq N$, while $p_i \leq 0$ for $s+1 \leq i \leq N$. This can be achieved with the following procedure: For a unit increase $\delta p_{s+1} = 1$ calculate δp_i, $i = s+2, \ldots, N$, using (1.139). Set j to the index i for which $\delta p_i / p_i$ is minimized. Then increase p_i by $\delta p_i (p_j / \delta p_j)$, $i = s+1, \ldots, \ldots, N$ (p_j thus becomes zero as intended).

Step 3 Update recursively forwards u_i and x_{i+1}, $i = s, \ldots, N$ according to (1.132) and (1.128), respectively. (The values u_i, x_{i+1}, p_{i+1} are left unchanged for $0 \leq i \leq s-1$.) Go to Step 1.

The algorithm will terminate with the unique optimal solution in a finite number of arithmetic operations. This happens when $p_N = 0$. In the worst case, which is encountered when s is incremented by only one in Step 1 ($\{x_i\}$ is a strictly increasing sequence), we may have to make a total of $\frac{1}{2}N^2$ comparisons in Step 1 and a number of comparisons proportional to $\frac{1}{2}N^2$ in Step 2. The initialization of the algorithm can be seen to increase only linearly with N. The computational complexity of the algorithm described is therefore $O(N^2)$.

It is interesting to observe that we find the optimal solution in a forward way. That is, we may know the optimal solutions u_i^*, $0 \leq i \leq s-1 < N$ without knowing the optimal solution for the remaining stages i. This is an example of a planning horizon, cf. Section 1.5. Moreover, if the algorithm is stopped at the end of Step 3 before the complete solution is found, the values (x, u) at that time of computation constitute a feasible solution.

A Backwards Algorithm

In the algorithm above we exploited the fact that x_0 could be assumed known. Then we could work forwards. We may also construct a backwards solution algorithm. In this case we do not know x_N^* but we know that $p_N^* = 0$, cf. (1.138).

Now we describe the backwards algorithm. Initially let $x_i = \underline{x}_0$ and $u_i = 0$ for all i and let $p_N = 0$. Then assume that at a certain stage of the algorithm there is an index s such that $x_s = \underline{x}_0$ and that (1.128), (1.132) and (1.135) - (1.138) are fulfilled for all $i > s+1$, and that $p_{s+1} \leq 0$; for $i \leq s$ there still holds $x_i = \underline{x}_0$ and $u_i = 0$. The assumptions are seen to hold initially with $s = N-1$.

Now consider stage s. Calculate p_s from (1.135) assuming $x_s = \underline{x}_0$ and $u_s = 0$. There are now two possible situations:

A $p_s \leq 0$: According to (1.132) - (1.135) the optimal u_s is $u_s = 0$. Therefore (1.128), (1.132) and (1.135) - (1.138) are fulfilled for $i \geq s$ and we can proceed to stage $s-1$.

B $p_s > 0$: Keep $x_s = \underline{x}_0$ and increase u_s until p_s calculated from (1.135) is 0 and (1.132) is fulfilled for $i = s$ keeping (1.128), (1.132) and (1.135) - (1.138) fulfilled for $i \geq s$.

If this can be continued to $s = 0$ the solution is found because then (1.128), (1.132) and (1.135) - (1.138) are fulfilled for all i.

The analysis can be simplified if we had initially chosen $\underline{x}_0 < \min_i\{y_i\}$. Since we set both p_N and u_{N-1} to zero, then p_{N-1} calculated by (1.135) will be positive. Thereafter p_{N-1} is decreased to zero so that the same situation will occur in the next stage and we will thus always be in situation B.

We now discuss situation B. It can be described alternatively in two steps: (1) keep $u_s = 0$ and increase x_s by δx_s until $p_s = 0$ assuming (1.128), (1.132) and (1.135) - (1.138) fulfilled for $i > s$; (2) let $u_s = \delta x_s$ and let $x_s = \underline{x}_0$. Therefore we now describe how to find the necessary change δx_s in x_s in order to get $p_s = 0$.

The analysis of these changes is facilitated by the following addition to the observations already made:

> u_i as given by (1.132) decreases linearly with increasing x_i such that (x_i+u_i) is constant, as long as $(y_{i+1} - x_i + p_{i+1}/(2w_{i+1})) > 0$, and u_i remains constant otherwise. p_i as given by (1.135) is constant as long as $(y_{i+1} - x_i + p_{i+1}/(2w_{i+1})) > 0$, and p_i is a linearly decreasing function of x_i otherwise.

First assume that for all $i > s$ we have $u_i = 0$. Increase x_s by δx_s and keep (1.128), (1.132) and (1.135) - (1.138) fulfilled. This implies that x_i is increased by δx_s for all $i > s$, cf. (1.128), while all u_i remain zero, cf. (1.132). Keeping $p_N = 0$, cf. (1.138), implies by (1.135) that the change δp_{N-1} in p_{N-1} is $-2\omega_N \delta x_s$, the change δp_{N-2} in p_{N-2} is $-2\delta x_s(\omega_N + \omega_{N-1})$ and continuing this way we see that

$$\delta p_s = -2\delta x_s \sum_{j=s+1}^{N} \omega_j \tag{1.140}$$

Now associate to each index $i > s$ another index $\mathcal{K}(i)$ such that $\kappa = \mathcal{K}(s)$ is the smallest index $s < \kappa < N$ such that $u_\kappa > 0$. $\kappa = \mathcal{K}(s)$ can be regarded as the first element of a linked list of the indexes $i > s$ that have $u_i > 0$. The next element of the list is $\mathcal{K}(\kappa)$. We introduce u_N artificially, and if u_N is chosen sufficiently big ($u_N = \max_i\{y_i\} - \underline{x}_0 + \epsilon, \epsilon > 0$ is seen to be sufficient) then the last element in the list will be N.

To increase x_s by δx_s and keep (1.128), (1.132) and (1.135) - (1.138) fulfilled implies that x_i is changed by δx_s for $s < i \leq \kappa$. This is achieved by increasing u_s by $\delta u_s = \delta x_s$. Now u_κ is changed by $\delta u_\kappa = -\delta x_s$, cf. (1.132), and therefore $\delta x_i = 0$ for $\kappa + 1 \leq i \leq N$, cf. (1.128). Using (1.138) and (1.135) we see that $\delta p_i = 0$ for $\kappa \leq i \leq N$, while $\delta p_{\kappa-1} = -2\omega_\kappa \delta x_s$, $\delta p_{\kappa-2} = -2\delta_s(\omega_\kappa + \omega_{\kappa+1})$, and continuing this way we see that

$$\delta p_s = -2\delta x_s \sum_{j=s+1}^{\kappa} \omega_j \tag{1.141}$$

We can consider (1.140) a special case of (1.141) by letting $\kappa = N$ in (1.141).

Assuming (1.141) holds we can then calculate the δx_s which makes $p_s = 0$:

$$\delta x_s = \frac{p_s}{2\sum_{j=s+1}^{\kappa} \omega_j} \tag{1.142}$$

While (1.140) holds for arbitrary positive δx_s, (1.141) only holds as long as $u_\kappa + \delta u_\kappa \geq 0$. We can therefore actually change δx_s by at most u_κ if (1.141) shall hold. If therefore the change

prescribed in (1.142) is not greater than u_κ there is no problem: we perform the changes and go to stage $s-1$. Otherwise we first change x_s by u_κ and then change the index $\kappa = \mathcal{K}(\kappa)$, so that it equals the new smallest stage index i after stage s for which $u_i > 0$; if $u_i = 0$ for $s < i < N$ then $\kappa = \mathcal{K}(s) = N$ will be equal to N. Now the increase in x_s can be continued. We proceed this way until $p_s = 0$.

In this way, to make $p_s = 0$, we need to change one value u_κ and eventually make one or more u_κ's irreversibly zero, thus eliminating them from further consideration.

We can now formalize this as the algorithm *Ostrava*$_1$ as follows. We initialize by letting $x_i = \underline{x}_0 < \min_i\{y_i\}$, $u_i = 0$ for $i = 0, \ldots, N-1$, $p_N = 0$, and $u_N = \max_i\{y_i\} - \underline{x}_0 + \epsilon, \epsilon > 0$. Let $s = N-1$ and $\kappa = N$. We then have:

Step 1 If $s = -1$ then go to Step 5. Else calculate p_s from (1.135).

Step 2 Use (1.142) to find δx_s. Set $\delta x_s = \max\{\delta x_s, u_\kappa\}$. Increase u_s by δx_s. Decrease u_κ by δx_s. Update p_s using (1.141). If u_κ becomes zero then make $\kappa = \mathcal{K}(k)$.

Step 3 If p_s is still positive then go to Step 2.

Step 4 Make $\mathcal{K}(s) = \kappa$, $\kappa = s$ and $s = s-1$. Go to Step 1

Step 5 Find all x_i using (1.128) and (1.130). Stop.

In Step 2 we have to find out how much x_s shall be increased. This is done by comparing the necessary change δx_s in x_s in order to make (1.136) - (1.137) fulfilled, with the change that will make $u_\kappa = 0$. If u_κ reaches zero then stage κ is no longer to be considered by the algorithm and κ is made to point to the next stage with non-zero u, that is, $\kappa = \mathcal{K}(\kappa)$. If this happens, p_s might still be positive, in which case Step 3 forces repetition of Step 2 , this time with the new κ.

Step 4 updates $\mathcal{K}(s)$ with the current κ and, since u_s will always become positive, sets $\kappa = s$, before proceeding to stage $s-1$.

Note that in Steps 1-4 only the values of u_i, $i = s, \ldots, N$ are kept updated. The sequence x_i, $i = 1, \ldots, N$ is only calculated in Step 5. Likewise, $\mathcal{K}(i)$, $i = s, \ldots, N-1$ need only be up to date for the indexes i that have non-zero u_i.

In this algorithm the solution is found in a backwards way. Unlike the first algorithm we have that if the computations are stopped before the complete solution is found, the value (x, u) at that time of computation does not constitute a feasible solution unless the operation in Step 5 is performed. Optimal u_i^* are not known before the algorithm terminates. The algorithm will terminate in a finite number of arithmetic operations with the unique optimal solution.

Let us finally consider the computational complexity of the algorithm.

The number of operations to be performed in the initialization is proportional to N, including finding $\min_i\{y_i\}$ and $\max_i\{y_i\}$.

To avoid explicitly calculating the summations in (1.141) and (1.142) we define a sequence $\omega^a cc_i, i = 0, \ldots, N$ by setting $\omega^a cc_0 = 0$ and making $\omega^a cc_i = \omega^a cc_{i-1} + w_i$ for $i = 1, \ldots, N$. This takes place in the initialization phase and involves N addition operations. We are then able to calculate $\sum_{j=s+1}^{\kappa} \omega_j = \omega^a cc_\kappa - \omega^a cc_s$ with a single subtraction.

The loop in Steps 1-4 is to be exectuted $N-1$ times and it is evident that all the operations involved represent constant factors in terms of complexity except for the loop in Steps 3-2 imposed by the go-to statement in Step 3. So, without considering Step 3 the algorithm's body complexity is $O(N)$. As explained previously, the go-to statement in Step 3 only occurs when, in Step 2, a value $u_k, k > s$ reaches zero and is subsequently and irreversibly taken out of the list defined by $\mathcal{K}(\kappa)$, otherwise p_s will be zero. This means that the go-to statement in Step 3 could be regarded as being associated with index κ and therefore occurs at most once per each possible κ during

the entire algorithm execution. It follows that Steps 2-3 are repeated at most $N-2$ extra times altogether, and therefore the loop in Steps 1-4 maintains complexity $O(N)$. Finally, Step 5 needs N operations.

We see that the computational complexity of the algorithm is $O(N)$.

Numerical Experiments

In this section we give an experimental illustration of the running times of the two algorithms described above and represent graphically examples of solutions to the problem. For this purpose a few test problems were generated, with the following characteristics:

$$N = 200$$

$$y_1 = 2;$$

$$w_i \in [1,2], i = 1,\ldots,N$$

$$y_i \in [y_{i-1} - AF, y_{i-1} + A(1-F)[, i = 2,\ldots,N; F \in [0,1]$$

The weights w_i and the values y_i for each problem are produced by a uniform pseudo-random generator. The parameter A determines the interval amplitude when generating y_i while the parameter F controls the probability of $y_i > y_{i-1}$. If F is equal to one then the sequence y_i is decreasing, while if F is zero then y_i will be an increasing sequence.

In one test problem all the values y_i are equal to 2 because A was set to zero. In the other sets $A = 1$ while F was assigned several different values, as shown in the legend included in Figure 1.13.

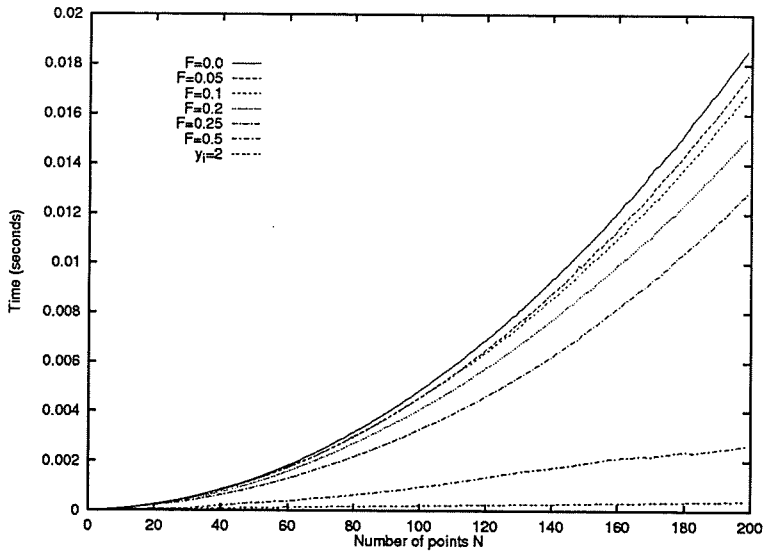


Figure 1.13: Time to solve isotonic regression - Forwards algorithm

Figure 1.13 presents the time to solve a few test problems plotted against the number of stages considered at each run. This chart seems to indicate a quadratic growth. The running times

observed for the backwards algorithm are shown in Figure 1.14 and seem to grow linearly with N. We thus confirm the theorectical analysis made previously to the effect that the forwards and backwards algorithms have computational complexities $O(N^2)$ and $O(N)$, respectively.

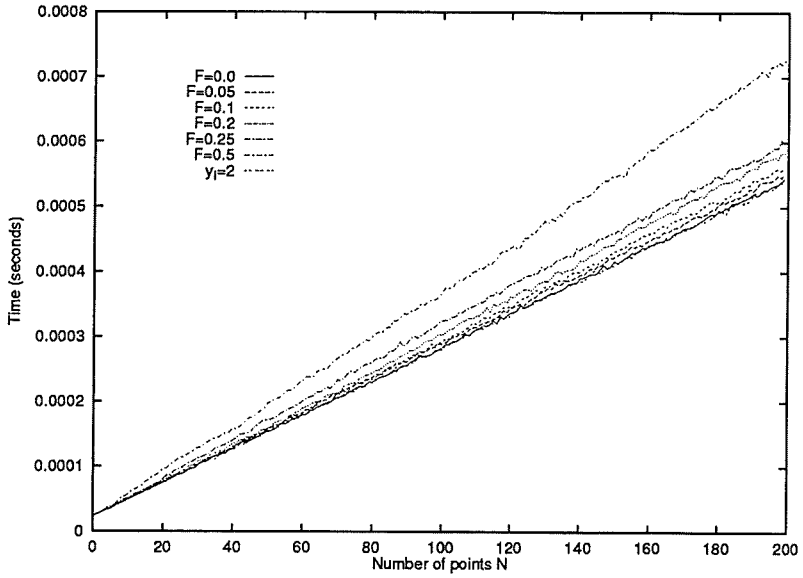


Figure 1.14: Time to solve isotonic regression - Backwards algorithm

Finally we reproduce graphical illustrations of two problems and their respective unique solutions (Figure 1.15). These examples correspond to the first stages of test problems were $F = 0.25$ and $F = 0.4$. The variables x_i are represented joined by a line for easier identification of their sequence.

In summary we see that application of the optimality conditions of the discrete time maximum principle to a specific problem in a natural and straightforward way permits the construction of algorithms that are transparent and which also have attractive computational complexities.

1.8 Conclusions

In this chapter we have presented the discrete time optimal control problem and by examples demonstrated that this problem provides a natural formulation for certain practical problems, viz., those optimization problems that are "staged" in a certain sense.

We have further by examples shown that the analysis of the structure of the optimal solution, as well as the interpretation of it, are naturally presented for this problem type. Moreover, examples have shown that exploitation of optimality conditions may further lead to attractive algorithms for solution of specific problems.

We have also pointed out relationships between the discrete time optimal control problem, and the continuous time optimal control problem on the one hand and mathematical programming on the other hand.

In the sequel we develop further aspects of this.

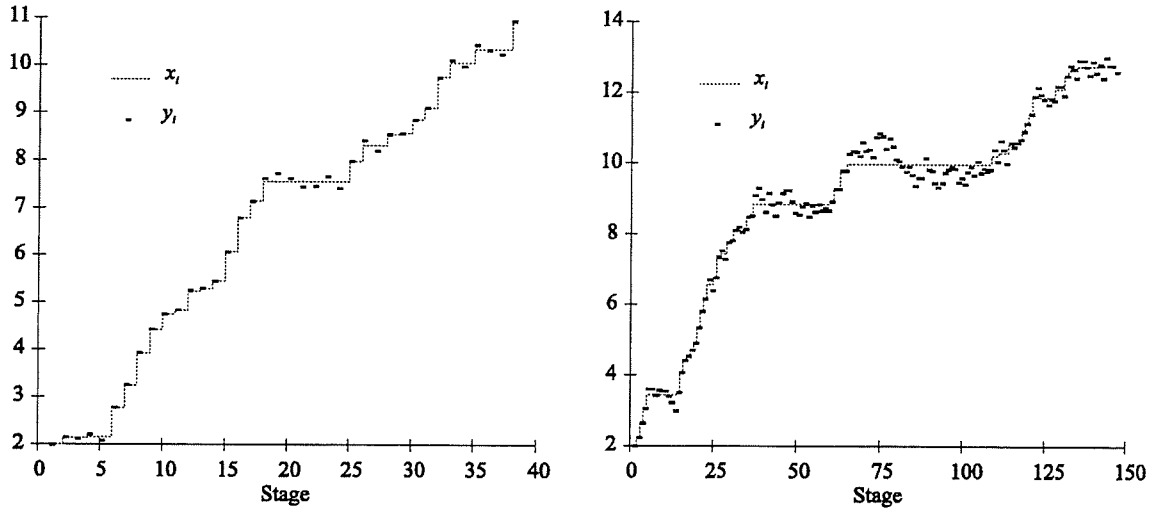


Figure 1.15: First stages of test problems with $F = 0.25$ and $F = 0.4$

Chapter 2

Upper Boundaries

Properties such as smoothness of the optimal value function are essential characteristics of an optimization problem, cf. Section 1.4. Smoothness, for instance, is the prerequisite for derivation of a (unique) marginal value or shadow price, a classical sensitivity result. Necessary and sufficient optimality conditions such as the Karush-Kuhn-Tucker (KKT) conditions are intimately linked to this. This obviously also hold true for the optimal control problem, OCP.

Clarification of these aspects took place around 1980 in relation to the nonlinear mathematical programming problem, see e.g. Fiacco (1976), Gauvin and Tolle (1977), Gauvin (1980), Gauvin and Debeau (1982), Rockafellar (1982, 1983, 1984), Clarke (1983), Fiacco and Kyparisis (1986). In relation to the classical sensitivity result the generalizations were in terms of non-smooth functions in the problem definition, non-unique solutions, non-unique Lagrange multipliers and non-convex problems.

Due to the stagewise character of the OCP it is possible to consider a subsequence of the stages, and still a problem with meaning is retained. Intuitively, the optimal solution to the thus truncated problem must have some strong relationships to the optimal solution of the original problem.

The best known exploitation of this observation is found in the principle of optimality which applies a systematic embedding of optimal control problems within each other in order to express optimality conditions. This in turn suggests a solutions procedure, viz., dynamic programming.

Dynamic programming works with what will here be called greater upper boundaries. These are also known as cost-to-go (working backwards through the stages) or cost-to-arrive (working forwards) functions. In the mathematical programming tradition these functions would be called optimal value or perturbation functions. They represent the optimal criterion function value of a truncated problem parameterized by initial or end point conditions. In addition to the greater upper boundaries we may consider the smaller upper boundaries. These are local, i.e. involve only two adjacent stages, in contrast to the greater upper boundaries, which involve a sequence of stages.

A major difference between the stagewise approach and an approach where the problem is treated without reference to the stagewise structure (as typically in a mathematical programming approach) is that the stagewise approach requires stronger assumptions. Thus for instance the value function of the whole problem, expressed e.g. in terms of the parameter x_N, may very well be smooth even if the upper boundaries for stages $i = 1, \ldots, N-1$ along the optimal trajectory are not smooth for all stages i. If the stagewise analysis therefore requires smoothness along the optimal trajectory at all stages, then stronger assumptions must be taken.

This observation is essential to the stagewise approach of optimal control theory. The key element in the analysis is the constraint qualifications which are known also in the mathematical programming literature to be essential for stability, differentiability etc. This is discussed extensively in Section 2.5 where also possible remedies in case of no fulfillment of the constraint qualifications are presented.

In this Chapter, we investigate the upper boundaries. We start off with defining in Section 2.1 the main concepts that we will apply. Then in the following sections we derive certain properties of the upper boundaries. First we treat in Section 2.2 the question of existence of an optimal solution, and then the following sections treat in turn upper-semi-continuity (Section 2.3), concavity and continuity (Section 2.4), constraint qualifications (Section 2.5), Lipschitz continuity (Section 2.6), continuous differentiability (Section 2.7) and twice continuous differentiability (Section 2.8).

2.1 Basic Concepts

In this section we present the basic definitions, which shall be used throughout. The problem we consider is the following discrete time optimal control problem (OCP), cf. Section 1.1:

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{2.1}$$

$$x_{i+1} = f_i(x_i, u_i),\ i = 0, \ldots, N-1 \tag{2.2}$$

$$(x_i, u_i) \in V_i,\ i = 0, \ldots, N-1 \tag{2.3}$$

$$x_N \in V_N \tag{2.4}$$

With the set V_i of controls and states admissible at stage i we relate two sets. One is the set of *locally admissible controls* at this stage for a given x_i, defined as

$$U_i(x_i) = \{u_i \mid (x_i, u_i) \in V_i\} \tag{2.5}$$

The other set is the set of *locally admissible states* x_i at stage i, defined as

$$X_i = \{x_i \mid \exists u_i : (x_i, u_i) \in V_i\} \tag{2.6}$$

Example 2.1.1 *Let $n = m = 1$ and let the constraints at stage i be given as $1 \leq u_i$, $u_i + 2x_i \geq 5$, $x_i \leq 3u_i - 8$ and $x_i \leq -(u-3)^2 - (2/3)u + 23/3$. We can then illustrate V_i, X_i and $U_i(x_i)$ as on Figure 2.1.* □

Often the local constraint set V_i is separable in x_i and u_i, i.e. it can be written as $V_i = A_i \times B_i$ where $A_i \subset R^n$ and $B_i \subset R^m$. Then $X_i = A_i$, and $U_i(x_i) = B_i$ for all x_i. In this case we shall therefore often write V_i as $X_i \times U_i$. In particular $X_N = V_N$.

The Forwards Direction

A sequence of controls $\{u_0, u_1, \ldots, u_i\}$ will be called a *strategy* (to stage i). The resulting sequence of states $\{x_0, x_1, x_2, \ldots, x_{i+1}\}$, (depending on x_0), satisfying $x_{j+1} = f_j(x_j, u_j)$ for $j = 0, \ldots, i$, will be called a *trajectory* (to stage $i+1$).

A strategy such that all its elements together with the corresponding trajectory satisfy $(x_j, u_j) \in V_j$ and $x_{j+1} = f_j(x_j, u_j)$, $j = 0, \ldots, i-1$, will be called an *admissible strategy and trajectory* (to stage i). Sometimes we shall also call this a *feasible solution* (to stage i).

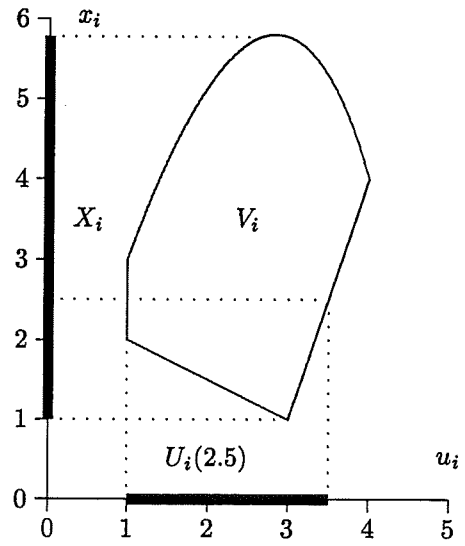


Figure 2.1: Illustration of V_i, X_i and $U_i(x_i)$

Define the set $Y_0 = X_0$. Varying x_i in $X_i \cap Y_i$ and varying additionally u_i in $U_i(x_i)$ we define recursively forwards for $i = 1, \ldots, N$ the *set Y_i of states which are reachable from an $x_0 \in X_0$ along admissible strategies and trajectories*:

$$Y_{i+1} = \{x_{i+1} \mid x_i \in Y_i, (x_i, u_i) \in V_i, x_{i+1} = f_i(x_i, u_i)\} \tag{2.7}$$

We define for $i = 0, \ldots, N-1$ the set of *locally reachable states* Z_i^{i+1} as

$$Z_i^{i+1} = \{x_{i+1} \in R^n \mid \exists (x_i, u_i) \in V_i : x_{i+1} = f_i(x_i, u_i)\} \tag{2.8}$$

Note that $Y_{i+1} \subset Z_i^{i+1}$.

Let us define the *extended state* $\hat{x}_i = (x_i^0, x_i^1, \ldots, x_i^n)' \in R^{n+1}$ where $\hat{x}_0^0 = 0$, and $\hat{x}_i$ is defined recursively forwards for $i = 1, \ldots, N$ along admissible strategies and trajectories as

$$\hat{x}_{i+1} = \begin{pmatrix} \hat{x}_{i+1}^0 \\ x_{i+1} \end{pmatrix} = \begin{pmatrix} \hat{x}_i^0 + r_i(x_i, u_i) \\ f_i(x_i, u_i) \end{pmatrix} \tag{2.9}$$

We may interpret $\hat{x}_{i+1}^0$ as the *forwards accumulated criterion* to stage i, since $\hat{x}_{i+1}^0 = \sum_{j=0}^{i} r_j(x_j, u_j)$ for a given strategy and trajectory.

Now similarly we can define recursively forwards the *set of extended states* which are reachable at the stage $(i+1)$ from $x_0 \in X_0$ along admissible strategies and trajectories. We let $\hat{Y}_0 = 0 \times X_0$ and then

$$\begin{aligned} &\hat{Y}_{i+1} = \\ &\{\hat{x}_{i+1} \mid \exists \hat{x}_i \in \hat{Y}_i, (x_i, u_i) \in V_i,\ x_{i+1} = f_i(x_i, u_i),\ \hat{x}_{i+1}^0 = \hat{x}_i^0 + r_i(x_i, u_i)\} \end{aligned} \tag{2.10}$$

On the set Y_i we define the function $UB_i : Y_i \to R$ for $i = 0, \ldots, N$ as

$$UB_i(x_i) = \begin{cases} 0 & \text{for } i = 0 \\ \sup[\sum_{j=0}^{i-1} r_j(x_j, u_j)] & \text{for } 0 < i < N \\ \sup[\sum_{j=0}^{N-1} r_j(x_j, u_j) + r_N(x_N)] & \text{for } i = N \end{cases} \tag{2.11}$$

over all admissible strategies and trajectories to stage i and state x_i. See Figure 2.2.

In this work we shall most of the time assume that the supremum in maximization problems like in the definition of UB_i in (2.11) is actually attained. In these cases we may write "max" in these expressions rather than "sup":

$$UB_i(x_i) = \begin{cases} 0 & \text{for } i = 0 \\ \max[\sum_{j=0}^{i-1} r_j(x_j, u_j)] & \text{for } 0 < i < N \\ \max[\sum_{j=0}^{N-1} r_j(x_j, u_j) + r_N(x_N)] & \text{for } i = N \end{cases} \tag{2.12}$$

The value $UB_i(x_i)$ is the *forwards (greater) upper boundary* of the partial criterion $\sum_{j=0}^{i-1} r_j(x_j, u_j)$ over all admissible strategies and trajectories leading to x_i, or the *criterion-to-come* to x_i (the *cost-to-come* in an minimization problem). Therefore in the space of extended states $\hat{x}_i$ the point $(UB_i(x_i), x_i')'$ belongs to the upper boundary (in the direction of $\hat{x}_i^0$) of the set $\hat{Y}_i$ of reachable extended states at stage i, whenever $x_i \in Y_i$. We denote by $\{UB_i\}$ the upper boundary at stage i, conceived as a set of points in R^{n+1}, and by UB_i the upper boundary at stage i, conceived as a function $R^n \to R$.

If we in particular define R^* as the optimal criterion value in the problem (2.1) - (2.4) we have $R^* = UB_N(x_N^*)$. Similarly $UB_i(x_i)$ is the optimal value of the *truncated problem* starting at stage 0 and ending at stage i with state x_i.

The Backwards Direction

Similar constructions can be made in the backwards direction, starting from stage N. A sequence of controls $\{u_i, u_{i+1}, \ldots, u_{N-1}\}$ will be called a *terminating strategy* (from stage i). The resulting sequence of states $\{x_i, x_{i+1}, \ldots, x_N\}$, (depending on x_i), satisfying the dynamic equation $x_{j+1} = f_j(x_j, u_j)$ for $j = i, \ldots, N-1$ will be called a *terminating trajectory* (from stage i).

A terminating strategy such that all its elements together with the corresponding trajectory satisfy $(x_j, u_j) \in V_j$, $x_N \in V_N$ and $x_{j+1} = f_j(x_j, u_j)$ for $j = i, \ldots, N-1$, will be called an *admissible terminating strategy and trajectory* (from stage i). For $i = 0$ we shall also call this a *feasible solution* to the OCP.

We define RY_N as $RY_N = V_N$ and then recursively backwards

$$RY_i = \{x_i \mid \exists (x_i, u_i) \in V_i : f_i(x_i, u_i) \in RY_{i+1}\} \tag{2.13}$$

We see that RY_i is the set of locally admissible states from which an $x_N \in V_N$ can be reached along an admissible terminating strategy and trajectory. Observe that $RY_i \subset X_i$.

Similarly the extended states $\check{x}_i \in R^{n+1}$ are defined:

$$\check{x}_i = \begin{pmatrix} \check{x}_i^0 \\ x_i \end{pmatrix} \tag{2.14}$$

with $\check{x}_N^0 = r_N(x_N)$ and $\check{x}_i^0 = r_i(x_i, u_i) + \check{x}_{i+1}^0$ along an admissible terminating strategy and trajectory.

We interpret $\check{x}_i^0$ as the *backwards accumulated criterion* from stage i, since $\check{x}_i^0 = \sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N)$ for a given admissible terminating strategy and trajectory.

The set of extended states at stage i, $\check{RY}_i$, from which the final state $x_N \in V_N$ can be reached along admissible terminating strategies and trajectories can now be recursively defined. Let

$$\check{RY}_N = \{0\} \times V_N \tag{2.15}$$

$$\check{RY}_i = \{\check{x}_i \mid \exists (x_i, u_i) \in V_i : f_i(x_i, u_i) \in RY_{i+1}, \check{x}_i^0 = \check{x}_{i+1}^0 + r_i(x_i, u_i)\} \tag{2.16}$$

On the set V_N we define the real function $RUB_N : V_N \to R$ as $RUB_N(x_N) = r_N(x_N)$. On the sets RY_i we define then the functions $RUB_i : RY_i \to R$ for $i = 0, \ldots, N-1$ as

$$RUB_i(x_i) = \max[\sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N)] \tag{2.17}$$

over all admissible terminating strategies and trajectories from stage i and state x_i. As noted in connection with the definition of UB_i in (2.12) the definition (2.17) assumes that the supremum is actually attained; otherwise the "max" should be substituted by "sup". The function RUB_i is the *backwards (greater) upper boundary*. We note that $R^* = RUB_0(x_0^*) = UB_N(x_N^*)$. Similarly $RUB_i(x_i)$ is the optimal value of the *truncated problem* starting at stage i with state x_i or the *criterion-to-go* from x_i (the *cost-to-go* in an minimization problem).

In the space of extended states $\check{x}_i$ the point $(RUB_i(x_i), x_i')'$ belongs to the upper boundary (in the direction of $\check{x}_{i+1}^0$) of the set $\check{RY}_i$ of extended states at stage i from which a $x_N \in V_N$ can be reached, whenever $x_i \in RY_i$. We denote by $\{RUB_i\}$ the upper boundary at stage i, conceived as a set of points in R^{n+1}, and by RUB_i the upper boundary at stage i, conceived as a function $R^n \to R$.

There is not a complete symmetry between the definitions in the forwards and the backwards directions. This is due to the dynamic equation which is not symmetrical. One implication of this is that in the definitions of Y_i and UB_i only restrictions $(x_j, u_j) \in V_j$ are involved for $0 \le j \le i-1$, while in the definition of RY_i and RUB_i restrictions $(x_j, u_j) \in V_j$ are involved for $i \le j \le N-1$ in addition to $x_N \in V_N$. That is, in the backwards direction the local constraints at stage i are involved, but not in the forwards direction.

The same OCP may be formulated in different ways. As an example, a constraint $x_i \le \overline{x}_i$ may be formulated as $f_{i-1}(x_{i-1}, u_{i-1}) \le \overline{x}_i$. This does not change the optimal solution. However, the sets Y_i, RY_i and W_i^{i+1} and the functions UB_i, RUB_i and ub_i^{i+1} may be changed.

Example 2.1.2 *There is an apparent symmetry between the definitions in the forwards and the backwards directions. One should not think that this also means that the sets and functions defined above will necessarily look alike in all cases.*

As an example of this, consider an integer optimization problem, e.g. the knapsack problem, which in the present context may be formulated as follows: $n = 1$, $m = 1$, $r_i(x_i, u_i) = a_i$, $f_i(x_i, u_i) = x_i + b_i u_i$, $x_0 = 0$, $V_i = \{(x_i, u_i) \mid x_i \in R, u_i \in \{0, 1\}\}$, $x_N \le \underline{x}_N$, *where* a_i *and* b_i *are given constants.*

We find that all Y_i *consist of discrete points, since* x_0 *is fixed and* u_i *can only attain integer values. But all* RY_i *are continuous sets of the form* $\{x_i \in R \mid x_i \le \alpha_i\}$ *for some* $\alpha_i \in R$. *Clearly this may have implications for the solution strategies in the forwards and the backwards directions, respectively.* □

The Smaller Upper Boundaries

Let us for $i = 0, \ldots, N-2$ define the functions $ub_i^{i+1}(.,.) : R^{2n} \to R$ as

$$ub_i^{i+1}(x_i, x_{i+1}) = \max_{u_i}[r_i(x_i, u_i)] \tag{2.18}$$

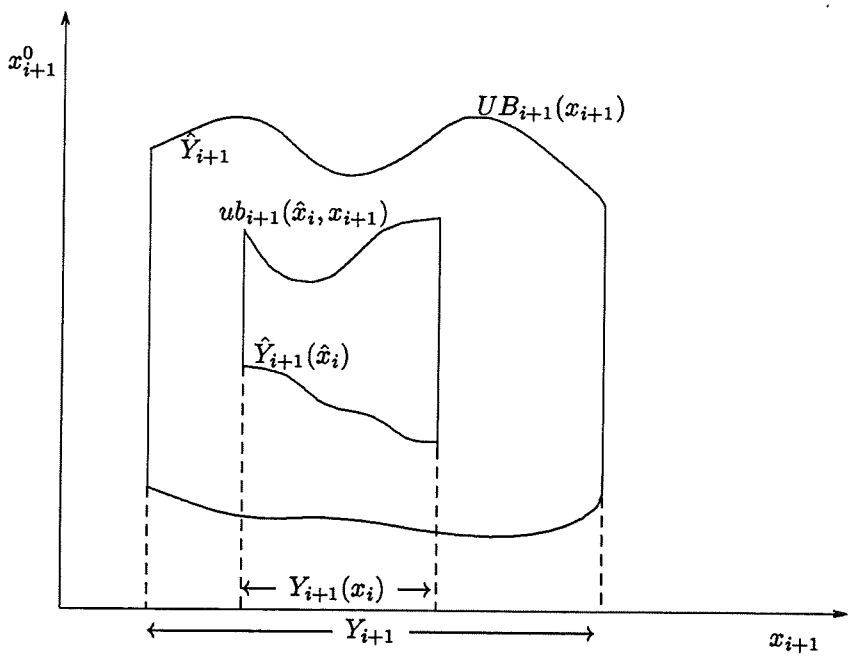


Figure 2.2: The set $Y_{i+1}(x_i)$ of states reachable from a given $x_i \in X_i$, the set Y_{i+1} of reachable states, the set $\hat{Y}_{i+1}(x_i)$ of extended states reachable from a given $\hat{x}_i$, the set Y_{i+1} of reachable extended states, the smaller upper boundary function ub_i^{i+1}, and the greater upper boundary function UB_{i+1}.

over all u_i satisfying $u_i \in U_i(x_i)$ and $f_i(x_i, u_i) = x_{i+1}$. We define $ub^N_{N-1}(.,.)$ as

$$ub^N_{N-1}(x_{N-1}, x_N) = \max_{u_{N-1}}[r_{N-1}(x_{N-1}, u_{N-1}) + r_N(x_N)] \tag{2.19}$$

over all u_{N-1} satisfying $u_{N-1} \in U_{N-1}(x_{N-1})$ and $f_{N-1}(x_{N-1}, u_{N-1}) = x_N$. Again we have to use "sup" rather than "max" if the supremum is not attained, cf. (2.11) - (2.12). We call these functions the *smaller upper boundaries*. A u_i maximizing in (2.18) or (2.19) we shall call an u_i which *defines* ub^{i+1}_i.

For $i = 0, \ldots, N-2$ let W^{i+1}_i be the set $\{(x_i, x_{i+1}) \mid \exists (x_i, u_i) \in V_i : x_{i+1} = f_i(x_i, u_i)\}$. Let W^N_{N-1} be the set $\{(x_{N-1}, x_N) \mid \exists (x_{N-1}, u_{N-1}) \in V_{N-1}, \exists x_N \in V_N : x_N = f_{N-1}(x_{N-1}, u_{N-1})\}$. Then ub^{i+1}_i is defined on W^{i+1}_i. We see that $W^{i+1}_i \subseteq X_i \times Z^{i+1}_i$. We let $W \subseteq R^{(N+1)n}$ be the set of $(x'_0, \ldots, x'_N)'$ for which a feasible solution for the OCP exists. Observe that in general W can not be expressed in a simple way from the W^{i+1}_i's.

We shall often consider ub^{i+1}_i as a function of x_{i+1}, parameterized by x_i. We then write $ub^{i+1}_i(x_i, .)$. We shall also sometimes conceive this as a set of points in R^{n+1}, and then denote it by $\{ub^{i+1}_i(x_i, .)\}$. In order to locate this in relation to $\{UB_{i+1}\}$, we may want to add the value $\hat{x}^0_i$ to $ub^{i+1}_i(x_i, .)$. We then say that $\hat{x}_{i+1}$ belongs to $\{ub^{i+1}_i(x_i, .) + \hat{x}^0_i\}$ if $\hat{x}^0_{i+1} = ub^{i+1}_i(x_i, x_{i+1}) + \hat{x}^0_i$ and $x_{i+1} = f_i(x_i, u_i)$. Similarly $ub^{i+1}_i(., x_{i+1})$ is seen as a function of x_i, parameterized by x_{i+1}.

It is easy to see that the optimal extended states $\hat{x}^*_i$ are located at both the greater upper boundary $\{UB_i\}$ and the smaller upper boundary $\{ub^i_{i-1}(x^*_{i-1}, .) + x^{0^*}_{i-1}\}$ at every stage, i.e., $\hat{x}^*_i \in \{UB_i\}$ and $\hat{x}^*_i \in \{ub^i_{i-1}(x^*_{i-1}, .) + \hat{x}^{0^*}_{i-1}\}$. Otherwise it would be possible to find u_{i-1} in the latter case and a strategy and trajectory $\{(x_j, u_j)\}$ in the former case, which would lead to a greater value of $\hat{x}^0_i$. This is actually Halkin's principle of optimal evolution (Halkin (1964)), cf. Section 3.1.

Finally we might define a reverse smaller upper boundary $rub^{i+1}_i : R^{2n} \to R$ similar to ub^{i+1}_i. However, it would be defined on the same sets W^{i+1}_i as ub^{i+1}_i and also take the same values. There is therefore no need for rub^{i+1}_i.

We have here given the basic concepts of the upper boundary approach. In the sequel we shall investigate the basic properties of the upper boundaries. By this we mean upper-semi-continuity, continuity, Lipschitz continuity, differentiability and concavity.

The analysis will largely rely on similar results from mathematical programming analysis of what is there called the value function. This is here applied in a stagewise manner. As will be seen, the stagewise analysis will permit the same type of results concerning the upper boundaries, except for one thing: the assumptions needed are stronger. The decomposition involved in the stagewise analysis is therefore not straightforward.

2.2 Existence of an Optimal Solution

The following result concerning existence of an optimal solution is fundamental:

Proposition 2.2.1 *Assume that r_i is upper semicontinuous, f_i is continuous, and V_i is compact for $i = 0, \ldots, N-1$ and that V_N is closed. Then, if there is a feasible solution, there is also an optimal solution.*

Proof. This can be seen by reformulating the OCP to a mathematical programming problem as in (1.54) - (1.56) in Section 1.4. Then the result follows from Weierstrass' theorem. However, here we shall prefer to give the proof in a stagewise manner:

The image of a compact set by a continuous function is compact. Therefore, with V_0 compact, we have Y_1 compact, since f_0 is continuous. $(Y_j \times R^m) \cap V_j$ is compact as it is the intersection between a closed and a compact set. By induction it follows that all Y_j are compact, and therefore also Y_N.

Since f_j is continuous the point-to-set map f_j^{-1} mapping Y_{i+1} into $2^{(Y_i \times R^m) \cap V_i}$ is upper semicontinuous because we only consider $x_{j+1} \in Y_{j+1}$ where $f_j^{-1} \neq \emptyset$.

Now consider the following problem defining UB_{j+1}:

$$UB_{j+1}(x_{j+1}) = \sup_{(x_j,u_j)} [r_j(x_j,u_j) + UB_j(x_j)]$$

$$(x_j,u_j) \in V_j$$

$$x_j \in Y_j$$

$$f_j(x_j,u_j) = x_{j+1}$$

where the last condition here can also be written $(x_j,u_j) \in f_j^{-1}(x_{j+1})$. This way of defining UB_{j+1} is equivalent to the definition in (2.11) as may be shown the same way as in Proposition 3.2.1. We consider this problem parameterized by x_{j+1} and the optimization is to take place with respect to (x_j,u_j). The criterion function in this problem is upper semicontinuous, if r_j and UB_j are so. Here r_j is upper semicontinuous by assumption. Therefore the criterion function in this problem is upper semicontinuous if UB_j is so.

UB_{j+1} will be upper semicontinuous if in addition the point to set map f_j^{-1} is upper semicontinuous. As shown above, f_j^{-1} is upper semicontinuous for all j.

Now, $UB_0 \equiv 0$, and therefore UB_0 is upper semicontinuous. By induction we see that UB_j is upper semicontinuous for all j, and therefore UB_N also is.

If $Y_N \cap V_N \neq \emptyset$ there is a feasible solution. An optimal x_N^* is found as an $x_N^* \in V_N \cap Y_N$ where UB_N attains a maximum. By assumption there is a feasible solution, and since UB_N is upper semicontinuous and $V_N \cap Y_N$ compact an x_N^* exists. The admissible strategy and trajectory leading to x_N^* is an optimal solution to OCP. □

The following corollary covers the important class of problems where there is no dependence on the state in the local constraints.

Proposition 2.2.2 *Assume that r_i are upper semicontinuous and f_i continuous. Assume that V_0 is compact and V_N closed. Assume that for $i = 1,\ldots,N-1$ V_i can be written in the separable form $X_i \times U_i$ where X_i are closed and U_i compact. Then, if there is a feasible solution, there is also an optimal solution.*

Proof. From the continuity of f_i and the compactness of U_i it follows that if Y_i is compact then Y_{i+1} is also compact. Since V_0 is compact by assumption, $Y_0 = X_0$ is compact and it follows by forwards induction that all Y_i are compact. Then the result follows as in the above Proposition 2.2.1. □

From now on we shall assume (unless specifically noted) that an optimal solution exists in the problems we shall consider. Therefore we may write "max" rather than "sup".

2.3 Upper-Semi-Continuity

Proposition 2.3.1 *Assume that r_j is upper semicontinuous, f_j is continuous, and V_j is compact for $0 \leq j < i$ (or V_i satisfy the assumptions of Proposition 2.2.2). If $Y_i \neq \emptyset$ then Y_i is compact and UB_i is upper semicontinuous on Y_i.*

Proof. This is shown as in the above Proposition 2.2.1 by considering the truncated problem ending at x_i. □

As seen from the proof of Proposition 2.2.1 the assumptions of the above two propositions ensure that with $x_i \in Y_i$ the truncated problem ending with x_i has an optimal solution.

Interestingly, it is not so easy to derive as strong results for the reverse upper boundary. This is due to the more complicated structure of the parameterization in this case.

Proposition 2.3.2 *Assume that r_j is upper semicontinuous, f_j is continuous, $i \leq j \leq N-1$, and that r_N is upper semicontinuous. Assume that the point-to-set maps $G_j(x_j) = \{u_j \mid (x_j, u_j) \in V_j \wedge f_j(x_j, u_j) \in RY_{j+1}\}$ are upper semicontinuous and non-empty on X_j for $i \leq j \leq N-1$. Then $RY_i \neq \emptyset$ and RUB_i is upper semicontinuous on RY_i.*

Proof. Consider the following problem defining RUB_i:

$$RUB_i(x_i) = \sup_{u_i}[r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i))]$$

$$f_i(x_i, u_i) \in RY_{i+1}$$

$$(x_i, u_i) \in V_i$$

This way of defining RUB_i is equivalent to the definition in (2.17) as may be shown in a way similar to that of Proposition 3.2.1. For this problem r_N is upper semicontinuous by assumption and the point-to-set map $G_{N-1}(x_{N-1})$ is upper semicontinuous and nonempty for all $x_{N-1} \in RY_{N-1}$. This ensures RY_{N-1} to be nonempty and RUB_{N-1} to be upper semicontinuous. Now r_i and RUB_{i+1} are upper semicontinuous and f_i is continuous and therefore $r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i))$ is upper semicontinuous. Therefore the result follows by recursive backwards induction. □

The assumption of upper semicontinuity of the point-to-set map G_j can be fulfilled e.g. in the following ways. Assume that $Z_i^{i+1} \subset RY_{i+1}$ i.e. for all $(x_j, u_j) \in V_j$ there holds $f_j(x_j, u_j) \in RY_{j+1}$. Then the assumption on G_j is fulfilled if the map $G_j^o(x_j) = \{u_j \mid (x_j, u_j) \in V_j\}$ is upper semicontinuous and non-empty. This holds in the important case of separability of the local restrictions on x_j and u_j, i.e. V_j is of the form $V_j = \{(x_j, u_j) \mid x_j \in X_j, u_j \in U_j\}$ where U_j is non-empty. Observe, that also in the forwards direction separability of the local constraints simplifies the analysis, cf. the above Proposition 2.2.2.

The above assumption also holds if e.g. $V_j = \{(x_j, u_j) \mid x_j \in X_j \wedge \underline{u}_j(x_j) \leq g_j^u(u_j) \leq \overline{u}_j(x_j)\}$ where the function $\underline{u}_j : R^n \to R^k$ is lower semicontinuous and the function $\overline{u}_j : R^n \to R^k$ is upper semicontinuous, the function $g_j^u : R^m \to R^k$ is continuous and the set $\{u_j \mid \underline{u}_j(x_j) \leq g_j^u(u_j) \leq \overline{u}_j(x_j)\}$ is nonempty for all $x_j \in X_j$. This in particular holds if e.g. $V_j = \{(x_j, u_j) \mid x_j \in X_j \wedge \underline{u}_j(x_j) \leq u_j \leq \overline{u}_j(x_j)\}$ where the function $\underline{u}_j : R^n \to R^m$ is lower semicontinuous and the function $\overline{u}_j : R^n \to R^m$ is upper semicontinuous and $\underline{u}_j(x_j) \leq \overline{u}_j(x_j)$ for all $x_j \in X_j$.

So we are left with the assumption that $f_j(x_j, u_j) \in RY_{j+1}$. This restriction is recursively defined through (2.15) - (2.16) and really difficult to handle. In many cases it seems impossible to guarantee that it will hold without taking it explicitly into account.

It is seen to hold under the following assumptions: the end point is free ($V_N = R^n$), there are no intermediate state constraints ($V_i = R^n \times U_i$ for $i = 1, \ldots, N-1$), $U_i \neq \emptyset$ and f_i is of the form $A_i x_i + f_i^u(u_i)$ where A_i is an $n \times n$ nonsingular matrix. This implies $RY_i = R^n$ and therefore $Z_i^{i+1} \subset RY_{i+1}$. Although these assumptions are rather strong, they are fulfilled in many OCPs with practical origin.

In summary we see that in general the backwards analysis is more difficult than the forwards analysis.

Discrete dynamic programming is an example of a technique where the implicit constraints $f_i(x_i, u_i) \in RY_{i+1}$ are easily handled, cf. Section 4.1. In this technique typically the whole relevant state space is searched, and a complete enumeration takes place at each stage. If there is no admissible terminating strategy and trajectory from x_i then this will be immediately detected. Similarly in the forwards direction. We discuss this further in Section 4.1.

For the smaller upper boundaries we have:

Proposition 2.3.3 *Assume that r_i is upper semicontinuous, f_i is continuous, and V_i is nonempty and compact. Then W_i^{i+1} is compact and $ub_i^{i+1}(x_i, .)$ is upper semicontinuous. If in addition the point to set map $G_i^{i+1} = \{u_i \mid (x_i, u_i) \in V_i, f_i(x_i, u_i) = x_{i+1}\}$ is upper semicontinuous and nonempty then $ub_i^{i+1}(., .)$ and $ub_i^{i+1}(., x_{i+1})$ are upper semicontinuous.*

Proof. Similar argumentation holds as in the proofs of Propositions 2.3.1 and 2.3.2 above. □

2.4 Concavity (and Continuity)

Let us now turn to concavity.

Proposition 2.4.1 *Assume that r_j is concave, f_j is affine and V_j is convex for $0 \leq j < i$. Then Y_i is convex and UB_i is concave. If in addition all r_j are strictly concave with respect to u_j on $U_j(x_j)$ for any $x_j \in X_j$ and r_0 is strictly concave for all $(x_0, u_0) \in V_0$ then UB_i is strictly concave and the optimal strategy and trajectory leading to $x_i \in X_i$ is unique.*

Proof. Let there be given two admissible strategies and trajectories up to stage i, $\{(u_j^o, x_j^o)\}$ and $\{(u_j^\diamond, x_j^\diamond)\}$. Let $\alpha \in [0, 1]$. Then also the strategy and trajectory $\{(u_j, x_j)\} = \{\alpha(u_j^o, x_j^o)\} + \{(1-\alpha)(u_j^\diamond, x_j^\diamond)\}$ is admissible up to stage i since f_j are affine and V_j are convex. Therefore Y_i is convex.

Now suppose the two strategies and trajectories defined above lead to extended states $\hat{x}_i^o$ and $\hat{x}_i^\diamond$, respectively, that are located at UB_i. Then by the concavity of r_j we have $\sum_{j=0}^{i-1} r_j(x_j, u_j) \geq \sum_{j=0}^{i-1} \alpha r_j(x_j^o, u_j^o) + \sum_{j=0}^{i-1}(1-\alpha) r_j(x_j^\diamond, u_j^\diamond)$, and therefore UB_i is concave.

If r_j are strictly concave as described then $\hat{x}_i^o \neq \hat{x}_i^\diamond$ implies $\sum_{j=0}^{i-1} r_j(x_j, u_j) > \sum_{j=0}^{i-1} \alpha r_j(x_j^o, u_j^o) + \sum_{j=0}^{i-1}(1-\alpha) r_j(x_j^\diamond, u_j^\diamond)$ for $0 < \alpha < 1$ and therefore UB_i is strictly concave.

Now assume that all r_j are strictly concave as described. If $\hat{x}_i^o \neq \hat{x}_i^\diamond$ then the strategies and trajectories leading to $\hat{x}_i^o$ and $\hat{x}_i^\diamond$, respectively, are different. With $0 < \alpha < 1$ and $\{(u_j, x_j)\} = \{\alpha(u_j^o, x_j^o)\} + \{(1-\alpha)(u_j^\diamond, x_j^\diamond)\}$ we have $\sum_{j=0}^{i-1} r_j(x_j, u_j) > \sum_{j=0}^{i-1} \alpha r_j(x_j^o, u_j^o) + \sum_{j=0}^{i-1}(1-\alpha) r_j(x_j^\diamond, u_j^\diamond)$ and therefore UB_i is strictly concave. By the same reasoning we see that if two different strategies were optimal then $\{(u_j, x_j)\} = \{\alpha(u_j^o, x_j^o)\} + \{(1-\alpha)(u_j^o, x_j^o)\}$ would be feasible and give a higher criterion value, contradicting the assumption of optimality. Therefore the solution is unique. □

The situation is slightly more complicated in the backwards direction.

We therefore define that *the dynamics is state sensitive* if $x_i^o \neq x_i^\diamond$, $x_{i+1}^o = f_i(x_i^o, u_i^o)$ and $x_{i+1}^\diamond = f_i(x_i^\diamond, u_i^\diamond)$ implies $(u_i^o, x_{i+1}^o) \neq (u_i^\diamond, x_{i+1}^\diamond)$. Examples of f_i that do not imply state sensitivity are $f_i(x_i, u_i) = u_i$ and $f_i(x_i, u_i) = 0$. On the other hand, the dynamics is state sensitive if e.g. f_i is of the form $A_i x_i + f_i^u(u_i)$ where A_i is an $n \times n$ nonsingular matrix.

Proposition 2.4.2 *Assume that r_j is concave, f_j is affine, and V_j is convex for $i \leq j \leq N-1$ and that r_N is concave and V_N is convex. Then RY_i is convex and RUB_i is concave. If in addition r_j for $i \leq j \leq N-1$ are strictly concave with respect to u_j on $U_j(x_j)$ for any $x_j \in X_j$, and r_N is concave then RUB_i is concave and the terminating strategy and trajectory from $x_i \in RY_i$ is unique. If in addition either (1) r_i is strictly concave for all i or (2) r_N is strictly concave and the dynamics is state sensitive, then RUB_i is strictly concave.*

Proof. The results are proved essentially as in the above Proposition 2.4.1. For the second case of strict concavity of RUB_i it is observed by recursive reasoning that the assumption of state sensitive dynamics assures that if $x_i^o \neq x_i^\diamond$ then either there is an index j, $i+1 \leq j \leq N-1$, such that the two corresponding optimal controls u_j^* are different or the two corresponding optimal states x_N^* are different. Therefore the strict concavity of the criterion with respect to u_j or r_N, respectively, assures the strict concavity of RUB_i. □

Proposition 2.4.3 *Assume that r_i is concave, f_i is affine, and V_i is convex. Then W_i^{i+1} is convex and $ub_i^{i+1}(.,.)$ is concave. If in addition r_i is strictly concave then ub_i^{i+1} is strictly concave and the u_i^* defining $ub_i^{i+1}(x_i, x_{i+1})$ is unique.*

Proof. This is proved as in the above Proposition 2.4.1. □

Concavity of $ub_i^{i+1}(x_i^*, .)$ is an essential assumption if one wants to apply the classical maximum principle. Therefore the next result is of particular interest in connection with this maximum principle.

Proposition 2.4.4 *Assume that r_i is concave with respect to u_i for any fixed x_i, f_i is affine with respect to u_i for any fixed x_i, and $U_i(x_i)$ is convex for any $x_i \in X_i$. Then the set $\{x_{i+1} \mid \exists u_i \in U_i(x_i) : x_{i+1} = f_i(x_i, u_i)\}$ is convex and $ub_i^{i+1}(x_i, .)$ is concave. If in addition r_i is strictly concave with respect to u_i for fixed x_i and the dynamics is state sensitive then $ub_i^{i+1}(x_i, .)$ is strictly concave and the u_i^* defining $ub_i^{i+1}(x_i, x_{i+1})$ is unique.*

Proof. This is proved as in the above Proposition 2.4.1. □

We observe that when the set V_i is defined as $\{(x_i, u_i) \mid g_i(x_i, u_i) \leq 0, h_i(x_i, u_i) = 0\}$ then V_i is convex if all g_i^j are quasiconvex and all h_i^j are both quasiconvex and quasiconcave. In particular, V_i is convex if all g_i^j are convex and all h_i^j are affine.

In the above Propositions 2.4.1, 2.4.2, 2.4.3 and 2.4.4 it suffices that f_i be quasilinear (i.e. both quasiconvex and quasiconcave), cf. the Remark on page 33 after Proposition 1.4.6.

The assumption of *directional convexity* was introduced in the early literature in order to achieve concavity of $ub_i^{i+1}(x_i, .)$, cf. Section 1.3. It can be shown that *the set of extended states $\hat{x}_{i+1}$ reachable from x_i is directionally convex if and only if $ub_i^{i+1}(x_i, .)$ is concave* (cf. Nahorski, Ravn and Vidal (1983) pp. 61 - 63).

Concavity of $ub_i^{i+1}(x_i, .)$ does not alone guarantee the existence of a p_{i+1} such that u_i^* maximizes the Hamiltonian. A constraint qualification is further required. For instance one that secures x_{i+1}^* to be interior to the set of states reachable from x_i^*. We discuss this further below.

Important special cases of problems with concave upper boundaries are the linear and the quadratic-linear problems. Define the linear problem as follows:

$$\max[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N] \tag{2.20}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{2.21}$$

$$G_i^x x_i + G_i^u u_i - \overline{g}_i \leq 0 \tag{2.22}$$

$$H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \tag{2.23}$$

where R_i^x, R_i^u, F_i^x, F_i^u, $\overline{f}_i$, G_i^x, G_i^u, $\overline{g}_i$, H_i^x, H_i^u and $\overline{h}_i$ are matrices of appropriate dimensions.

Proposition 2.4.5 *For the linear problem (2.20) - (2.23) Y_i, RY_i and W_i^{i+1} are convex polyhedra and UB_i, RUB_i and ub_i^{i+1} are polyhedral concave.*

Proof. Consider Y_i. The constraints on $(x_0, u_0, ..., x_i)$ are linear and therefore $(x_0, u_0, ..., x_i)$ are contained in a convex polyhedron. This also applies to x_i and therefore Y_i is a convex polyhedron. The same applies to RY_i and W_i^{i+1}. For the second result see Fiacco and Kyparisis (1986) Propositions 2.12 and 2.13 □

In the quadratic-linear problem the criterion is given as

$$\max[\sum_{i=0}^{N-1} \tfrac{1}{2} x_i' R_i^{xx} x_i + x_i' R_i^{xu} u_i + \tfrac{1}{2} u_i' R_i^{uu} u_i + R_i^x x_i + R_i^u u_i + \tfrac{1}{2} x_N' R_N^{xx} x_N + R_N^x x_N] \tag{2.24}$$

where R_i^{xx}, R_i^{xu}, R_i^{uu}, R_i^x and R_i^u are matrices of appropriate dimensions with R_i^{xx} and R_i^{uu} symmetric. The dynamics and the constraints are linear as in (2.21) - (2.23). We shall refer to this problem as the QLEI problem. If the problem is unconstrained (except for a given initial point $x_0 = \underline{x}_0$) we refer to the problem as the QL problem.

The next Proposition 2.4.6 describes the upper boundaries as being piecewise quadratic along a line segment. This is for UB_i to be understood as follows. Let $a \in R^n$, $\alpha \in R$ and $\beta \in R^n$ and define $x_i = \alpha a + \beta$. Then we can consider UB_i as a function of α. As Y_i is convex and $\alpha a + \beta$ is an affine function of α we will have $x_i \in Y_i$ for α in an interval and therefore for x_i along a line segment. For these α UB_i is piecewice quadratic. Similar ideas apply to RUB_i. For ub_i^{i+1} it applies with $a \in R^{2n}$, $\alpha \in R$, $\beta \in R^{2n}$ and $(x_i', x_{i+1}')' = \alpha a + \beta$.

Proposition 2.4.6 *For the problem (2.21) - (2.24) assume that* $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} \leq 0$ *for $i = 0, \ldots, N-1$ and $R_N^{xx} \leq 0$. Then Y_i, RY_i and W_i^{i+1} are convex polyhedra and UB_i, RUB_i and ub_i^{i+1} are concave.*

If $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} \leq 0$ *for $i = 1, \ldots, N-1$,* $\begin{pmatrix} R_0^{xx} & R_0^{xu} \\ R_0^{xw\prime} & R_0^{uu} \end{pmatrix} < 0$, *$R_N^{xx} \leq 0$ and $R_i^{uu} < 0$ for all i then UB_i is strictly concave and the optimal strategy and trajectory leading to $x_i \in X_i$ is unique.*

If $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} \leq 0$ *for* $i = 0, \ldots, N-1$, $R_N^{xx} \leq 0$ *and* $R_i^{uu} < 0$ *for all* i *then* RUB_i *is concave and the terminating strategy and trajectory from* $x_i \in RY_i$ *is unique.*

If $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} \leq 0$ *for* $i = 0, \ldots, N-1$, $R_N^{xx} \leq 0$ *and* $R_i^{uu} < 0$ *for all* i *and either (1)* $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} < 0$ *for* $i = 0, \ldots, N-1$ *or (2)* $R_N^{xx} < 0$ *and* F_i^x *is nonsingular for all* i *then* RUB_i *is strictly concave.*

If $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} < 0$ *then* ub_i^{i+1} *is strictly concave and the* u_i^* *defining* $ub_i^{i+1}(x_i, x_{i+1})$ *is unique.*

For all the cases above with assumptions implying strict concavity UB_i, RUB_i *and* ub_i^{i+1} *are piecewise quadratic along any line segment for which* $x_i \in Y_i$, $x_i \in RY_i$ *and* $(x_i, x_{i+1}) \in W_i^{i+1}$, *respectively.*

Proof. Y_i, RY_i and W_i^{i+1} are convex polyhedra as shown in Proposition 2.4.5 above. The concavity and strict concavity results follow from Proposition 2.4.1, 2.4.2 and 2.4.3 since the assumptions on r_i taken there are fulfilled with the above assumptions on the matrices.

The result on being piecewise quadratic along a line segment may be seen by considering a solution procedure for the quadratic-linear problem (2.21) - (2.24) which is based on the simplex method, e.g. Wolfe (1959). Consider Y_i and let x_i move along a line segment in Y_i. The unique optimal solution corresponding to x_i is found by performing a parametric analysis in the simplex tableau. It follows from this that the optimal solution is a piecewise linear function of x_i. Each piece corresponds to x_i being in an interval where no basis changes take place. Inserting this piecewise linear function into the quadratic criterion function (2.24) then results in a piecewise quadratic expression and therefore UB_i is piecewise quadratic along this line segment. Similar argumentation applies to RUB_i and ub_i^{i+1}. □

Finally we observe that *concavity of a function implies continuity in the interior of the set, on which the function is defined.* The above assumptions implying concavity of the upper boundaries therefore imply that UB_i is continuous on the interior of Y_i, RUB_i is continuous on the interior of RY_i and ub_i^{i+1} is continuous on the interior of W_i^{i+1}, respectively. A concave function is lower-semi-continuous; it may be discontinuous, but only at the boundary of its domain. It therefore follows that a concave upper-semi-continuous upper boundary is continuous. Under the combined assumptions of Section 2.3 and Section 2.4 the upper boundaries are therefore continuous.

2.5 Constraint Qualifications

Next we turn to Lipschitz continuity and differentiability of the upper boundaries. To analyze this, we need to define and discuss at length constraint qualifications. We first introduce these under the assumption that all functions are Lipschitz continuous and then under differentiability assumptions.

We consider in the remaining part of this chapter the OCP (2.1) - (2.4) with the local constraints (2.3) given on the form

$$g_i(x_i, u_i) \leq 0 \tag{2.25}$$

$$h_i(x_i, u_i) = 0 \tag{2.26}$$

and similarly for (2.4):

$$g_N(x_N) \leq 0 \tag{2.27}$$
$$h_N(x_N) = 0 \tag{2.28}$$

Constraint Qualification for Lipschitz functions

Consider the following problem where $v \in R^a$, $z \in R^b$, $r : R^{a+b} \to R$, $g : R^{a+b} \to R^k$, $h : R^{a+b} \to R^\ell$, and it is assumed that all functions are Lipschitz continuous:

$$ub(z) = \max_v [r(z,v)] \tag{2.29}$$
$$g(z,v) \leq 0 \tag{2.30}$$
$$h(z,v) = 0 \tag{2.31}$$

The OCP may be written in this form when the dynamic equation is reformulated as $f_i(x_i,u_i) - x_{i+1} = 0$ and the parameter z is omitted (i.e. kept constant), cf. also Section 1.4, formulae (1.54) - (1.56) on page 26. The truncated problem ending at x_i may be written in this form by letting $v = (x_0', u_0', \ldots, x_{i-1}', u_{i-1}')'$ and $z = x_i$. The truncated problem starting from x_i may be written in this form by letting $v = (u_i', x_{i+1}', ..., u_{N-1}', x_N')'$ and $z = x_i$. The problem defining ub_i^{i+1} may be written in this form by letting $v = u_i$ and $z = (x_i', x_{i+1}')'$.

We form the Lagrangian $L^\rho(z,v,\rho,\lambda,\mu)$ as follows, where λ and μ are multiplier row vectors, and ρ is a scalar:

$$L^\rho = \rho r(z,v) - \lambda g(z,v) - \mu h(z,v) \tag{2.32}$$

Thus, the usual Lagrangian L may be defined in terms of L^ρ as $L \equiv L^1$.

The following *Lagrange Multiplier Rule* (LMR), formulated using the generalized gradient ∂, holds (cf. Clarke (1983)):

> If v^* solves (2.29) - (2.31) then there exist $(\rho^*, \lambda^*, \mu^*)$, such that
> $0 \in \partial_v L^\rho(z, v^*, \rho^*, \lambda^*, \mu^*)$, $\rho^* \geq 0$, $\lambda^* \geq 0$, $\lambda^* g(z,v^*) = 0$,
> $(\rho^*, \lambda^*, \rho^*) \neq (0,0,0)$

We shall be interested in the case, where we can take $\rho^* = 1$. We say that the problem is *LMR regular*, if there is no solution for (ρ,λ,μ) to the LMR with $\rho^* = 0$. This can also be formulated as follows: the problem is LMR regular if with $\rho^* = 0$ the only solution for (λ^*, μ^*) to $0 \in \partial_v L^\rho(z, v^*, \rho^*, \lambda^*, \mu^*)$, $\lambda^* \geq 0$, $\lambda^* g(z,v^*) = 0$, is $(0,0)$.

The LMR regularity condition is referred to as a *constraint qualification.*

We see that the conditions $\lambda^* \geq 0$, $\lambda^* g(z,v^*) = 0$ is a *complementary slackness condition.* Thus, if $g^j(z,v^*) < 0$ we must have $\lambda^{j*} = 0$; if $\lambda^{j*} > 0$ we must have $g^j(z,v^*) = 0$. We refer to the index set $\{j \mid g^j(z,v^*) = 0\}$ as the set of *active inequality constraints.*

Another concept needed in the following sections is tameness. We say that the problem is *tame* at z, if $ub(z)$ is finite and there exists a compact subset Ω of R^a and an $\alpha > 0$ such that for all $z^\diamond$ with $\| z - z^\diamond \| < \alpha$ for which $ub(z^\diamond) > ub(z) - \alpha$ the problem (2.29) - (2.31) has a solution which lies in Ω (cf. Clarke (1983) p. 241).

Proposition 2.5.1 *Assume that r_i, f_i, g_i and h_i are Lipschitz. Assume that for any truncated problem there is a feasible solution. Then for a truncated OCP ending at $x_i \in Y_i$ the assumption of tameness is fulfilled under any of the following additional assumptions:*

- *V_j are compact for $j = 0, \ldots, i-1$.*
- *V_0 is compact and for $j = 1, \ldots, i-1$, V_j can be written in the separable form $X_j \times U_j$ where X_j is closed and U_j compact.*
- *There is a β such that for $j = 0, \ldots, i-1$, the following hold: $r_j(x_j, u_j) \leq \beta$ for all $(x_j, u_j) \in V_j$ and the sets $\{(x_j, u_j) \mid r_j(x_j, u_j) \geq \alpha, (x_j, u_j) \in V_j\}$ (and if $i = N$ also $r_N(x_N) \leq \beta$ for all $x_N \in V_N$ and the set $\{x_N \mid r_N(x_N) \geq \alpha\}$) are compact for any $\alpha \in R$.*
- *For all feasible strategies and trajectories $\{(x_j, u_j)\}$ leading to x_i the set $\{\{(x_j, u_j)\} \mid \sum_{j=0}^{i-1} r_j(x_j, u_j) \geq \alpha\}$ (and if $i = N$ the set $\{\{(x_j, u_j)\} \mid \sum_{j=0}^{i-1} r_j(x_j, u_j) + r_N(x_N) \geq \alpha\}$) is compact for any $\alpha \in R$.*

and for a truncated OCP starting at $x_i \in RY_i$ the assumption of tameness is fulfilled under any of the following additional assumptions:

- *V_j are compact for $j = i, \ldots, N$.*
- *V_N is compact, V_j can for $j = i, \ldots, N-1$ be written in the separable form $X_j \times U_j$ where X_j is closed and U_j compact.*
- *There is a β such that for $j = i, \ldots, N-1$, the following hold: $r_j(x_j, u_j) \leq \beta$ for all $(x_j, u_j) \in V_j$ and $r_N(x_N) \leq \beta$ for all $x_N \in V_N$, and the sets $\{(x_j, u_j) \mid r_j(x_j, u_j) \geq \alpha, (x_j, u_j) \in V_j\}$ and $\{x_N \mid r_N(x_N) \geq \alpha\}$ are compact for any $\alpha \in R$.*
- *For all feasible strategies and trajectories $\{(x_j, u_j)\}$ leading from x_i the set $\{\{(x_j, u_j)\} \mid \sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N) \geq \alpha\}$ is compact for any $\alpha \in R$.*

Proof. Consider the forwards direction. Taking $\Omega = V_0 \times V_1 \times \ldots \times V_{i-1}$ in the above definition of tameness we see that Ω is compact under the compactness or compactness/closedness assumptions, cf. the proof of Propositions 2.2.1 and 2.2.2. Therefore for any $x_i^\diamond \in Y_i$ and any $\alpha > 0$ the optimal strategy and trajectory leading to $x_i^\diamond$ exists and lies by definition in the compact set Ω, and the optimal criterion value is finite. Therefore the two first conditions are seen to imply tameness.

Consider now the third condition. We see that UB_i is finite due to the assumption involving β. Now we choose $x_i \in Y_i$ and then any $x_i^\diamond \in Y_i$. We see that the strategy and trajectory $\{(x_j^\diamond, u_j^\diamond)\}$ leading to $x_i^\diamond \in Y_i$ satisfies $(x_j^\diamond, u_j^\diamond) \in V_j$ and $\sum_{j=0}^{i-1} r_j(x_j^\diamond, u_j^\diamond) > UB_i(x_i) - \alpha$ for some α. This implies that $r_j(x_j^\diamond, u_j^\diamond) > UB_i(x_i) - \alpha - (i-1)\beta$ because by assumption $r_k(x_k, u_k) \leq \beta$ for all $(x_k, u_k) \in V_k$ for all k. By assumption then this set is compact. As this holds for all j, $0 \leq j \leq i-1$, any optimal strategy and trajectory leading to $x_i^\diamond \in Y_i$ lies in a compact set and the truncated problem is tame.

For the fourth condition we have directly that as by assumption $UB_i(x_i^\diamond) > UB_i(x_i) - \alpha$ for some α the optimal strategy and trajectory leading to $x_i^\diamond$ must lie in $\{(x_j^\diamond, u_j^\diamond) \mid \sum r_j(x_j^\diamond, u_j^\diamond) > UB_i(x_i) - \alpha\}$ which by assumption is a compact set.

Similar argumentation holds for the truncated problem starting at stage i. □

Example 2.5.1 *Consider the QL problem (2.21), (2.24) with fixed initial point $x_0 = \underline{x}_0$. First assume that $R_i^{xx} = 0$, $i = 1, \ldots, N$, $R_i^{xu} = 0$, $i = 0, \ldots, N-1$ and $R_i^{uu} < 0$ for $i = 0, \ldots, N-1$; R_0^{xx} and R_0^{xu} are not needed when the initial point is fixed. For this problem V_i are not compact. Consider the forwards direction. The first two conditions of Proposition 2.5.1 are not fulfilled. The*

third condition is not fulfilled because r_i are linear with respect to x_i. The last condition is fulfilled. Therefore any partial problem ending at a $x_i \in Y_i$ is tame for any i.

Consider the backwards direction. Again the first three conditions are not fulfilled. The last condition is fulfilled, but only for $i = 0$. Therefore the whole problem is tame, but any partial problem starting at x_i for $i \geq 1$ need not be so.

Now in addition assume that the local constraints (2.22) - (2.23) are given as $x_0 = \underline{x}_0$ and for $i = 0, \ldots, N-1$ $\underline{u}_i \leq u_i \leq \overline{u}_i$, where $\underline{u}_i$ and $\overline{u}_i$ are finite and $\underline{u}_i \leq \overline{u}_i$. In the forwards direction the second and the last conditions are fulfilled but the two others are not. In the backwards direction the only assumption fulfilled is the last one for $i = 0$. If a constraint $\underline{x}_N \leq x_N \leq \overline{x}_N$ is added where $\underline{x}_N$ and $\overline{x}_N$ are finite and $\underline{x}_N \leq \overline{x}_N$ then the second and the last conditions are fulfilled in the backwards direction, while the two others are not.

Finally assume again that the problem is unconstrained and that now $\begin{pmatrix} R_i^{xx} & R_i^{xu} \\ R_i^{xu\prime} & R_i^{uu} \end{pmatrix} < 0$ *for $i = 0, \ldots, N-1$ and $R_N^{xx} < 0$. For both the forwards and the backwards direction the last two conditions are fulfilled but the first two are not. This is also true if the condition on the matrix* $\begin{pmatrix} R_0^{xx} & R_0^{xu} \\ R_0^{xu\prime} & R_0^{uu} \end{pmatrix}$ *is weakened to $R_0^{uu} < 0$ provided x_0 is constrained to a compact set (e.g., if the initial point constraint $x_0 = \underline{x}_0$ is added). It is also true if the condition on the matrix R_N^{xx} is omitted provided x_N is constrained to a compact set.* □

The assumptions of LMR regularity and tameness are essential to the analysis of mathematical programming problems. Tameness can, except for pathological examples, be assumed to hold for most well formulated problems with practical origin, cf. the relatively weak assumptions of Proposition 2.5.1.

Therefore the essential assumption is LMR regularity. It appears that in many optimal control problems truncated problems are *not* LMR regular. The following examples illustrate this.

Example 2.5.2 *For the problem (2.29) - (2.31) let $z \in R$, $v \in R$, $r(z,v) = v$, $h(z,v) = v + z + 3$, $g(z,v) = v - 1$. We find the Lagrangian $L^\rho = \rho v - \mu(v + z + 3) - \lambda(v - 1)$. The condition $0 \in \partial_v L$ means in this case (all functions continuously differentiable) that $0 = \nabla_v L = \rho - \mu - \lambda$.*

Assume $z = 2$. Then $v^* = -5$. For $\rho = 0$ the only solution to $0 = \nabla_v L$ is $(\mu, \lambda) = (0,0)$. Therefore the assumption of LMR regularity of the problem is fulfilled.

Now let $z = -4$. Then $v^* = 1$. With $\rho = 0$ the solution set to $0 = \nabla_v L^\rho$ is $\{(\lambda, \mu) \mid \lambda \geq 0 \wedge 0 = \mu + \lambda\}$. Since this solution set contains elements other than $(0,0)$ the problem is not LMR regular at $z = -4$. It is easily verified that $ub(z) = -z - 3$, and that ub is defined for $z \geq -4$. We see that ub is in fact linear wherever it is defined. Thus, the difficulty arises at the boundary of the set on which ub is defined. *Cf. Figure 2.3.* □

Example 2.5.3 *The above Example 2.5.2 may be modified to illustrate the first and last stages of an OCP. Let $n = 1$, $m = 1$, $N = 3$, $r_0(x_0, u_0) = u_0$, $r_1(x_1, u_1) = u_1/2$, $r_2(x_2, u_2) = u_2$, $r_3(x_3) = 0$, $f_i(x_i, u_i) = x_i + u_i + 3$, $g_i(x_i, u_i) = u_i - 1$, $i = 0, 1, 2$, $x_0 = \underline{x}_0 = -4$, $x_3 = \underline{x}_3 = 7$.*

We find $Y_0 = \{-4\}$, $Y_1 = \{x_1 \mid x_1 \leq 0\}$, $Y_2 = \{x_2 \mid x_2 \leq 4\}$, $Y_3 = \{x_3 \mid x_3 \leq 8\}$, $RY_3 = \{7\}$, $RY_2 = \{x_2 \mid x_2 \geq 3\}$, $RY_1 = \{x_1 \mid x_1 \geq -1\}$ and $RY_0 = \{-4\}$. Cf. Figure 2.4.

We find the unique optimal solution $u_0^* = 1$, $u_1^* = 0$, $u_2^* = 1$, $x_1^* = 0$ and $x_2^* = 3$.

Thus x_1^* is at the boundary of Y_1 and x_2^* is at the boundary of RY_2.

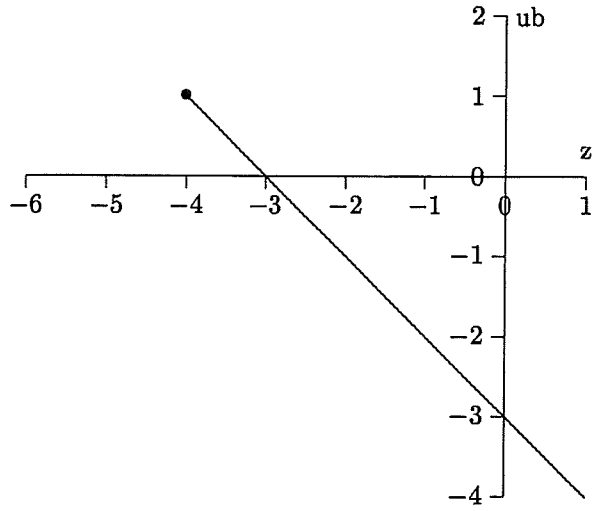


Figure 2.3: The upper boundary, cf. Example 2.5.2

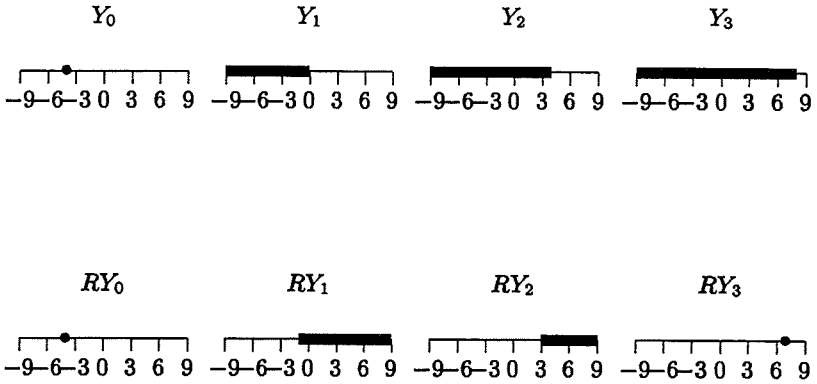


Figure 2.4: Y_i and RY_i, cf. Example 2.5.3

The Lagrangian of the truncated problem ending at x_1^* is

$$\rho u_0 - \lambda_0(u_0 - 1) - \mu_0(\underline{x}_0 + u_0 + 3)$$

$0 \in \partial_u L^\rho$ *means in this case* $\nabla_u L^\rho = 0$, *i.e.* $\rho - \lambda_0 - \mu_0 = 0$. *With* $\rho = 0$ *we see that there are solutions to this for* (λ_0, μ_0), *e.g.* $(1, -1)$. *Since this solution also satisfies the complementary slackness condition the problem is not LMR regular.*

The Lagrangian to the truncated problem ending at x_2^* is, considering (u_0, u_1, x_1) as variables,

$$\rho(u_0 + u_1/2) - \lambda_0(u_0 - 1) - \mu_0(x_0 + u_0 + 3 - x_1) - \lambda_1(u_1 - 1) - \mu_1(x_1 + u_1 + 3)$$

The condition $0 \in \partial L^\rho$ *means in this case* $0 = \nabla L^\rho$ *which can be written*

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} \rho - \lambda_0 - \mu_0 \\ \rho/2 - \lambda_1 - \mu_1 \\ \mu_0 - \mu_1 \end{pmatrix}$$

We have $g_1(x_1^*, u_1^*) = 0 - 1 < 0$ *and therefore* $\lambda_1^* = 0$. *We then see that with* $\rho^* = 0$ *the only solution is* $(\lambda_0^*, \lambda_1^*, \mu_0^*, \mu_1^*) = (0, 0, 0, 0)$, *and this truncated problem is therefore LMR regular.*

It may be verified that the OCP is LMR regular and also the truncated problem starting at x_1^*, but the truncated problem starting at x_2^* is not. So as in the above example 2.5.2 we see that if x_i^* is at the boundary of RY_i the problem is not LMR regular. As seen from Propositions 2.6.1, 2.6.2 and 2.6.3 below this is not incidental: LMR regularity implies interiority. □

Constraint Qualification and KKT Conditions for Differentiable Functions

Now we assume that all functions r_i, f_i, g_i and h_i are continuously differentiable.

We define the Lagrangian L^ρ as in (2.32):

$$L^\rho(z, v, \rho, \lambda, \mu) = \rho r(z, v) - \lambda g(z, v) - \mu h(z, v) \tag{2.33}$$

Similarly to the above the following *Lagrange Multiplier Rule* holds:

> If v^* solves (2.29) - (2.31) then there exist $(\rho^*, \lambda^*, \mu^*)$, such that
> $0 = \nabla_v L^\rho(z, v^*, \rho^*, \lambda^*, \mu^*)$, $\rho^* \geq 0$, $\lambda^* \geq 0$, $\lambda^* g(z, v^*) = 0$,
> $(\rho^*, \lambda^*, \rho^*) \neq (0, 0, 0)$

This corresponds to the *Fritz-John's conditions*. If $\rho^* \neq 0$ we can take $\rho^* = 1$, and the LMR corresponds to the *Karush-Kuhn-Tucker conditions* formulated in the usual Lagrangian L:

$$\nabla_v L = \nabla_v(r(z, v^*) - \lambda^* g(z, v^*) - \mu^* h(z, v^*)) = 0 \tag{2.34}$$

$$\lambda^* g(z, v^*) = 0, \lambda^* \geq 0 \tag{2.35}$$

As before we shall say that the problem is LMR regular if with $\rho^* = 0$ the only solution for (λ^*, μ^*) in the Lagrange Multiplier Rule is $(0, 0)$.

Similar to the Lipschitz case, LMR regularity at z implies that z is in the interior of the set for which there are feasible solutions, Gauvin and Debeau (1982).

The KKT conditions hold at an optimal v^* if a constraint qualification holds. We shall here discuss two constraint qualifications, the linear independence (LI) and the Mangasarian-Fromowitz

(MF) constraint qualification (CQ). They are defined in relation to the problem (2.29) - (2.31). We shall develop them in stagewise versions suitable for application in connection with maximum principles and dynamic programming. This implies that at stage i we consider only u_i as an optimization variable while x_i is considered as a parameter.

For the problem (2.29) - (2.31) these two CQs are as follows:

LI-CQ

The gradients $\nabla_v h$, $\nabla_v g_j$, $j \in \{k \mid g_k(z, v^*) = 0\}$ are linearly independent.

MF-CQ

- The gradients $\nabla_v h$ are linearly independent
- There is a δv such that
  - $\nabla_v h \delta v = 0$
  - $\nabla_v g_j \delta v < 0$, $j \in \{k \mid g_k(z, v^*) = 0\}$.

For the OCP the corresponding stagewise CQs will be formulated as follows:

The Stagewise LI-CQ

The gradients $\nabla_u h_i(x_i, u_i^*)$, $\nabla_u g_i^j(x_i, u_i^*)$, $j \in \{j \mid g_i^j(x_i, u_i^*) = 0\}$ are linearly independent for $i = 0, \ldots, N-1$

The Stagewise MF-CQ

- The gradients $\nabla_u h_i(x_i, u_i^*)$ are linearly independent for $i = 0, \ldots, N-1$
- There is a $\delta u_i \in R^m$ such that for $i = 0, \ldots, N-1$
  - $\nabla_u h_i(x_i, u_i^*)\delta u_i = 0$,
  - $\nabla_u g_i(x_i, u_i^*)\delta u_i < 0$, $j \in \{k \mid g_i^k(x_i, u_i^*) = 0\}$.

It is seen that if the conditions are fulfilled at all stages individually then they are also fulfilled simultaneously for all stages $i = 0, \ldots, N-1$.

The following examples show that the CQ may be fulfilled for the OCP without being fulfilled for the truncated problems. This may be contributed to the fact that the stagewise CQ involve only a subset of the variables of the whole problem (viz., the control variables).

Example 2.5.4 *Consider the following problem with* $n = 1$, $N = 3$, $u_0 \in R$, $u_1 \in R$ *and* $u_2 \in R^2$*:*

$$\max[-\tfrac{1}{2}(u_0)^2 - \tfrac{1}{2}(u_1)^2 - (u_2^1)^2 - 8u_2^2]$$

$$x_{i+1} = f_i(x_i, u_i) = x_i + u_i^1$$

$$g_2^1(x_2, u_2) = 2x_2 + u_2^1 - 9 \leq 0$$

$$g_2^2(x_2, u_2) = -u_2^2 \leq 0$$

$$h_2(x_2, u_2) = x_2 + u_2^1 + u_2^2 - 5 = 0$$

$$x_0 = 0$$

$$x_3 \in R$$

One may verify that the unique solution is $u_0^* = 2$, $u_1^* = 2$, $u_2^* = (1, 0)'$, $x_1^* = 2$, $x_2^* = 4$ *and* $x_3^* = 3$.

We rewrite the dynamical equation as $f_i(x_i,u_i)-x_{i+1}=0$ *and consider* x_0 *a fixed value. Then the problem is of the form (2.29) - (2.31) with variables* $(u_0,x_1,u_1,x_2,u_2^1,u_2^2,x_3)$. *We then find that the gradient of the active constraints at the optimum (all inequality constraints are active) can be indicated as the matrix*

	u_0	x_1	u_1	x_2	u_2^1	u_2^2	x_3
∇f_0	1	−1	0	0	0	0	0
∇f_1	0	1	1	−1	0	0	0
∇f_2	0	0	0	1	1	0	−1
∇g_2^1	0	0	0	2	1	0	0
∇g_2^2	0	0	0	0	0	−1	0
∇h_2	0	0	0	1	1	1	0

We see that the rows of the matrix are linearly independent. We also see that the three rows corresponding to g_2 *and* h_2 *for the two columns corresponding to* u_2, *i.e. the rows of the matrix*

$$\begin{pmatrix} 1 & 0 \\ 0 & -1 \\ 1 & 1 \end{pmatrix}$$

are not linearly independent. Thus, the LI-CQ holds for the OCP but not stagewise. □

Example 2.5.5 *We continue the above Example 2.5.4. We may verify that with* $\delta u_0=0$, $\delta u_1=0$, $\delta u_2=(-1,1)'$, $\delta x_1=0$, $\delta x_2=0$ *and* $\delta x_3=-1$ *we have all linearized equality constraints fulfilled and all linearized inequality constraints holding as strict inequalities. Thus the MF-CQ holds for the OCP. The* $(\delta x,\delta u)$ *chosen also illustrates that the MF-CQ holds stagewise.* □

Example 2.5.6 *We continue the above Example 2.5.4 and introduce an additional constraint* $g_2^3(x_2,u_2)=-u_2^1+1\leq 0$. *One may verify that now the stagewise MF-CQ does not hold at stage 2. The MF-CQ is seen to hold for the whole problem with* $\delta u_0=0$, $\delta u_1=-2$, $\delta u_2=(1,1)'$, $\delta x_1=0$, $\delta x_2=-2$ *and* $\delta x_3=-1$. □

The OCP with free end point has traditionally been the main problem analyzed, cf. Section 1.3. If originally the OCP contains end constraints then these may be eliminated by transformation to stage $N-1$ using the dynamic equation $x_N=f_{N-1}(x_{N-1},u_{N-1})$ (this is not a panacea, though, since this introduces state dependence in the local constraints). In relation to constraint qualifications this problem may have attractive properties, as seen in the following two propositions 2.5.2 and 2.5.3. Observe the one-way implication in the following results. Also in other respects these assumptions are convenient, see e.g. Propositions 2.7.3 and 3.4.3.

Proposition 2.5.2 *Consider an OCP with free end point (i.e. no constraints* g_N *or* h_N*). Assume that* f_i, g_i *and* h_i *are continuously differentiable. If LI-CQ holds stagewise then LI-CQ holds for the OCP in the form (2.29) - (2.31) and for any truncated problem.*

Proof. The rows of the gradient of the reformulated dynamical equation $f_i(x_i,u_i)-x_{i+1}$ are linearly independent due to the term x_{i+1}. Moreover, these are linearly independent with the similar rows from the reformulated dynamical equation $f_j(x_j,u_j)-x_{j+1}$ since variables x_{i+1} are different from variables x_{j+1} for $i\neq j$. Since the reformulated dynamic equation $f_i(x_i,u_i)-x_{i+1}=0$ contains the variable x_{i+1} and h_i and g_i do not and since variables u_i are different from variables u_j, $i\neq j$, the linear independence of the rows of the equality constraint gradient matrix for the OCP is assured from the first part of the stagewize LI-CQ. □

Proposition 2.5.3 *Consider an OCP with free end point (i.e. no constraints g_N or h_N). Assume that f_i, g_i and h_i are continuously differentiable. If MF-CQ holds stagewise then MF-CQ holds for the OCP in the form (2.29) - (2.31) and for any truncated problem.*

Proof. The rows of the gradient of the reformulated dynamical equation $f_i(x_i, u_i) - x_{i+1}$ are linearly independent due to the term x_{i+1}. Moreover, these are linearly independent with the similar rows from the reformulated dynamical equation $f_j(x_j, u_j) - x_{j+1}$ since variables x_{i+1} are different from variables x_{j+1} for $i \neq j$. Since the reformulated dynamic equation $f_i(x_i, u_i) - x_{i+1} = 0$ contains the variable x_{i+1} and h_i does not and since variables u_i are different from variables u_j, $i \neq j$, the linear independence of the rows of the equality constraint gradient matrix for the OCP is assured from the first part of the stagewize MF-CQ.

Now let $\delta x_0 = 0$. Choose δu_0 such that the stagewise MF-CQ hold at stage 0. Calculate δx_1 according to the linearized dynamics, i.e. $\delta x_1 = \nabla_u f_0(x_0^*, u_0^*)\delta u_0$. Choose δu_1 such that $\nabla_x h_1(x_1^*, u_1^*)\delta x_1 + \nabla_u h_1(x_1^*, u_1^*)\delta u_1 = 0$ and $\nabla_x g_1^j(x_1^*, u_1^*)\delta x_1 + \nabla_u g_1^j(x_1^*, u_1^*)\delta u_1 < 0$ for active inequality constraints. This is possible, because when the stagewise MF-CQ holds at x_1^* then also it holds for any other $x_1 = x_1^* + \delta x_1$ since the system is linear. As seen, the MF-CQ holds at stage 1, with the given δx_1. Continue to calculate δx_i according to the linearized dynamical equation and δu_i fulfilling the stagewise MF-CQ up to stage $N-1$. With this $(\delta x_0, \delta u_0, \delta x_1, ..., \delta u_{N-1})$ also the second part of the MF-CQ are seen to hold for the OCP. Similar argumentation can be used for the truncated problem. □

Stagewise KKT Conditions

We finally define the *stagewise KKT conditions* as follows: At an optimal solution (x^*, u^*) to the OCP there exists at each stage a $(p_{i+1}^*, \lambda_i^*, \mu_i^*)$ such that for $i = 0, \ldots, N-1$ there holds:

$$\nabla_u(H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)) = 0 \tag{2.36}$$

$$\lambda_i^* \geq 0, \lambda_i^* g_i(x_i^* u_i^*) = 0 \tag{2.37}$$

and at stage N there holds

$$\nabla(r_N(x_N^*)\lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*)) = 0 \tag{2.38}$$

$$\lambda_N^* \geq 0, \lambda_N^* g_N(x_N^*) = 0 \tag{2.39}$$

Here, H_i is the Hamiltonian

$$H_i(x_i, u_i, p_{i+1}) = r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) \tag{2.40}$$

Define the adjoint equations:

$$p_i^* = \nabla_x(H_i(x_i^*, u_i^*, p_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)) \tag{2.41}$$

$$p_N^* = \nabla(r_N(x_N^*) - \lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*)) \tag{2.42}$$

We then have

Proposition 2.5.4 *Assume that r_i, f_i, g_i and h_i are continuously differentiable. Let (x^*, u^*) solve the OCP. Let a corresponding (p^*, λ^*, μ^*) be given. Then the KKT conditions hold for the OCP in the form (2.29) - (2.31) with (p^*, λ^*, μ^*) if and only if the stagewise KKT conditions and the adjoint equation hold with (p^*, λ^*, μ^*) for $i = 0, \ldots, N$.*

Proof. The KKT conditions for the OCP are exactly the same as the stagewise KKT conditions and the adjoint equations *taken together*, for $i = 0, \ldots, N$. □

In Section 1.4 in connection with Proposition 1.4.6 we called the KKT conditions the weak maximum principle. We therefore see that *the weak maximum principle holds if and only if the stagewise KKT conditions and the adjoint equations hold for some* (p^*, λ^*, μ^*).

An important observation made above is that *the requirement of constraint qualification for truncated problems seems not to be fulfilled in many OCP.* But stagewise decomposition as this is done in the maximum principle and dynamic programming work with truncated problems.

We now discuss two ways to get around the difficulties when the constraint qualifications do not hold. One is to *aggregate stages.* The other is to reformulate the problem by introducing *additional variables.*

Stage Aggregation

Suppose that in the definition of ub_i^{i+1} the problem does not fulfill a CQ. We may then attempt to get a CQ fulfilled by redefinition of the OCP by aggregation of stages i and $(i+1)$. This may be done as follows. Let the original control variable be called u_j as usual. Then define new control variables z_j: for $0 \le j \le i-1$ let $z_j = u_j$; let $z_i = (u_i', u_{i+1}')'$; for $i+2 \le j \le N-1$ let $z_{j-1} = u_j$. The state variables, dynamic equations, local constraints, number of stages and stage indexes are transformed accordingly. Then maybe a CQ are fulfilled at stage i in the new formulation.

A more direct way to attain essentially the same is to maintain the original definition of the problem and variables, but skip the definitions of ub_i^{i+1} and ub_{i+1}^{i+2} and then define ub_i^{i+2} as

$$ub_i^{i+2}(x_i, x_{i+2}) = \max_{u_i, x_{i+1}, u_{i+1}} [r_i(x_i, u_i) + r_{i+1}(x_{i+1}, u_{i+1})] \tag{2.43}$$

$$f_i(x_i, u_i) = x_{i+1} \tag{2.44}$$

$$f_{i+1}(x_{i+1}, u_{i+1}) = x_{i+2} \tag{2.45}$$

$$u_i \in U_i(x_i) \tag{2.46}$$

$$(x_{i+1}, u_{i+1}) \in V_{i+1} \tag{2.47}$$

or by eliminating x_{i+1} as

$$ub_i^{i+2}(x_i, x_{i+2}) = \max_{u_i, u_{i+1}} [r_i(x_i, u_i) + r_{i+1}(f_i(x_i, u_i), u_{i+1})] \tag{2.48}$$

$$f_{i+1}(f_i(x_i, u_i), u_{i+1}) = x_{i+2} \tag{2.49}$$

$$(x_i, u_i) \in V_i \tag{2.50}$$

$$(f_i(x_i, u_i), u_{i+1}) \in V_{i+1} \tag{2.51}$$

Similarly, if the problem defining $RUB_j(x_j)$ does not fulfill the CQ then this stage may be skipped, and RUB_{j-1} is considered; and if the problem defining $UB_k(x_k)$ does not fulfill the CQ then this stage may be skipped and UB_{k+1} is considered.

We shall call the application of such ideas *stage aggregation.*

Proposition 2.5.5 *Suppose a CQ holds for the OCP. Then there is a stage aggregation such that the CQ holds stagewise.*

Proof. Trivially, we can aggregate all stages such that a one-stage system attains. As this is the OCP the CQ holds by assumption. □

Additional Variables

Another way to treat the difficulty when a constraint qualification does not hold is to introduce artificial control variables $\phi_i^+ \in R^n$, $\phi_i^- \in R^n$, $\gamma_i \in R^k$, $\eta_i^+ \in R^l$, and $\eta_i^- \in R^l$, and then redefine the dynamic equation and the local restrictions as

$$x_{i+1} = f_i(x_i, u_i) + \phi_u^+ - \phi_u^- \tag{2.52}$$
$$g_i(x_i, u_i) - \gamma_i \leq 0 \tag{2.53}$$
$$h_i(x_i, u_i) + \eta_i^+ - \eta_i^- = 0 \tag{2.54}$$
$$g_N(x_N) - \gamma_N \leq 0 \tag{2.55}$$
$$h_N(x_N) + \eta_N^+ - \eta_N^- = 0 \tag{2.56}$$
$$\phi_i^+ \geq 0, \phi_i^- \geq 0, \gamma_i \geq 0, \eta_i^+ \geq 0, \eta_i^- \geq 0 \tag{2.57}$$

In order to have the new variables take the optimal values zero, if possible, they are penalized by a large positive constant c in the criterion. Thus the local criterion at stage i is

$$r_i(x_i, u_i) - c(\sum_{j=1}^{n}((\phi_i^+)^j + (\phi_i^-)^j) + \sum_{j=1}^{k}(\gamma_i)^j + \sum_{j=1}^{l}((\eta_i^+)^j + (\eta_i^-)^j)) \tag{2.58}$$

and at stage N:

$$r_N(x_N) - c(\sum_{j=1}^{k}(\gamma_N)^j + \sum_{j=1}^{l}((\eta_N^+)^j + (\eta_N^-)^j)) \tag{2.59}$$

We say that the original problem and the new problem have the same optimal solution if they have the same optimal solution with respect to (x, u) and all additional variables are zero.

The following result is formulated for the OCP but obviously also applies to any truncated problem.

Proposition 2.5.6 *In the new formulation (2.52) - (2.59) we have $X_i = R^n$, $Y_i = R^n$, $RY_i = R^n$ and $W_i^{i+1} = R^{2n}$ for all i.*

Now assume that r_i, f_i, g_i and h_i are Lipschitz continuous. Then the problem in the new formulation is LMR regular.

Finally assume in addition that the problem in the original formulation is tame and that the original and the new problems have the same optimal solution sets. Then the problem in the new formulation is tame.

Proof. The first part is obvious. To prove the second part we first let $\rho = 0$, and must show that all Lagrange multipliers (LM) are zero. Consider the first restriction in (2.53): $g_i^1(x_i, u_i) - \gamma_i^1 \leq 0$. The LM to this restriction is non-negative. The variable γ_i^1 enters only in one other restriction, viz., $-\gamma_i^1 \leq 0$ (which is a reformulation of (2.57): $\gamma_i^1 \geq 0$). The LM to this is also non-negative. In both restrictions γ_i^1 enters with a minus sign. Since we had $\rho = 0$, this forces us to take these two LM to zero, since otherwise we could not have $0 \in \partial_u L^\rho$ (in case of Lipschitz functions) or $0 = \nabla_u L^\rho$

(in case of continuously differentiable functions) with respect to the component corresponding to γ_i^1. Similar argumentation applies to all inequality constraints.

Now consider the first equality constraint in (2.52) corresponding to $(\phi_i^+)^1$ and $(\phi_i^-)^1$. Assume $(\phi_i^+)^1 > 0$. The LM corresponding to $-(\phi_i^+)^1 \leq 0$ (a reformulation of (2.57): $(\phi_i^+)^1 \geq 0$) must be zero, by the complementary slackness condition. Therefore also the LM to the equality constraint (2.52) in which $(\phi_i^+)^1$ is involved must be zero, since the two places are the only ones, where the variable $(\phi_i^+)^1$ enters, and since we had $\rho = 0$ and require $0 \in \partial_u L^\rho$ (or $0 = \nabla_u L^\rho$). Similarly, if we assume $(\phi_i^-)^1 > 0$ the LM to the equality constraint (2.52) in which $(\phi_i^-)^1$ is involved must be zero.

If $(\phi_i^+)^1 = (\phi_i^-)^1 = 0$ we argue as follows. Assume that the LM to the equality restriction (2.52) in which these two variables are involved is positive. Then it will not be possible to have $0 \in \partial_u L^\rho$ with respect to the component corresponding to $(\phi_i^-)^1$, since this variable only enters one other place (viz., in $-(\phi_i^-)^1 \leq 0$, a reformulation of (2.57)) and the LM corresponding to this is restricted to be non-negative. Similarly it is seen that the LM on the equality constraint cannot be negative. If it is zero, then again we see that the LM on $-(\phi_i^+)^1 \leq 0$ and $-(\phi_i^-)^1 \leq 0$ must be zero. Similar argumentation holds for all other equality constraints. The conclusion is that if $\rho = 0$ then all LM must be zero. Therefore the problem is LMR regular.

Finally we show that the new problem is tame. By assumption the original problem is tame. This implies that the solution set lies in a compact set. As it is assumed that the original problem and the reformulated problem have the same optimal solution sets also the reformulated problem has the optimal solutions in a compact set. Moreover, since the optimal solutions are the same, so are the optimal criterion function values. Therefore the new formulation yields a tame problem. □

It is a consequence of the constraint qualification that *the point in question is interior to the set on which solutions exits*, cf. Propositions 2.6.1, 2.6.2 and 2.6.3 below. Thus for instance, if we are considering RUB_i, the assumption of constraint qualifications being fulfilled for the truncated problem starting from at x_i implies that x_i is interior to RY_i. As Example 2.5.3 above shows it can not always be expected that x_i^* is in the interior of Y_i or RY_i. As seen, the above Proposition 2.5.6 confirms that if the problem is reformulated then any x_i is in the interior of these sets.

Therefore one of the attractive properties of such reformulation is that $U_i(x_i) \neq \emptyset$ for all $x_i \in R^n$. This means that for any trajectory up to stage i an admissible strategy can be found. In particular, an initial admissible solution can easily be found in connection with algorithms.

When artificial variables are introduced in connection with algorithms it is usually assumed that a sufficiently large value of c will secure that the solution to the reformulated problem is also the solution to the original problem.

The reformulation indicates a complication, though. At the boundary of the original sets Y_i, the new UB_i will normally be *non-smooth*, and similarly the new RUB_i will normally be *non-smooth* at the boundary of the original RY_i. This is because typically (in the smooth case) the MF-CQ will hold but not the LI-CQ.

However, the non-smoothness introduced will be irrelevant since they are located precisely at the boundaries of Y_i, RY_i and W_i^{i+1}, and therefore effectively eliminated by the constraints.

Example 2.5.7 *Consider the problem formulated at the beginning of Example 2.5.2, with $x = -4$. This problem was shown not to be LMR regular. We introduce the artificial variables ϕ^+, ϕ^- and γ and penalize by $c = 3$. We then get the problem*

$$\max_{u,\phi^+,\phi^-,\gamma} [r(x,u) - 3(\phi^+ + \phi^- + \gamma)]$$

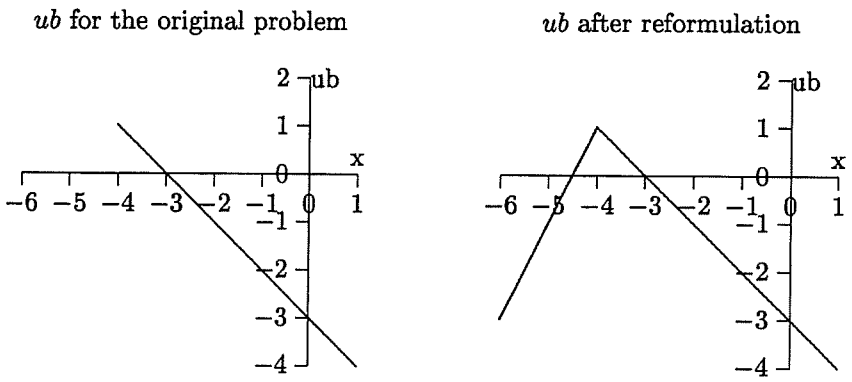


Figure 2.5: ub before and after reformulation with artificial variables

$$u + x + 3 + \phi^+ - \phi^- = 0$$
$$u - 1 - \gamma \leq 0$$
$$\phi^+ \geq 0, \phi^- \geq 0, \gamma \geq 0$$

Now it may be verified that the problem is LMR regular for any $x \in R$. We find ub for the problem as $ub(x) = -x-3$ for $x \geq -4$ (i.e., the same as in the original formulation) and $ub(x) = 2x+9$ for $x \leq -4$, cf. Figure 2.5. We see that ub is Lipschitz continuous, and differentiable everywhere except at $x = -4$. This point was at the boundary of the set on which ub was defined before the introduction of the additional variables. □

Finally observe that *the above reformulation preserves concavity, convexity and continuity properties* of the functions involved in the definition of the OCP. Thus, if the assumptions of Propositions 2.4.1, 2.4.2, 2.4.3 and 2.4.4 are fulfilled for the original problem they are also fulfilled for the reformulated problem. Also *linearity and affinity are preserved.* Finally *state sensitivity (p. 71) is preserved.*

The criterion in the new formulation is not strictly concave as all terms involving the new variables are linear. It is easy to see that we may *preserve strict concavity of the criterion by adding quadratic, strictly concave terms* to the expressions in (2.58) - (2.59) such that the criterion at stage i in the new formulation is:

$$\begin{aligned} & r_i(x_i, u_i) \\ & -c_1(\sum_{j=1}^{n}((\phi_i^+)^j + (\phi_i^-)^j) + \sum_{j=1}^{k}(\gamma_i)^j + \sum_{j=1}^{l}((\eta_i^+)^j + (\eta_i^-)^j)) \\ & -c_2(\sum_{j=1}^{n}(((\phi_i^+)^j)^2 + ((\phi_i^-)^j))^2 + \sum_{j=1}^{k}((\gamma_i)^j)^2 + \sum_{j=1}^{l}(((\eta_i^+)^j)^2 + ((\eta_i^-)^j))^2) \end{aligned} \tag{2.60}$$

and at stage N:

$$r_N(x_N) - c_1(\sum_{j=1}^{k}(\gamma_N)^j + \sum_{j=1}^{l}((\eta_N^+)^j + (\eta_N^-)^j)) \tag{2.61}$$

$$-c_2(\sum_{j=1}^{k}((\gamma_N)^j)^2 + \sum_{j=1}^{l}(((\eta_N^+)^j)^2 + ((\eta_N^-)^j))^2)$$

Here $c_1 > 0$ and $c_2 > 0$. With this formulation we see that if the original criterion is strictly concave and/or quadratic with respect to the control and state variables or with respect to the control variables alone the new criterion (2.60) - (2.61) has the same properties.

We now return to the discussion of the upper boundaries.

2.6 Lipschitz Continuity

We next consider conditions leading to Lipschitz continuity of the upper boundary functions.

Proposition 2.6.1 *Assume that r_j, f_j, g_j and h_j are locally Lipschitz for $0 \leq j < i$. Assume that the truncated problem ending with $x_i \in Y_i$ is tame and LMR regular along every optimal strategy and trajectory $\{(x_j^*, u_j^*)\}$ leading to x_i. Then UB_i is Lipschitz near x_i and $\partial UB_i \subset co\{p_i\}$, where $co\{p_i\}$, denotes the convex hull of multipliers p_i satisfying the LMR rule (see after (2.32)) for an optimal solution to the truncated problem ending at x_i.*

Proof. We can formulate the truncated problem as a problem of the form (2.29) - (2.31): here the parameter z in (2.29) - (2.31) corresponds to x_i and the variable v in (2.29) - (2.31) corresponds to $\{(x_j, u_j)\}$, $j = 0, \ldots, i-1$. Then the result follows from Clarke (1983) p. 242. - For problems with continuously differentiable r_j, f_j, g_j and h_j the results also follow from Gauvin and Dubeau (1982) pp. 105 and 116-117. □

Proposition 2.6.2 *Assume that r_j, f_j, g_j and h_j are locally Lipschitz for $i \leq j \leq N-1$, and that r_N, g_N, h_N are locally Lipschitz. Assume that the truncated problem starting from $x_i \in RY_i$ is tame and LMR regular along every optimal terminating strategy and trajectory $\{(x_j^*, u_j^*)\}$ from x_i. Then RUB_i is Lipschitz near x_i and $\partial RUB_i \subset co\{-p_i\}$, where $co\{p_i\}$, denotes the convex hull of multipliers p_i satisfying the LMR rule (see after (2.32)) for an optimal solution to the truncated problem starting at x_i.*

Proof. We can formulate the truncated problem as a problem of the form (2.29) - (2.31); here z in (2.29) - (2.31) corresponds to x_i and v in (2.29) - (2.31) corresponds to $\{(x_j, u_j)\}$, $j = i+1, \ldots, N-1$, and x_N. Then the result follows as in the proof of Proposition 2.6.1. □

In relation to the two previous Propositions we observe the following. A truncated problem (e.g. from stage 0 to x_k or from x_k to stage N) need not satisfy the constraint qualification even if the whole OCP does. Therefore if UB_i is Lipschitz continuous near x_i this does not imply that any UB_j, $j < i$, is locally Lipschitz near the x_j which is in an optimal trajectory leading to x_i. Similarly if RUB_i is Lipschitz continuous near x_i this does not imply that any RUB_j, $i < j$, is locally Lipschitz near the x_j which is in an optimal terminating trajectory starting from x_i. This is a consequence of the earlier observation that stagewise decomposition needs stronger assumptions.

For the smaller upper boundaries we define the Lagrangian

$$L_i = r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) - \lambda_i g_i(x_i, u_i) - \mu_i h_i(x_i, u_i) \tag{2.62}$$

and the LMR rule is

$$0 \in \partial_{x_i, x_{i+1}} L_i \tag{2.63}$$

Proposition 2.6.3 *Assume that r_i, f_i, g_i and h_i are locally Lipschitz. Assume that the problem defining $ub_i^{i+1}(x_i, x_{i+1})$ is tame, and that it is LMR regular for every u_i^* defining $ub_i^{i+1}(x_i, x_{i+1})$. Then $ub_i^{i+1}(.,.)$ is Lipschitz near (x_i, x_{i+1}) and $\partial UB_i^{i+1} \subset co\{-p_i, p_{i+1}\}$.*

Proof. We can formulate this problem as a problem of the form (2.29) - (2.31): here z in (2.29) - (2.31) corresponds to (x_i, x_{i+1}) and v in (2.29) - (2.31) corresponds to u_i. Then the result follows as in the proof of Proposition 2.6.1. □

The results on Lipschitz continuity may be extended to parameters other than the states, cf. the remark at the end of the next section.

2.7 Continuous Differentiability of Upper Boundaries

Now turn to continuous differentiability of the upper boundaries. The multipliers referred to are those defined in relation to the stagewise KKT conditions, cf. Proposition 2.5.4.

Proposition 2.7.1 *Assume that r_j, f_j, g_j and h_j are locally Lipschitz for $0 \leq j < i$. Consider the truncated problem ending at $x_i \in Y_i$ and assume that it is tame. If the optimal $\{(x_j^*, u_j^*)\}$ leading to x_i is unique and if this problem is LMR regular and has unique multipliers (p^*, λ^*, μ^*) then UB_i is continuously differentiable at x_i and $\nabla UB_i(x_i) = -p_i^*$*

Proof. We consider the truncated problem ending at x_i given in the formulation (2.29) - (2.31). It follows from Clarke (1983) p. 242 that under the assumptions made UB_i is strictly differentiable at x_i and that the derivative is as stated. Due to the Lipschitz continuity of all functions the assumption of uniqueness of the solution and the multipliers will also hold true for any x_i in a small neighborhood of x_i. From Clarke (1983) pp. 33 - 34 it follows that UB_i is continuously differentiable. - For continuously differentiable functions r_j, f_j, g_j and h_j the result also follows from Gauvin (1980). □

Proposition 2.7.2 *Assume that r_j, f_j, g_j and h_j are locally Lipschitz for $i \leq j \leq N-1$, and that r_N, g_N, h_N are locally Lipschitz. Consider the truncated problem starting from $x_i \in RY_i$ and assume that it is tame. If the optimal terminating strategy and trajectory $\{x_j^*, u_j^*\}$ from x_i is unique and if this problem is LMR regular and has unique multipliers (p^*, λ^*, μ^*) then RUB_i is continuously differentiable at x_i and $\nabla RUB_i(x_i) = p_i$.*

Proof. The result follows as in the proof of the above Proposition 2.7.1 by considering the truncated problem in the formulation (2.29) - (2.31). □

The problem where the local constraints are independent of x_i and the end point is free is an important special case.

Proposition 2.7.3 *Assume that r_j, f_j, g_j and h_j are continuously differentiable for $i \leq j \leq N-1$, and that g_j and h_j are independent of x_j for $j = i, \ldots, N-1$ and that the end point is free. Consider the truncated problem starting from $x_i \in RY_i$ and assume that it is tame. If the optimal terminating strategy and trajectory $\{x_j^*, u_j^*\}$ from x_i is unique then RUB_i is continuously differentiable at x_i and $\nabla RUB_i(x_i) = p_i$.*

Proof. The expression for ∇RUB_i is $\nabla_{x_i}(H_i(x_i,u_i^*,p_{i+1}^*) - \lambda_i^* g_i(x_i,u_i^*) - \mu_i^* h_i(x_i,u_i^*))$. If g_j and h_j are independent of x_i then this expression yields a unique value, even if the multipliers λ_j and μ_j are not unique. □

For the smaller upper boundaries we have:

Proposition 2.7.4 *Assume that r_i, f_i, g_i and h_i are Lipschitz continuous. Consider ub_i^{i+1} at $(x_i,x_{i+1}) \in W_i^{i+1}$ and assume that the problem defining ub_i^{i+1} is tame. If the optimal u_i^* in the determination of $ub_i^{i+1}(x_i,x_{i+1})$ is unique and this problem is LMR regular and has unique multipliers $(p_{i+1}^*,\lambda_i^*,\mu_i^*)$ then ub_i^{i+1} is continuously differentiable at (x_i,x_{i+1}) with*

$$\nabla_{x_i} ub_i^{i+1}(x_i,x_{i+1}) = \partial_{x_i}(H_i(x_i,u_i^*,p_{i+1}^*) - \lambda_i^* g_i(x_i,u_i^*) - \mu_i^* h_i(x_i,u_i^*))$$

and

$$\nabla_{x_{i+1}} ub_i^{i+1}(x_i,x_{i+1}) = -p_{i+1}^*$$

Proof. The result follows as in the proof of the above Proposition 2.7.1 by considering the truncated problem in the formulation (2.29) - (2.31). Defining the Lagrangian to the problem determining u_i^* as

$$L(x_i,u_i,p_{i+1},\lambda_i,\mu_i) = r_i(x_i,u_i) + p_{i+1} f_i(x_i,u_i) - \lambda_i g_i(x_i,u_i) - \mu_i h_i(x_i,u_i)$$

it follows that

$$\nabla_{x_i} ub_i^{i+1}(x_i,x_{i+1}) = \partial_{x_i} L = \partial_{x_i}(H_i(x_i^*,u_i^*,p_{i+1}^*) - \lambda_i^* g_i(x_i^*,u_i^*) - \mu_i^* h_i(x_i^*,u_i^*))$$

and

$$\nabla_{x_{i+1}} ub_i^{i+1}(x_i,x_{i+1}) = \partial_{x_{i+1}} L = -p_{i+1}^*$$

□

The first of the expressions of Proposition 2.7.4 has been seen before. Defining p_i^* as in (2.41) we see that the relations of Proposition 2.7.4 may be written:

$$\nabla_{x_i} ub_i^{i+1}(x_i,x_{i+1}) = p_i^* \tag{2.64}$$

$$\nabla_{x_{i+1}} ub_i^{i+1}(x_i,x_{i+1}) = -p_{i+1}^* \tag{2.65}$$

Here the last expression implies

$$\nabla_{x_i} ub_{i-1}^{i}(x_{i-1},x_i) = -p_i^* \tag{2.66}$$

This links the local approach of the ub's with the approach of the adjoint equations.

Example 2.7.1 *In relation to Example 2.5.4 it may be verified that the optimal solution for u_2 in the determination of RUB_2 is as follows:*

$$(u_2^1,u_2^2) = \begin{cases} (4,1-x_2) & \text{if } x_2 \le 1 \\ (5-x_2,0) & \text{if } 1 \le x_2 \le 4 \\ (9-2x_2,x_2-4) & \text{if } 4 \le x_2 \end{cases} \tag{2.67}$$

Consequently we get RUB_2:

$$RUB_2(x_2) = \begin{cases} -24+8x_2 & \text{if } x_2 \le 1 \\ -(5-x_2)^2 & \text{if } 1 \le x_2 \le 4 \\ -(2x_2-7)^2 & \text{if } 4 \le x_2 \end{cases} \tag{2.68}$$

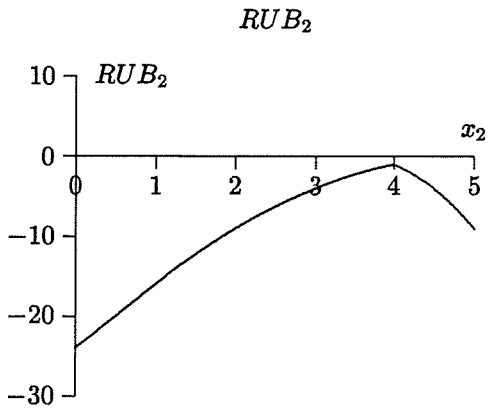


Figure 2.6: A nonsmooth RUB_i

At $x_2 = 4$ RUB_2 is not differentiable but has a "downwards kink", cf. Figure 2.6. As shown in Example 2.5.4 this may be attributed to the non-uniqueness of the KKT multipliers in the problem involved in determining $RUB_2(4)$.

On the other hand it may be verified that for any x_2 the optimal strategy leading from $\underline{x}_0$ to x_2 is $(u_0, u_1) = (x_2/2, x_2/2)$. This implies $UB_2(x_2) = -(x_2)^2/4$. Therefore UB_2 is continuously differentiable in accordance with the unique strategy leading to x_2 and the corresponding unique (p^*, λ^*, μ^*) (there are actually no (λ, μ) in this because there are no local constraints for $i = 0$ and $i = 1$). □

In accordance with the stagewise structure of the OCP the analysis in this chapter has the state variables as parameters in the analysis. However, the analysis may also be performed with respect to other parameters. An obvious example is parametrization with respect to the right hand sides of the local constraints (2.25) - (2.28), but also the criterion and dynamic functions may contain parameters. As above, the essential elements for the analysis are tameness, constraint qualifications and uniqueness of solution and multipliers.

2.8 Twice Continuous Differentiability

For the problem (2.29) - (2.31) on page 74 we may formulate the second order sufficient conditions for (local) unique optimality of v^* at a given $z^\diamond$ as follows:

$$y' \nabla^2_{vv} L(z^\diamond, v^*, \lambda^*, \mu^*) y < 0 \tag{2.69}$$

for all $y \in R^a$, $y \neq 0$, satisfying

$$\nabla_v g_j(z^\diamond, v^*) y = 0 \text{ if } \lambda^*_j > 0 \tag{2.70}$$

$$\nabla_v g_j(z^\diamond, v^*) y \leq 0 \text{ if } \lambda^*_j = 0 \tag{2.71}$$

$$\nabla_v h_j(z^\diamond, v^*) y = 0 \tag{2.72}$$

Here again L is the Lagrangian

$$L(z, v, \lambda, \mu) = r(z, v) - \lambda g(z, v) - \mu h(z, v) \tag{2.73}$$

It is here assumed that the functions are twice continuously differentiable with respect to (z, v).

Define $\tilde{g}$ as the set of active g_j (i.e., those g_j for which $g_j(z^\diamond, v^*) = 0$).

It is now further assumed that the gradients with respect to v of h and $\tilde{g}$ are linearly independent.

We say that *strict complementarity* is fulfilled if $g_j(z^\diamond, v^*) = 0$ implies $\lambda_j^* > 0$. Under this assumption (2.71) is irrelevant and the analysis simplifies.

Under these assumptions, v^* is locally unique and the Lagrange multipliers (λ^*, μ^*) to (2.30) - (2.31) are unique.

This hold for all z in a neighborhood of $z^\diamond$, i.e. we may write $(v^*, \lambda^*, \mu^*) = (v^*(z), \lambda^*(z), \mu^*(z))$. Moreover, the functions (v^*, λ^*, μ^*) are continuously differentiable. Further, for all z in a neighborhood of $z^\diamond$ the assumptions are reproduced, i.e., (1) the second order sufficient conditions hold, (2) the set of active constraints remains the same, (3) strict complementarity holds, and (4) the gradients with respect to v of h and $\tilde{g}$ are linearly independent.

It is possible to give an explicit statement of an approximation to $(v^*(z), \lambda^*(z), \mu^*(z))$ as follows. Let the dimensions of the matrices and functions be as follows, v is $a \times 1$, z is $b \times 1$, $\tilde{g}$ is $k \times 1$, h is $\ell \times 1$, $\tilde{\lambda}$ (corresponding to $\tilde{g}$) is $k \times 1$ and μ (corresponding to h) is $\ell \times 1$. Define the following matrices of dimensions $(a + k + \ell) \times (a + k + \ell)$ and $(a + k + \ell) \times b$ with derivatives taken at $(z^\diamond, v^*, \tilde{\lambda}^*, \mu^*)$:

$$M = \begin{pmatrix} -\nabla_{vv}^2 L & \nabla_v \tilde{g}' & \nabla_v h' \\ \nabla_v \tilde{g} & 0 & 0 \\ \nabla_v h & 0 & 0 \end{pmatrix} \tag{2.74}$$

$$N = \begin{pmatrix} \nabla_{zv}^2 L \\ -\nabla_z \tilde{g} \\ -\nabla_z h \end{pmatrix} \tag{2.75}$$

Then

$$\begin{pmatrix} v^*(z) \\ \tilde{\lambda}^*(z)' \\ \mu^*(z)' \end{pmatrix} = \begin{pmatrix} v^*(z^\diamond) \\ \tilde{\lambda}^*(z^\diamond)' \\ \mu^*(z^\diamond)' \end{pmatrix} + M^{-1}N(z - z^\diamond) + o(\| z - z^\diamond \|) \tag{2.76}$$

where $o(\epsilon)/\epsilon \to 0$ as $\epsilon \to 0$. This implies

$$\begin{pmatrix} \nabla_z v^*(z^\diamond) \\ \nabla_z \tilde{\lambda}^*(z^\diamond)' \\ \nabla_z \mu^*(z^\diamond)' \end{pmatrix} = M^{-1}N \tag{2.77}$$

Further, ub is twice continuously differentiable at $z^\diamond$, and

$$\begin{aligned} \nabla^2 ub(z^\diamond) = \quad & \nabla_{vz}^2 L(z^\diamond, v^*, \tilde{\lambda}^*, \mu^*) \nabla_z v^*(z^\diamond) \\ & -\nabla_z \tilde{g}(z^\diamond, v^*)' \nabla_z \mu(z^\diamond) - \nabla_z h(z^\diamond, v^*)' \nabla_z \tilde{\lambda}(z^\diamond) \\ & +\nabla_{zz} L(z^\diamond, v^*, \tilde{\lambda}^*, \mu^*) \end{aligned} \tag{2.78}$$

An alternative expression for $\nabla^2 ub$, convenient for subsequent development, may be derived as follows. Define the quadratic-linear approximation to the problem (2.29) - (2.31), where all derivatives

are taken at $(z^\diamond, v^*, \tilde{\lambda}^*, \mu^*)$ and $\hat{z} = z - z^\diamond$, $\hat{v} = v - v^*$:

$$\max_v [\tfrac{1}{2}\hat{z}'\nabla^2_{zz}L\hat{z} + \hat{z}'\nabla^2_{zv}L\hat{v} + \tfrac{1}{2}\hat{v}'\nabla^2_{vv}L\hat{v} + \nabla_z r\hat{z} + \nabla_v r\hat{v}] \tag{2.79}$$

$$\nabla_z h\hat{z} + \nabla_v h\hat{v} = 0 \tag{2.80}$$

$$\nabla_z \tilde{g}\hat{z} + \nabla_v \tilde{g}\hat{v} = 0 \tag{2.81}$$

By appropriate definition of matrices this may, for later reference, be formulated as:

$$\max_v [z'Az + z'Bv + v'Cv + Dv + Ez] \tag{2.82}$$

$$H^z z + H^v v - \overline{h} = 0 \tag{2.83}$$

Obviously this problem has the same solution as (2.29) - (2.31).

Under the assumptions of strict complementarity, linearly independent gradients of $\tilde{g}$ and h and second order sufficient conditions for the problem (2.29) - (2.31), the rows of H^v will be linearly independent and the second order sufficiency conditions will be fulfilled for the problem (2.82) - (2.83).

The linear approximation to the solution (v, μ) given as in (2.76) holds exactly in this case:

$$\begin{pmatrix} v^*(z) \\ \mu^*(z)' \end{pmatrix} = \begin{pmatrix} v^*(z^\diamond) \\ \mu^*(z^\diamond)' \end{pmatrix} + M^{-1}N(z - z^\diamond) \tag{2.84}$$

$$\equiv \begin{pmatrix} v^*(z^\diamond) \\ \mu^*(z^\diamond)' \end{pmatrix} + \begin{pmatrix} (M^{-1}N)^v \\ (M^{-1}N)^\mu \end{pmatrix}(z - z^\diamond)$$

where M and N, defined in (2.74) - (2.75), are readily obtained from (2.82) - (2.83) as

$$M = \begin{pmatrix} -2A & H^{v\prime} \\ H^v & 0 \end{pmatrix} \tag{2.85}$$

$$N = \begin{pmatrix} B \\ -H^z \end{pmatrix} \tag{2.86}$$

Since (2.83) holds for all z we see using (2.84) that

$$H^z = -H^v(M^{-1}N)^v \tag{2.87}$$

The KKT stationarity condition relative to (2.82) - (2.83) specifies

$$B'z + 2Cv^*(z) + D' - H^{v\prime}\mu^*(z) = 0 \tag{2.88}$$

This holds for all z, therefore using (2.84) there holds

$$B' + 2C(M^{-1}N)^v - H^{v\prime}(M^{-1}N)^\mu = 0 \tag{2.89}$$

Multiplying (2.89) by $(M^{-1}N)^{v\prime}$ we find

$$(M^{-1}N)^{v\prime}B' + 2(M^{-1}N)^{v\prime}C(M^{-1}N)^v = (M^{-1}N)^{v\prime}H^{v\prime}(M^{-1}N)^\mu \tag{2.90}$$

and using (2.87) this implies

$$(M^{-1}N)^{v\prime}B' + 2(M^{-1}N)^{v\prime}C(M^{-1}N)^v = -H^{z\prime}(M^{-1}N)^\mu \tag{2.91}$$

In the notation (2.82) - (2.83), (2.78) may be written

$$\nabla^2 ub(z^\diamond) = B(M^{-1}N)^v - H^{z\prime}(M^{-1}N)^\mu + 2A \tag{2.92}$$

which, using (2.91) may be reformulated as

$$\nabla^2 ub(z^\diamond) = \tag{2.93}$$
$$2A + B(M^{-1}N)^v + (M^{-1}N)^{v\prime}B' + 2(M^{-1}N)^{v\prime}C(M^{-1}N)^v$$

The conclusion of this is the following. In order to obtain the second order approximation of the upper boundary function it suffices to (1) solve the problem and find the Lagrange multipliers from the first order stationarity conditions, (2) define the problem (2.79) - (2.81) or (2.82) - (2.83) with quadratic criterion and linear equality constraints, and (3) insert the linear expression for the optimal solution, cf. the upper part of (2.84) into the quadratic criterion function (2.82).

We may apply the results directly to the OCP formulated in the form (2.29) - (2.31), cf. also the explanation following (2.31). For the truncated problem ending at x_i, x_i corresponds to $z^\diamond$ and we consider all the dynamic equations, all the local constraints and all the variables (x_j, u_j) for $j < i$. For the truncated problem starting at x_i, x_i corresponds to $z^\diamond$ and we consider all the dynamic equations and all the local constraints for $j \geq i$, the variable u_i, the variables (x_j, u_j) for $i+1 \leq j \leq N-1$, and the variable x_N. For the problem defining ub_i^{i+1}, $(x_i', x_{i+1}')'$ corresponds to $z^\diamond$ and we consider the dynamics and the local constraints at stage i and the variable u_i.

Proposition 2.8.1 *Assume that for the problem (2.29) - (2.31) all functions are twice continuously differentiable. Assume that for a given $z^\diamond$ and corresponding locally optimal solution the KKT multipliers are unique, that strict complementarity holds and that the second order sufficient conditions (2.69) - (2.72) hold. Then for all z in a neighborhood of $z^\diamond$ we have: (1) All the above assumptions hold, (2) The optimal solution v^* is locally unique and it is given as a continuously differentiable function of z with derivative as in (2.77), and (3) All the KKT multipliers corresponding to v^* are unique and given as continuously differentiable functions of z with derivatives as in (2.77).*

If the problem is tame and the solution is globally unique then ub is twice continuously differentiable at $z^\diamond$. A second order approximation to ub may be obtained as described above.

If the conditions above hold, with the assumed order of differentiability being $k+1$ ($k \geq 1$), then in a neighborhood of $z^\diamond$, $(v^*(z), \lambda^*(z), \mu^*(z))$ is k times continuously differentiable and ub is $k+1$ times continuously differentiable.

By suitable interpretation this also applies to any of the problems defining UB_i, RUB_i and ub_i^{i+1}.

Explicit formulae for $\nabla^2 UB_i$, $\nabla^2 RUB_i$ and $\nabla^2 ub_i^{i+1}$ will be given in (4.82), (4.31) and (5.34).

Proof. The results for the problem (2.29) - (2.31), in particular (2.76) and (2.78), are given in Fiacco (1976), in relation to a local solution. The extension to global optimality and the optimal value function is immediate under the assumptions. The derivation of the expression (2.93) was done above. □

2.9 Conclusions

We have given the basic properties of the upper boundaries in terms of upper semicontinuity, continuity, concavity, Lipschitz continuity, and one or more times continuous differentiability. Such

properties are essential for the analysis of an optimization problem. The types of results are similar to those of mathematical programming. However, the structure of the OCP has been exploited to derive result not only for the whole OCP but also for any truncated problem consisting of a subsequence of stages. In particular the truncated problems consisting of stages $j = 0,\ldots,i$, $j = i,\ldots,N$ (greater upper boundaries) and $j = i, i+1$ (smaller upper boundaries) have been considered.

It has been shown that the stagewise approach to the development of such properties is indeed possible. However, another major conclusion is that properties that hold for an OCP need not hold for a truncated problem. Indeed, it is seen that for some OCP's this will in fact be the case; the discussion on constraint qualifications in Section 2.5 is essential in this respect.

To the extent that a truncated problem is indeed the focus of the analysis there is obviously nothing to do about this. Thus, if for instance we consider slight changes in the right hand side $\underline{x}_i$ of a constraint $x_{\leq}\underline{x}_i$ then it may of course be essential whether UB_i and/or RUB_i are smooth at $\underline{x}_i$. However, if the focus is on obtaining the optimal solution to the whole OCP then it would be desirable to use methods that are independent of properties of intermediate functions such as UB_i and RUB_i. Numerical methods based on the stagewise approach (such as dynamic programming and the maximum principle) may be disadvantageous since they rely on stagewise properties.

This severe limitation for the applicability of the stagewise approach will be further discussed in later chapters. In the next chapter we present optimality conditions in a stagewise formulation and show, among other things, that the properties of the upper boundaries are essential for this.

Chapter 3

Optimality and Maximum Principles

Necessary and sufficient optimality conditions are essential where an optimization perspective is taken towards problem modeling and analysis. In the optimal control literature many important optimality conditions are formulated as "maximum principles", cf. Section 1.3. Initially formulated in close analogy to the continuous time maximum principle of Pontryagin et al. (1962), the discrete time maximum principle is the best known one. This principle is formulated using the costate vector p_i. As pointed out in Section 3.4, it is necessary to generalize this to nonlinear functions π_i if the maximum principle shall hold as necessary optimality condition.

The classical maximum principle was established under the assumptions of smooth functions and suitable convexity properties, cf. Section 1.3, where also generalizations in relation to this are mentioned.

In this chapter we present and relate optimality principles and we relate them to the basic concepts of the upper boundaries, introduced in Chapter 2. We shall in particular be interested in pointing out results which link dynamic programming and maximum principle ideas between them as well as with the upper boundaries.

As will be seen, there are strong relationships between the properties of the upper boundaries, as analyzed in the previous chapter, and the properties of the auxiliary functions π_i which are used in the maximum principles. Further, as the upper boundaries are used in dynamic programming, this implies similarly strong relationships between dynamic programming and maximum principles.

We have grouped the various optimality principles according to the following characteristics. Dynamic programming works directly with the upper boundaries, i.e., no approximation is made to these. Maximum principles, on the other hand, work with auxiliary functions that may be seen as approximations to the upper boundaries. In the classical setting, the maximum principles have two parts, such that maximization is performed with respect to the control variables, while only stationarity is required for the state variables. As a variation of this, the extended maximum principle has maximization taking place with respect to both state and control variables; Lagrangian relaxation may be seen as a special case of this. With respect to dynamic programming we observe that for backwards dynamic programming, optimization is with respect to the control variables only (the state variables are treated as parameters) while in forwards dynamic programming the maximization is with respect to both state and control variables simultaneously.

We initialize in Section 3.1 with the principle of optimal evolution, while the closely related

dynamic programming and the principle of optimality are treated in the following Section 3.2. Then the smaller upper boundaries and the linkage to the greater upper boundaries, in particular through the global maximum principle, are treated in Section 3.3. The maximum principle with maximization in the control variables only is treated in Section 3.4 where it is shown that the classical maximum principle with linear auxiliary functions π_i is but a special case. A major point is the development of the adjoint relations when π_i is nonsmooth. The extended maximum principle, where maximization at stage i is performed simultaneously in u_i and x_i, is dealt with in Section 3.5. Finally, in Section 3.6 the duality framework is applied. Basic results are derived in a setting of non-linear dual (price) functions and it is shown that some of the previous results, particularly from Section 3.5, may be interpreted in the duality perspective. Since the results of Section 3.5 in turn relate heavily with those of the previous sections, the duality perspective is essential for the whole chapter.

3.1 Principle of Optimal Evolution

As already pointed out in Section 1.3, the observation that an optimal $\hat{x}_i^*$ must necessarily be situated at $\{UB_i\}$ was made in Halkin (1964) and named the *principle of optimal evolution.* As the name suggests, this result is linked to the forward direction point of view. Similar result holds for the backwards direction. Recall from Section 2.1 the definitions of the extended states $\hat{x}_i$ (forwards) and $\check{x}_i$ (backwards).

Proposition 3.1.1 *Assume that OCP has an optimal solution with criterion value R^*. Then*

- *$\hat{x}_i^*$ is optimal if and only if $\hat{x}_i^*$ is situated at $\{UB_i\}$ and $(R^* - \hat{x}_i^{0*}, x_i^{*\prime})'$ is situated at $\{RUB_i\}$.*
- *$\check{x}_i^*$ is optimal if and only if $(R^*, 0')' - \check{x}_i^*$ is situated at $\{UB_i\}$ and $(R^* - \check{x}_i^{0*}, x_i^{*\prime})'$ is situated at $\{RUB_i\}$.*

Proof. Assume that $\hat{x}_i^*$ is optimal, but not situated at $\{UB_i\}$. Optimality implies that there is an admissible strategy and trajectory $\{(u_j^*, x_j^*)\}$ including x_i^*. If $\hat{x}_i^*$ is not situated at $\{UB_i\}$ then $\hat{x}_i^0 = \sum_{j=0}^{i-1} r_j(x_j^*, u_j^*) < UB_i(x_i^*)$. (The relation $\sum_{j=0}^{i-1} r_j(x_j^*, u_j^*) > UB_i(x_i^*)$ is not possible due to the definition of UB_i.) By definition of UB_i there is an admissible strategy and trajectory $\{(u_j^\diamond, x_j^\diamond)\}$ leading to x_i^* and having $\sum_{j=0}^{i-1} r_i(x_i^\diamond, u_i^\diamond) = UB_i(x_i^*)$. Now choose $(x_j^\diamond, u_j^\diamond) = (x_j^*, u_j^*)$ for $j = i, \ldots, N-1$ and $x_N^\diamond = x_N^*$. This gives an admissible strategy and trajectory. Then we have $\sum_{j=0}^{N-1} r_j(x_j^*, u_j^*) + r_N(x_N^*) < \sum_{j=0}^{N-1} r_j(x_j^\diamond, u_j^\diamond) + r_N(x_N^\diamond)$, contradicting the assumption that x_i^* was optimal. Therefore $\hat{x}_i^*$ is situated at $\{UB_i\}$.

Assume now that $\hat{x}_i^*$ is optimal, but that $(R^* - \hat{x}_i^{0*}, x_i^{*\prime})'$ is not situated at $\{RUB_i\}$. If $(R^* - \hat{x}_i^{0*}, x_i^{*\prime})'$ is not situated at $\{RUB_i\}$ then $R^* - \hat{x}_i^{0*} = R^* - \sum_{j=i}^{N-1} r_j(x_j^*, u_j^*) - r_N(x_N^*) > R^* - RUB_i(x_i^*)$ for the admissible terminating strategy and trajectory $\{(x_i^*, u_i^*)\}$. (The relation $R^* - \hat{x}_i^{0*} = R^* - \sum_{j=i}^{N-1} r_j(x_j^*, u_j^*) - r_N(x_N^*) < R^* - RUB_i(x_i^*)$ is not possible due to the definition of RUB_i.) As above it can then be argued, that a strategy identical to $\{(x_j^*, u_j^*)\}$ for $j = 0, \ldots, i-1$, and optimal for $j = i, \ldots, N$, including x_i^*, will give a higher criterion value that the strategy and trajectory $\{(x_j^*, u_j^*)\}$. This contradicts the assumption that $\hat{x}_i^*$ is optimal, and therefore $(R^* - \hat{x}_i^{0*}, x_i^{*\prime})'$ is situated at $\{RUB_i\}$.

We now assume that $\hat{x}_i^*$ is situated at $\{UB_i\}$ and $(R^* - \hat{x}_i^{0*}, x_i^{*\prime})'$ is situated at $\{RUB_i\}$. This means that there is an admissible strategy and trajectory $\{(x_i^*, u_i^*)\}$, including x_i^*, such that $\sum_{j=0}^{N-1} r_j(x_j^*, u_j^*) + r_N(x_N^*) = UB_i(x_i^*) + RUB_i(x_i^*) = R^*$. Therefore $\hat{x}_i^*$ is optimal.

The second part of the Proposition is shown in a way similar to the above. □

The principle of optimal evolution therefore can be interpreted to mean that if a strategy and trajectory is optimal for the OCP then it is optimal for any truncated problem ending at x_i^* and any truncated problem starting from x_i^*. Therefore the two directions are beautifully linked as follows.

Proposition 3.1.2 *Assume that OCP has an optimal solution with criterion value R^*. Then $Y_i \cap RY_i \neq \emptyset$ and*

$$UB_i(x_i) + RUB_i(x_i) \leq R^*$$

for all $x_i \in Y_i \cap RY_i$ with equality at x_i^* if and only if x_i^* is optimal.

Proof. If $Y_i \cap RY_i = \emptyset$ then no feasible solution exists, contradicting the assumption.

Assume therefore that there were an $x_i^\diamond \in X_i \cap RY_i$ such that $UB_i(x_i^\diamond) + RUB_i(x_i^\diamond) > R^*$. The condition $x_i^\diamond \in X_i \cap RY_i$ implies that there is an admissible strategy and trajectory $\{(x_i^\diamond, u_i^\diamond)\}$, including $x_i^\diamond$, such that $\sum_{i=0}^{N-1} r_i(x_i^\diamond, u_i^\diamond) + r_N(x_N^\diamond) = UB_i(x_i^\diamond) + RUB_i(x_i^\diamond) > R^*$. This contradicts the assumption that R^* was the optimal criterion value in the problem. Therefore the relation holds.

If the relation holds with equality at x_i^* then with the same argumentation x_i^* is optimal. And if x_i^* is optimal then $x_i^* \in Y_i \cap RY_i$ and the relation must hold with equality, since otherwise the trajectory including x_i^* would not be optimal.

Observe that the restriction to $Y_i \cap RY_i$ may be omitted if we let $UB_i(x_i) = -\infty$ for $x_i \notin Y_i$ and $RUB_i(x_i) = -\infty$ for $x_i \notin RY_i$ and if there exists a feasible solution. □

We see that the above result may also be formulated as follows: x_i^* is optimal if and only if it maximizes $[UB_i(x_i) + RUB_i(x_i)]$ over $x_i \in Y_i \cap RY_i$.

Example 3.1.1 *If the OCP is to be solved on two parallel processors, one may find UB_i while the other finds RUB_i. Here i is taken as approximately $\frac{1}{2}N$. Then x_i^* is found by solving the problem* $\max[UB_i(x_i) + RUB_i(x_i)]$ *subject to $x_i \in Y_i \cap RY_i$. Finally the two processors reconstruct the optimal truncated strategies and trajectories leading to x_i^* and starting from x_i^*, respectively.* □

Example 3.1.2 *The result of the above Example 3.1.1 may be used as follows. Suppose that for $0 \leq j \leq i-1$ the functions r_i and f_i and the sets V_i have a special, common structure, that makes it relatively easy to find UB_i. Suppose also that for $i \leq j \leq N-1$ the functions r_j and f_j and the sets V_j, together with r_N and V_N, have another special, common structure, that makes it relatively easy to find RUB_i. Then the application of the alternative formulation of Proposition 3.1.2 may be advantageous.* □

3.2 Dynamic Programming and the Principle of Optimality

The following two results are the backbones of the recursive solution technique of dynamic programming.

Proposition 3.2.1 *For all $x_i \in RY_i$ there holds for $i = N-1, \ldots, 0$*

$$RUB_i(x_i) = \max_{u_i}[r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i))]$$

$$u_i \in U_i(x_i)$$

$$f_i(x_i, u_i) \in RY_{i+1}$$

For an OCP with fixed initial point $\underline{x}_0$, (x^*, u^*) *is optimal if and only if* u_i^* *maximizes as above,* $x_0^* = \underline{x}_0$ *and* $x_{i+1}^* = f_i(x_i^*, u_i^*)$ *for i=0,...,N-1.*

Proof. Assume that RUB_{i+1} is given correctly. The restrictions on the problem in the Proposition assure that there is an admissible terminating strategy and trajectory from x_i. Now assume that we could get a higher value of RUB_i from the recursive formula than from using the definition in Section 2.1 by using a u_i° satisfying the restrictions. This means $r_i(x_i, u_i^\circ) + RUB_{i+1}(f_i(x_i, u_i^\circ)) > RUB_i(x_i)$. Since there is an admissible terminating strategy and trajectory, starting with u_i° from x_i, this contradicts the definition of $RUB_i(x_i)$ and the assumptions that RUB_{i+1} was correct. Now assume that we could only get a lower value from the recursive formula than from the definition. In the same way it is seen that this lead to a contradiction of the definition of RUB_i and the assumption the RUB_{i+1} was correct. We therefore conclude that we get the correct value for RUB_i by using the recursive formula.

For $i = N$, RUB_N is given directly as $RUB_N(x_N) = r_N(x_N)$ on V_N. Therefore the recursive formula holds for $i = N - 1$. By induction it is seen to hold for all i.

Suppose now that (x^*, u^*) is optimal. Assume that there is a largest index i such that u_i^* does not maximize the above. Let u_i° be a maximizing u_i; it exists as x_i^* was assumed optimal. We then have

$$\begin{aligned} RUB_i(x_i^*) &= r_i(x_i^*, u_i^\circ) + RUB_{i+1}(f_i(x_i^*, u_i^\circ)) \\ &> r_i(x_i^*, u_i^*) + RUB_{i+1}(f_i(x_i^*, u_i^*)) \end{aligned}$$

However this shows that there is a strategy and trajectory from x_i^*, starting with u_i° which yields a higher criterion value than the assumed optimal terminating strategy and trajectory starting with u_i°. Therefore (x^*, u^*) cannot be optimal. As this contradicts the assumption, u_i^* is maximizing. This holds for all i. Moreover we have $x_{i+1}^* = f_i(x_i^*, u_i^*)$ as (x^*, u^*) could not be optimal if it did not satisfy the dynamic equation.

Now suppose u_i^* maximizes the above and $x_{i+1}^* = f_i(x_i^*, u_i^*)$ for $i = 0, \ldots, N-1$. Then (x^*, u^*) is optimal because it is feasible and by the definition of RUB_i we have $\sum_{i=0}^{N-1} r_i(x_i^*, u_i) + r_N(x_N^*) = RUB_0(\underline{x}_0)$. □

It is seen that the restriction $f_i(x_i, u_i) \in RUB_{i+1}$ may be omitted if we let $RUB_{i+1}(x_{i+1}) = -\infty$ for $x_{i+1} \notin RY_{i+1}$ and there exists an optimal solution to the OCP.

For a problem where the initial point is not fixed we must at stage 0 solve the following problem:

$$\max_{x_0, u_0} [r_0(x_0, u_0) + RUB_1(f_0(x_0, u_0))] \tag{3.1}$$

$$(x_0, u_0) \in V_0 \tag{3.2}$$

$$f_0(x_0, u_0) \in RY_1 \tag{3.3}$$

in order to find x_0^*. This problem may be separated in two: First find RUB_0 for all $x_0 \in RY_0$; second find x_0^* by solving $\max[RUB_0(x_0)]$, $x_0 \in RY_0$. With these modifications the above Proposition 3.2.1 extends also to this problem.

Proposition 3.2.2 *For all x_{i+1} in Y_{i+1} there holds for i=0,...,N-1*

$$UB_{i+1}(x_{i+1}) = \max_{x_i,u_i}[r_i(x_i,u_i) + UB_i(x_i)]$$

$$(x_i,u_i) \in V_i$$

$$x_i \in Y_i$$

$$x_{i+1} = f_i(x_i,u_i)$$

For a problem with fixed final point $\underline{x}_N$, (x^*,u^*) is optimal if and only if (x_i,u_i^*) maximizes as above, $x_{i+1}^* = f_i(x_i^*,u_i^*)$ for i=0,...,N-1 and $x_N^* = \underline{x}_N$.

Proof. The proof is similar to the proof of the previous Proposition 3.2.1. It is observed that UB_0 is correctly given, and then the result follows by induction. □

For a problem where the final point is not fixed we must at stage N solve the problem

$$\max_{x_N}[UB_N(x_N)] \tag{3.4}$$

$$x_N \in V_N \cap Y_N \tag{3.5}$$

in order to find x_N^*. With this modification the above proposition extends also to this problem.

In general the problems in the Propositions are difficult to solve. This is mainly due to the fact that UB_i and RUB_i are normally complicated functions. If there is some special structure, is may be easier; particularly the problem with quadratic criterion and linear dynamics may be easy, see Section 4.2. See also Example 3.1.2 above. In case of discrete problems, the brute force of complete enumeration is often used to solve the local problems in the recursions.

We observe that the maximization in Proposition 3.2.2 is with respect to (x_i,u_i) while in Proposition 3.2.1 it is only with respect to u_i. It might appear easier then to use the backwards than the forwards recursion when using dynamic programming. On the other hand the maximization in Proposition 3.2.1 is subject to the constraint $f_i(x_i,u_i) \in RY_{i+1}$ which may be difficult to handle. It will depend on the precise problem formulation whether the forwards or the backwards direction is easier.

The next two results which are restatements of the first parts in the above propositions will be useful in later results.

Proposition 3.2.3 *Let (x^*,u^*) be optimal in OCP. Then for $i = 0,\ldots,N-1$, (x_i^*,u_i^*) maximize*

$$[r_i(x_i,u_i) + UB_i(x_i) - UB_{i+1}(f_i(x_i,u_i))]$$

subject to the restrictions $(x_i,u_i) \in V_i$ and $x_i \in Y_i$, and x_N^* maximizes UB_N over $V_N \cap Y_N$.

Proof. Observe that $(x_i,u_i) \in V_i$ and $x_i \in Y_i$ implies $f_i(x_i,u_i) \in Y_{i+1}$. Suppose that (x_i^*,u_i^*) were not maximizing. Then there would be a $(x_i^\diamond,u_i^\diamond)$ satisfying the restrictions involved, such that

$$r_i(x_i^\diamond,u_i^\diamond) + UB_i(x_i^\diamond) - UB_{i+1}(f_i(x_i^\diamond,u_i^\diamond)) >$$

$$r_i(x_i^*,u_i^*) + UB_i(x_i^*) - UB_{i+1}(f_i(x_i^*,u_i^*))$$

Since by Proposition 3.2.2 $r_i(x_i^*,u_i^*) + UB_i(x_i^*) - UB_{i+1}(f_i(x_i^*,u_i^*)) = 0$ this implies $r_i(x_i^\diamond,u_i^\diamond) + UB_i(x_i^\diamond) > UB_{i+1}(f_i(x_i^\diamond,u_i^\diamond))$, contradicting Proposition 3.2.2.

Finally, if x_N^* were not maximizing UB_N over $V_N \cap Y_N$ then there would be another $x_N^\circ \in V_N \cap Y_N$ such that $UB_N(x_N^\circ) > UB_N(x_N^*)$, contradicting the assumption that x_N^* were optimal. □

Proposition 3.2.4 *Let (x^*, u^*) be optimal in OCP. Then for $i = 0, \ldots, N-1$, (x_i^*, u_i^*) maximize*

$$[r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i)) - RUB_i(x_i)]$$

subject to the restrictions $(x_i, u_i) \in V_i$ and $f_i(x_i, u_i) \in RY_{i+1}$.

Proof. The proof is similar to the proof of Proposition 3.2.3 above. □

As seen, an essential feature of the OCP is the partial separability of the criterion. This means that the contribution to the criterion from stage i can be written as $r_i(x_i, u_i)$, i.e., the value is independent of (x_j, u_j), $j < i$ - as long as the strategy and trajectory $\{(x_j, u_j)\}$ up to stage i leads to x_i. Similarly the contribution from stage i is independent of (x_j, u_j), $i < j$ - as long as the terminating strategy and trajectory $\{(x_j, u_j)\}$ from stage i starts with x_i.

There is also a partial separability of the constraints. An admissible strategy and trajectory leading to x_i remains admissible up to stage i, irrespective of how the terminating strategy and trajectory from x_i is chosen. An admissible terminating strategy and trajectory from x_i remains admissible, irrespective of the choice of strategy and trajectory up to x_i. The partial separability of the criterion and the constraints implies that necessary optimality conditions can be formulated for a partial problem. The proofs of the previous two propositions used this special structure.

These observations are versions of the same insight in the nature of the problem, as is captured in Bellman's classical formulation of the *principle of optimality*:

> An optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

See Sniedovich (1992), Chapter 13, for a recent discussion of this formulation.

3.3 Smaller Upper Boundaries. The Global Maximum Principle

We now formulate results that relate the greater upper boundaries, treated above, with the smaller upper boundaries. This means, that the variable u_i has been eliminated and that it is implicitly given from (x_i, x_{i+1}).

The convenience of using the smaller upper boundaries rather than working directly with r_i is that ub_i^{i+1} may be simpler. If $n < m$, ub_i^{i+1} is of dimension $2n$ while r_i is of dimension $n+m$, which in some cases may be much smaller. On the other hand, ub_i^{i+1} will typically be more complicated than r_i, for instance, ub_i^{i+1} may be nonsmooth even if r_i is smooth. This may be attributed to the fact that ub_i^{i+1} shall reflect the constraints $(x_i, u_i) \in V_i$ and $f_i(x_i, u_i) = x_{i+1}$ besides the function r_i itself.

Let us first observe the following elementary relationship to UB_i and RUB_i.

Proposition 3.3.1 *For an OCP with an optimal solution x^* there hold*

$$ub_i^{i+1}(x_i^*, x_{i+1}) - ub_i^{i+1}(x_i^*, x_{i+1}^*) \leq UB_{i+1}(x_{i+1}) - UB_{i+1}(x_{i+1}^*)$$

and

$$ub_i^{i+1}(x_i, x_{i+1}^*) - ub_i^{i+1}(x_i^*, x_{i+1}^*) \leq RUB_i(x_i) - RUB_i(x_i^*)$$

on the sets on which both functions are defined. Equality holds at x_i^*.

Proof. If there were a x_{i+1}^o violating the first relationship this would contradict the definition of UB_{i+1} because this would imply $UB_i(x_i^*) + ub_i^{i+1}(x_i^*, x_{i+1}^o) > UB_{i+1}(x_{i+1}^o)$. The second relation follows via Proposition 3.1.2. If there is an optimal solution x^* then all the functions are defined at x_i^* and if equality did not hold this would contradict the assumption of optimality and/or the definitions of the functions involved. □

Forwards and backwards dynamic programming may then be given formulations that use ub_i^{i+1}.

Proposition 3.3.2 *For all $x_{i+1} \in Y_{i+1}$*

$$UB_{i+1}(x_{i+1}) = \max_{x_i}[ub_i^{i+1}(x_i, x_{i+1}) + UB_i(x_i)]$$

$$x_i \in \{x_i \in R^n \mid U_i(x_i) \neq \emptyset\} \cap Y_i$$

Proof. Observing the definition in Section 2.1 of the smaller upper boundary involved and Proposition 3.2.1 then this is obvious. □

Proposition 3.3.3 *For all $x_i \in RY_i$*

$$RUB_i(x_i) = \max_{x_{i+1}}[ub_i^{i+1}(x_i, x_{i+1}) + RUB_{i+1}(x_{i+1})]$$

$$x_{i+1} \in \{x_{i+1} \in R^n \mid \exists u_i \in U_i(x_i) : x_{i+1} = f_i(x_i, u_i)\} \cap RY_{i+1}$$

Proof. Observing the definition in Section 2.1 of the smaller upper boundary involved and Proposition 3.2.1 then this is obvious. □

From this we get the *Global Maximum Principle* (Vidal (1986)), which links a maximum principle (in the sense that maximization is with respect to the control variables only) with dynamic programming (to secure optimality without assumptions of convexity):

Proposition 3.3.4 *An admissible strategy and trajectory $\{x_i^*, u_i^*\}$ is optimal if and only if for all i it satisfies the two principles*

- *The Maximum Principle:*

  u_i^* maximizes $r_i(x_i^*, u_i)$ subject to $u_i \in U_i(x_i^*)$ and $x_{i+1}^* = f_i(x_i^*, u_i)$

- *The Principle of Optimality:*

  $RUB_i(x_i^*) = \max_{x_{i+1}}[ub_i^{i+1}(x_i^*, x_{i+1}) + RUB_{i+1}(x_{i+1})]$ *over*

  $$x_{i+1} \in \{x_{i+1} \in R^n \mid \exists u_i \in U_i(x_i^*) : x_{i+1} = f_i(x_i, u_i)\} \cap RY_{i+1}$$

Proof. This is proved essentially as in Proposition 3.2.1 observing that by the definition of ub_i^{i+1} we have the same optimal value in

$$\max_{x_{i+1}}[ub_i^{i+1}(x_i, x_{i+1}) + RUB_{i+1}(f_i(x_i, u_i))]$$

$$x_{i+1} \in \{x_{i+1} \mid \exists u_i \in U_i(x_i) : x_{i+1} = f_i(x_i, u_i)\}$$

$$x_{i+1} \in RY_{i+1}$$

as in

$$\max_{u_i}[r_i(x_i, u_i) + RUB_{i+1}(f_i(x_i, u_i))]$$

$$u_i \in U_i(x_i)$$

$$f_i(x_i, u_i) \in RY_{i+1}$$

□

The remaining results in this section are formulated exclusively in terms of ub_i^{i+1}'s. A major disadvantage in this is that the sets W_i^{i+1} on which ub_i^{i+1} are defined are not in general easy to give a convenient representation. In particular, the assumption that $(x_i, x_{i+1}) \in W_i^{i+1}$ for all i does not imply that $\{x_i\}$ constitutes a feasible trajectory, so that not even for this essential property is there an easy relation between the W_i^{i+1}'s and W. In other words, the desirable separability between stages is not obvious. An exception is where there are no local constraints (except possibly linear ones) and the dynamics is linear and non-degenerate, in which case $W_i^{i+1} = R^{2n}$.

Proposition 3.3.5 *A trajectory x^* is optimal in OCP if and only if it maximizes*

$$\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})$$

over all feasible trajectories $\{x_i\}$.

Proof. Consider any state variables x_i^o and x_{i+1}^o in a feasible trajectory $\{x_j\}$. Then obviously $(x_i^o, x_{i+1}^o) \in W_i^{i+1}$.

Assume that x^* is optimal in OCP. If it does not maximize $\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})$ over all feasible trajectories $\{x_i\}$, then there would be another admissible strategy and trajectory $(x_i^\diamond, u_i^\diamond)$ such that $\sum_{i=0}^{N-1} ub_i^{i+1}(x_i^*, x_{i+1}^*) < \sum_{i=0}^{N-1} ub_i^{i+1}(x_i^\diamond, x_{i+1}^\diamond) = \sum_{i=0}^{N-1} r_i(x_i^\diamond, u_i^\diamond) + r_N(x_N^\diamond)$, where $u_i^\diamond$ is the control defining $ub_i^{i+1}(x_i^\diamond, x_{i+1}^\diamond)$. This contradicts the assumption that x^* was optimal and therefore x^* is maximizing $\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})$ over all feasible trajectories $\{x_i\}$.

Now assume that x^* maximizes $\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})$ over all feasible trajectories $\{x_i\}$. If x^* does not solve OCP then there would be another feasible x^o such that $\sum_{i=0}^{N-1} r_i(x_i^\diamond, u_i^\diamond) + r_N(x_N^\diamond) > \sum_{i=0}^{N-1} r_i(x_i^*, u_i^*) + r_N(x_N^*)$. However, for this (x^o, u^o) we would have $\sum_{i=0}^{N-1} ub_i^{i+1}(x_i^\diamond, x_{i+1}^\diamond) > \sum_{i=0}^{N-1} ub_i^{i+1}(x_i^*, x_{i+1}^*)$, contradicting the assumption of x^* being maximizing. Therefore x^* is optimal in OCP. □

A geometrical consequence of this is:

Corollary 1. *If a trajectory x^* is optimal then $\hat{x}^*_{i+1}$ is located at $\{ub_i^{i+1}(x_i^*,.)+\hat{x}_i^{0*}\}$ for all i.*
Proof. This follows from the previous Proposition 3.3.5 and the definition of $\{ub_i^{i+1}(x_i^*,.)+\hat{x}_i^{0*}\}$. □

The following is a simple consequence of the above. Observe the one-way implication between local optimality of x_i^* and the global optimum x^*.

Corollary 2 *If x^* is optimal then x_i^* maximizes $[ub_{i-1}^i(x_{i-1}^*,x_i)+ ub_i^{i+1}(x_i,x_{i+1}^*)]$ over $\{x_i \mid \exists u_{i-1}\in U_{i-1}(x_{i-1}^*) : x_i = f_{i-1}(x_{i-1}^*,u_{i-1})\}\cap\{x_i \mid \exists(x_i,u_i)\in V_i : x_{i+1}^* = f_i(x_i,u_i)\}$.*
Proof. This is a corollary to the above Proposition 3.3.5. □

The following results relate stationarity and optimality.

Proposition 3.3.6 *Consider an OCP with fixed initial and end points. Assume that r_i, f_i, g_i and h_i are continuously differentiable. Assume that x^* maximizes $[\sum_{i=0}^{N-1} ub_i^{i+1}(x_i,x_{i+1})]$ over all feasible trajectories $\{x_i\}$. Assume that for $i=0,\dots,N-1$ the problem defining $ub_i^{i+1}(x_i^*,x_{i+1}^*)$ is tame and LMR regular, and that the u_i^* defining $ub_i^{i+1}(x_i^*,x_{i+1}^*)$ is unique and that the corresponding KKT multipliers $(p_{i+1}^*,\lambda_i^*,\mu_i^*)$ are unique. Then (x^*,u^*) satisfies the weak maximum principle (Proposition 1.4.6) with multipliers (p^*,λ^*,μ^*).*

Proof. The assumptions about uniqueness imply that all ub_i^{i+1} are continuously differentiable at (x_i^*,x_{i+1}^*). Therefore as x_i^* maximizes $[ub_{i-1}^i(x_{i-1}^*,x_i)+ub_i^{i+1}(x_i,x_{i+1}^*)]$ we have

$$\nabla_{x_i}(ub_{i-1}^i(x_{i-1}^*,x_i^*)+ub_i^{i+1}(x_i^*,x_{i+1}^*))=0$$

or, using Proposition 2.7.4 and rearranging, there hold at (x_i^*,u_i^*),

$$p_i^* = \nabla_{x_i}(r_i(x_i^*,u_i^*)+p_{i+1}^* f_i(x_i^*,u_i^*)-\lambda_i^* g_i(x_i^*,u_i^*)-\mu_i^* h_i(x_i^*,u_i^*))$$

Here $(p_{i+1}^*,\lambda_i^*,\mu_i^*)$ are determined in relation to the KKT conditions of u_i^* defining ub_i^{i+1}:

$$\nabla_{u_i}(r_i(x_i^*,u_i^*)+p_{i+1}^* f_i(x_i^*,u_i^*)-\lambda_i^* g_i(x_i^*,u_i^*)-\mu_i^* h_i(x_i^*,u_i^*))=0$$

with $\lambda_i^*\geq 0$, $\lambda_i^* g_i = 0$. This, together with the feasibility, is the weak maximum principle. □

The following result concerns stagewise conditions, and is in this sense more convenient.

Proposition 3.3.7 *Consider an OCP with fixed initial and end points. Assume that r_i, f_i, g_i and h_i are continuously differentiable. Assume that for all i x_i^* maximizes $[ub_{i-1}^i(x_{i-1}^*,x_i)+ ub_i^{i+1}(x_i,x_{i+1}^*)]$ over $\{x_i|(x_{i-1}^*,x_i)\in W_{i-1}^i$ and $(x_i,x_{i+1}^*)\in W_i^{i+1}$, and that for $i=0,\dots,N-1$ the u_i^* defining $ub_i^{i+1}(x_i^*,x_{i+1}^*)$ is unique and that the corresponding KKT multipliers $(p_{i+1}^*,\lambda_i^*,\mu_i^*)$ are unique. Then (x^*,u^*) satisfies the weak maximum principle (Proposition 1.4.6) with multipliers (p^*,λ^*,μ^*). If in addition r_i are concave, f_i and h_i linear and g_i convex then (x^*,u^*) solves OCP, and (x^*,u^*) is optimal in the OCP and satisfies the classical maximum principle (Proposition 1.3.5).*

Proof. The first part is proved as in the previous Proposition 3.3.6. For the sufficiency for optimality, cf. Proposition 1.3.5. □

As the following example shows the assumptions of continuous differentiability of ub_i^{i+1} is essential.

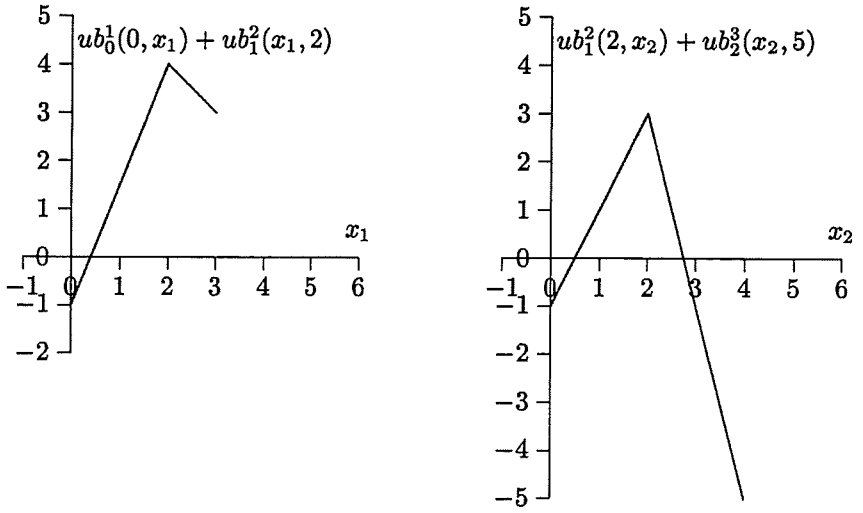


Figure 3.1: Two nonsmooth *ub*s

Example 3.3.1 *Consider the following problem:* $N = 2$, $n = 1$, $u_0 \in R$, $u_1 \in R^2$, $u_2 \in R$, $r_0 = 2u_0$, $r_1 = -3u_1^1 - 3u_1^2$, $r_2 = u_2$, $f_0 = x_0 + u_0$, $f_1 = x_1 + u_1^1 - u_1^2$, $f_2 = x_2 + u_2$, $0 \leq u_0 \leq 3$, $0 \leq u_1^1 \leq 3$, $0 \leq u_1^2 \leq 3$, $1 \leq u_2 \leq 5$, $x_0 = \underline{x}_0 = 0$, $x_3 = \underline{x}_3 = 5$.

We find, cf. Figure 3.1,

$$ub_0^1(0,x_1) = 2x_1, \text{ defined for } 0 \leq x_1 \leq 3$$

$$ub_1^2(x_1,x_2) = -3 \mid x_1 - x_2 \mid, \text{ defined for } -3 \leq x_2 - x_1 \leq 3$$

$$ub_2^3(x_2,5) = 5 - x_2, \text{ defined for } 0 \leq x_2 \leq 4$$

Take the feasible strategy $u^\diamond = (2,(0,0),3)'$ *and the resulting feasible trajectory* $x^\diamond = (0,2,2,5)'$. *We may find* $ub_0^1(0,x_1)+ub_1^2(x_1,2) = 2x_1-3 \mid x_1-2 \mid$ *(defined for* $0 \leq x_1 \leq 3$*) which has the unique maximizer* $x_1 = 2$*; and* $ub_1^2(2,x_2) + ub_2^3(x_2,5) = -3 \mid 2 - x_2 \mid +5 - x_2$ *(defined for* $0 \leq x_2 \leq 4$*) which has the unique maximizer* $x_2 = 2$.

Thus we see that for $i = 1,2$ $x_i^\diamond$ *maximizes* $[ub_{i-1}^i(x_{i-1}^\diamond, x_i) + ub_i^{i+1}(x_i, x_{i+1}^\diamond)]$. *Yet the unique optimal solution to the OCP is not* $(x^\diamond, u^\diamond)$ *but* $(x,u) = ((0,3,3,5)',(3,(0,0),2)')$.

It is easily verified that $u = u^\diamond$ *is the unique value in the definition of* $ub_i^{i+1}(x_i^\diamond, x_{i+1}^\diamond)$. *Therefore all assumptions of the above Proposition 3.3.7 are fulfilled, except that the KKT multipliers be unique.*

The KKT conditions in the definition of $ub_1^2(2,2)$ *are:* $\lambda_1^1 \geq 0$, $\lambda_1^2 \geq 0$ *and*

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -3 \\ -3 \end{pmatrix} - \lambda_1^1 \begin{pmatrix} -1 \\ 0 \end{pmatrix} - \lambda_1^2 \begin{pmatrix} 0 \\ -1 \end{pmatrix} + p_2 \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

It is seen that this does not yield a unique (λ_1, p_2), *and the assumptions of Proposition 3.3.7 are violated.* □

3.4 The Classical Maximum Principle and Generalizations

In the classical maximum principle the maximization part ascertains that the optimal control u_i^* maximizes the Hamiltonian subject to $u_i \in U_i(x_i^*)$. Thus, the state is fixed at x_i^*. The optimality conditions with respect to x_i is formulated as the adjoint equation which essentially is a stationarity condition with respect to x_i with u_i kept at u_i^*.

In this section we consider variants of this theme. We keep the classical perspective of maximizing with respect to u_i and having stationarity with respect to x_i. We shall generalize the results in two directions. First, by admitting a larger class of auxiliary functions $\pi_i : R^n \to R$ in the definition of the Hamiltonian:

$$H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i)) \tag{3.6}$$

In the classical maximum principle, a linear function was used, defined by the n-dimensional (row) vector p_{i+1}:

$$H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) \tag{3.7}$$

Second, we generalize by not requiring continuous differentiability.

The analysis of the smaller upper boundaries leads us to the understanding of the maximum part of the classical and the generalized maximum principles:

Proposition 3.4.1 *$(r_i(x_i, u_i^o), f_i(x_i, u_i^o)')'$ is situated at $ub_i^{i+1}(x_i, .)$ if and only if there is a $\pi_{i+1} : R^n \to R$ such that u_i^o maximizes $H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i))$ over $u_i \in U_i(x_i)$.*

Proof. Assume that u_i^o maximizes $H_i(x_i, u_i, \pi_{i+1})$. If $(r_i(x_i, u_i^o), f_i(x_i, u_i^o)')'$ is not situated at $ub_i^{i+1}(x_i, .)$, then there is another $u_i^\diamond \in U_i(x_i)$ such that $r_i(x_i, u_i^\diamond) > r_i(x_i, u_i^o)$ and $f_i(x_i, u_i^\diamond) = f_i(x_i, u_i^o)$. This means that $H_i(x_i, u_i^\diamond, \pi_{i+1}) = r_i(x_i, u_i^\diamond) + \pi_{i+1}(f_i(x_i, u_i^\diamond)) > r_i(x_i, u_i^o) + \pi_{i+1}(f_i(x_i, u_i^o)) = H_i(x_i, u_i^o, \pi_{i+1})$ contradicting that u_i^o was maximizing.

Now assume that $(r_i(x_i, u_i^o), f_i(x_i, u_i^o)')'$ is situated at $ub_i^{i+1}(x_i, .)$. Choose the function $\pi_{i+1} : R^n \to R$ such that $\pi_{i+1}(f_i(x_i, u_i^o)) = -ub_i^{i+1}(x_i, f_i(x_i, u_i^o))$ and $\pi_{i+1}(x_{i+1}) \geq -ub_i^{i+1}(x_i, x_{i+1})$ for all x_{i+1} where $ub_i^{i+1}(x_i, x_{i+1})$ is defined, and arbitrarily elsewhere. If u_i^o does not maximize the Hamiltonian, there is a $u_i^\diamond \in U_i(x_i)$ such that $H_i(x_i, u_i^\diamond, \pi_{i+1}) = r_i(x_i, u_i^\diamond) - ub_i^{i+1}(x_i, f_i(x_i, u_i^\diamond)) > H_i(x_i, u_i^o, \pi_{i+1}) = r_i(x_i, u_i^o) - UB_{i+1}(x_i, f_i(x_i, u_i^o)) = 0$. This implies $r_i(x_i, u_i^\diamond) > ub_i^{i+1}(x_i, f_i(x_i, u_i^\diamond))$, contradicting the definition of $ub_i^{i+1}(x_i, .)$. Therefore the postulated π_{i+1} exists. □

The existence of a function π_{i+1} satisfying Proposition 3.4.1 was easily demonstrated. In the proof of that Proposition the relationship to $ub_i^{i+1}(x_i, .)$ was used. See Figure 3.2.

If x_i had been assumed optimal, we could have used the function UB_{i+1}, cf. Proposition 3.2.3, or we could have used a backwards perspective, using RUB_{i+1}, cf. Proposition 3.2.4. We have already used these observations in Section 1.3.

From a practical perspective these choices are not so fortunate. First of all, there is no point in applying the classical or generalized maximum principle, if we first have to calculate upper boundaries; this is, at least for the greater upper boundaries, essentially the same as solving the problem. Second, the upper boundaries can be expected to be very complicated.

The interest therefore lies in finding assumptions that guarantee that we can use a more convenient approximation function π_{i+1}. We shall now characterize such function with respect to the upper boundaries.

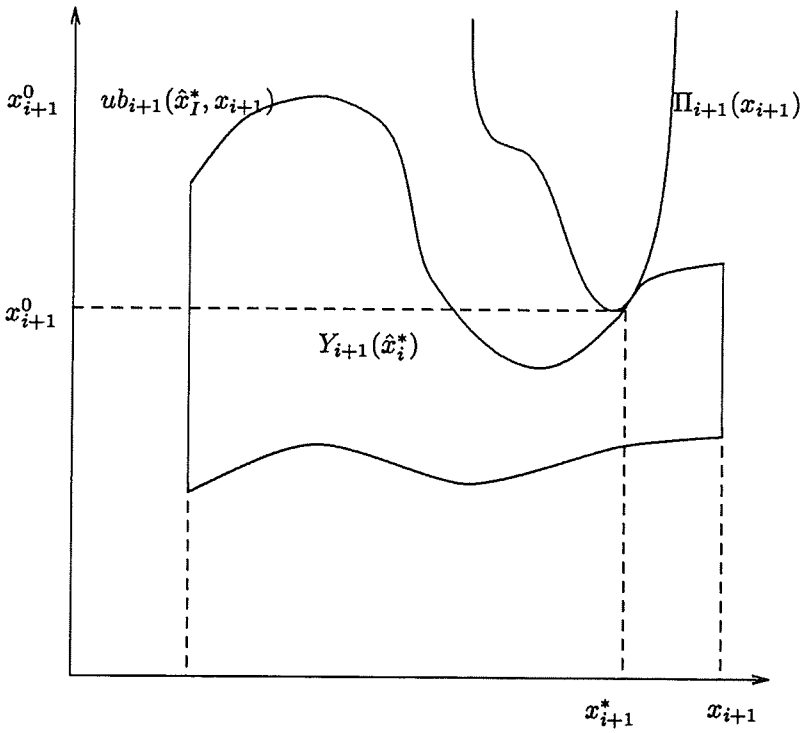


Figure 3.2: The Generalized Maximum Principle: Nonlinear support.

Proposition 3.4.2 *Assume that* $\hat{x}_{i+1} = (r_i(x_i, u_i^o), f_i(x_i, u_i^o)')'$ *belongs to* $\{ub_i^{i+1}(x_i, .) + \hat{x}_i^o\}$. *Denote* $f_i(x_i, u_i^o) = x_{i+1}^o$. *Then* u_i^o *maximizes* $H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i))$ *over* $u_i \in U_i(x_i)$ *if and only if*

$$-\pi_{i+1}(x_{i+1}) + \pi_{i+1}(x_{i+1}^o) \geq ub_i^{i+1}(x_i, x_{i+1}) - ub_i^{i+1}(x_i, x_{i+1}^o)$$

for all x_{i+1} *for which* $ub_i^{i+1}(x_i, .)$ *is defined.*

Proof. Suppose u_i^o is maximizing. Assume that there is a $x_{i+1}^\diamond$ for which $ub_i^{i+1}(x_i, x_{i+1}^\diamond)$ is defined such that

$$-\pi_{i+1}(x_{i+1}^\diamond) + \pi_{i+1}(x_{i+1}^o) < ub_i^{i+1}(x_i, x_{i+1}^\diamond) - ub_i^{i+1}(x_i, x_{i+1}^o)$$

By definition there is a $u_i^\diamond$ such that $x_{i+1}^\diamond = f_i(x_i, u_i^\diamond)$ and $ub_i^{i+1}(x_i, x_{i+1}^\diamond) = r_i(x_i, u_i^\diamond)$. The above relations then imply

$$-\pi_{i+1}(x_{i+1}^\diamond) + \pi_{i+1}(x_{i+1}^o) < r_i(x_i, u_i^\diamond) - r_i(x_i, u_i^o)$$

which can be rearranged to

$$\pi_{i+1}(x_{i+1}^o) + r_i(x_i, u_i^o) < \pi_{i+1}(x_{i+1}^\diamond) + r_i(x_i, u_i^\diamond)$$

However, this contradicts the assumption. Therefore

$$-\pi_{i+1}(x_{i+1}) + \pi_{i+1}(x_{i+1}^o) \geq ub_i^{i+1}(x_i, x_{i+1}) - ub_i^{i+1}(x_i, x_{i+1}^o)$$

Now suppose that the last relation holds. Assume u_i^o was not maximizing, i.e. there is an $u_i^\diamond$ such that $x_{i+1}^\diamond = f_i(x_i, u_i^\diamond)$ and

$$r_i(x_i, u_i^o) + \pi_{i+1}(f_i(x_i, u_i^o)) < r_i(x_i, u_i^\diamond) + \pi_{i+1}(f_i(x_i, u_i^\diamond))$$

As $\hat{x}_{i+1}$ was assumed to belong to $\{ub_i^{i+1}(x_i, .) + \hat{x}_i^o\}$, $ub_i^{i+1}(x_i, f_i(x_i, u_i^o)) = ub_i^{i+1}(x_i, x_{i+1}^o) = r_i(x_i, u_i^o)$ and therefore to belong to

$$ub_i^{i+1}(x_i, x_{i+1}^o) + \pi_{i+1}(x_{i+1}^o) < r_i(x_i, u_i^o) + \pi_{i+1}(x_{i+1}^\diamond)$$

As by definition $r_i(x_i, u_i^\diamond) \leq ub_i^{i+1}(x_i, x_{i+1})$ for all x_{i+1} with $x_{i+1} = f_i(x_i, u_i^\diamond)$ the last relation implies

$$-\pi_{i+1}(x_{i+1}^\diamond) + \pi_{i+1}(x_{i+1}^o) < ub_i^{i+1}(x_i, x_{i+1}^\diamond) - ub_i^{i+1}(x_i, x_{i+1}^o)$$

contradicting the assumption. Therefore u_i^o is maximizing. □

In the classical maximum principle a linear function was chosen: $\pi_i(x_i) = p_i x_i$. See (3.7) and Figure 3.3. Conditions for application of linear π as in the classical maximum principle were given in Proposition 1.4.3, which also gives sufficient optimality conditions.

Another classical result is the following, where the condition $\nabla_u H_i(x_i^*, u_i^*, p_{i+1}^*)\delta u_i \leq 0$ may be referred to as a "weak" maximum principle, cf. Proposition 1.3.4.

Proposition 3.4.3 *Consider the OCP (1.32) - (1.36) but without state dependence in the local constraints and with a free end point. Assume that* f_i *and* r_i *are continuously differentiable. Assume that* (x^*, u^*) *is optimal. Then there exist* $p_{i+1}^* \in R^n$ *such that the adjoint equations hold and such that*

$$\nabla_u H_i(x_i^*, u_i^*, p_{i+1}^*)\delta u_i \leq 0$$

where $\delta u_i \in R^m$ *and* $u_i^* + \epsilon \delta u_i \in U_i$ *for sufficiently small positive* ϵ. *If in addition* U_i *is convex and* H_i *is concave with respect to* u_i *then* u_i^* *maximizes* $H_i(x_i^*, u_i^*, p_{i+1}^*)$ *over* U_i.

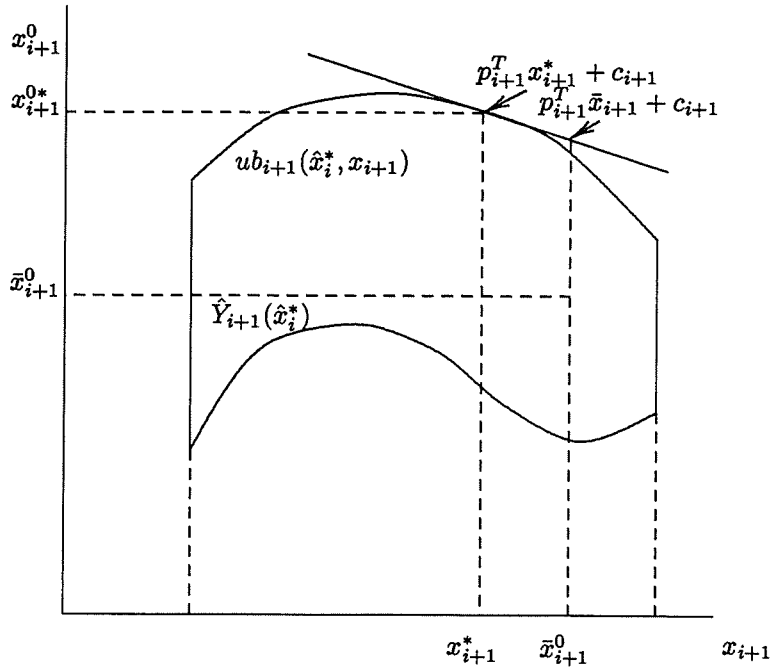


Figure 3.3: The Classical Maximum Principle: Linear support

Proof. As shown in relation to e.g. (1.57) - (1.62) and (6.8) - (6.9) the states may be eliminated and the criterion function expressed exclusively in terms of u as $r(u)$ where $r : R^{Nm} \to R$. It is seen that r is continuously differentiable. The adjoint equation will hold by appropriate definition of p_{i+1}, cf. Proposition 6.1.1. A necessary optimality condition is $\nabla r(u^*)\delta u \leq 0$, where $u^* + \epsilon\delta u \in U_0 \times \ldots \times U_{N-1}$ and $\epsilon > 0$ is sufficiently small. This condition implies $\nabla_u H_i(x^*_i, u^*_i, p^*_{i+1})\delta u_i \leq 0$, cf. Proposition 6.1.2. If H_i is concave with respect to u_i and U_i is convex then this in turn implies that u^*_i maximizes $H_i(x^*_i, u^*_i, p^*_{i+1})$ over U_i. □

The assumptions made here are fulfilled in some problems of real origin, but often they are too strong. Generalizations must therefore be found.

As seen from the above Propositions, π_{i+1} can be chosen of the same form as $-UB_{i+1}$, RUB_{i+1} or $ub^{i+1}_i(x^*_i, .)$. That is, it is always possible to choose an appropriate π_{i+1}, an insight which is due to Everett (1963). From Chapter 2 we therefore directly get conditions that permit selection of e.g. Lipschitz continuous or smooth π_{i+1} as far as the *maximization* part of the maximum principle is concerned. While this solves the difficulties with respect to the maximization, it may introduce other difficulties.

Therefore now turn to the adjoint equations part. In general, neither a linear nor a smooth π_{i+1} will work in the maximum part (even if the functions in the definition of the problem are smooth) and therefore we have to use a non-smooth π_{i+1}, if the maximum part shall hold. We then define the generalized Hamiltonian as

$$H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i)) \tag{3.8}$$

where $\pi_{i+1} : R^n \to R$.

For the remaining part of this section we assume V_i given by constraints $g_i(x_i,u_i) \leq 0$, $h_i(x_i,u_i) = 0$ and V_N given by $g_N(x_n) \leq 0$, $h_N(x_N) = 0$. However, also for the more general formulation $x_i \in V_i$ is it possible to derive results quite similar to those presented below using non-smooth calculus.

The following results then guarantee that the *generalized maximum principle* holds with non-smooth Hamiltonian. Recall the discussion of tameness and constraint qualifications in Section 2.5, and recall that ∂ denotes the generalized gradient.

Proposition 3.4.4 *Assume that r_i, f_i, g_i and h_i are Lipschitz continuous for $0 \leq i \leq N-1$ and that r_N, g_N and h_N are Lipschitz continuous. Let (x^*,u^*) be optimal. Assume that for all i, $i = 1,\ldots,N$, the truncated problems ending with the state x_i^* are tame and satisfy a constraint qualification along every optimal strategy and trajectory leading to x_i^*. Then there exist Lipschitz continuous π_i^* and multipliers λ_i^* and μ_i^*, $i = 0,\ldots,N$, such that*

- *u_i^* maximizes $H_i(x_i^*,u_i,\pi_{i+1}^*)$ subject to $g_i(x_i^*,u_i) \leq 0$, $h_i(x_i^*,u_i) = 0$, $i = 0,\ldots,N-1$*
- *$\lambda_i^* \geq 0$, $\lambda_i^* g_i(x_i^*,u_i^*) = 0$, $i = 0,\ldots,N-1$*
- *$0 \in \partial_x(H_0(x_0^*,u_0^*,\pi_1^*) - \lambda_0^* g_0(x_0^*,u_0^*) - \mu_0^* h_0(x_0^*,u_0^*))$*
- *$0 \in \partial_x(H_i(x_i^*,u_i^*,\pi_{i+1}^*) - \pi_i^*(x_i^*) - \lambda_i^* g_i(x_i^*,u_i^*) - \mu_i^* h_i(x_i^*,u_i^*))$, $i = 1,\ldots,N-1$*
- *$0 \in \partial(r_N(x_N^*) - \lambda_N^* g_N(x_N^*) - \mu_N^* h_N(x_N^*) - \pi_N^*(x_N^*))$*
- *$\lambda_N^* \geq 0$, $\lambda_N^* g_N(x_N^*) = 0$*

Proof. Define $\pi_i^* = -UB_i$ for $i = 0,\ldots,N$. From Proposition 3.2.3 we have that (x_i^*,u_i^*) maximize $[r_i(x_i,u_i) - \pi_i^*(x_i) + \pi_{i+1}^*(f_i(x_i,u_i))]$ subject to $(x_i,u_i) \in V_i$, $x_i \in Y_i$.

Therefore with $x_i = x_i^*$, u_i^* maximizes the Hamiltonian subject to $u_i \in V_i(x_i^*)$, i.e $g_i(x_i^*,u_i) \leq 0$, $h_i(x_i^*,u_i) = 0$, for $i = 0,\ldots,N-1$.

By the assumptions of all truncated problems being regular and tame along every optimal strategy and trajectory leading to x_i^* it follows that π_i is Lipschitz continuous on Y_i for $i = 0,\ldots,N$. It also follows that x_i^* is in the interior of Y_i and $f_i(x_i^*,u_i^*)$ is in the interior of Y_{i+1}, cf. Proposition 2.6.1. Therefore the constraint $x_i \in Y_i$ is not active at (x_i^*,u_i^*). Consequently, for $i = 0,\ldots,N-1$, (x_i^*,u_i^*) maximizes locally $[r_i(x_i,u_i) - \pi_i^*(x_i) + \pi_{i+1}^*(f_i(x_i,u_i))]$ subject only to $(x_i,u_i) \in V_i$, and x_N^* maximizes locally UB_N subject only to $V_N \cap Y_N$. For those local maxima, the remaining conditions hold by the LM rule (p. 74). □

Proposition 3.4.5 *Assume that r_i, f_i, g_i and h_i are Lipschitz continuous for $0 \leq i \leq N-1$ and that r_N, g_N and h_N are Lipschitz continuous. Let (x^*,u^*) be optimal. Assume that for all i, $i = 0,\ldots,N$, the truncated problems starting at x_i^* are tame and satisfy a constraint qualification along every optimal strategy and trajectory from x_i^*. Then there exist Lipschitz continuous π_i^* and multipliers λ_i^* and μ_i^*, $i = 0,\ldots,N$, such the conclusions in Proposition 3.4.4 hold.*

Proof. Define $\pi_i = RUB_i$ for $i = 0,\ldots,N$. From Proposition 3.2.4 we have that (x_i^*,u_i^*) maximize $[r_i(x_i,u_i) - \pi_i(x_i) + \pi_{i+1}(f_i(x_i,u_i))]$ subject to $(x_i,u_i) \in V_i$ and $f_i(x_i,u_i) \in RY_{i+1}$. With x_i fixed at x_i^* then the parts on maximization with respect to u_i hold.

By the assumptions of all partial problems being regular and tame along every optimal strategy and trajectory from x_i^* it follows that all π_i^* are Lipschitz continuous on RY_i. It also follows that x_i^* is in the interior of RY_i and that $f_i(x_i^*, u_i^*)$ is in the interior of RY_{i+1}, cf. Proposition 2.6.2. Therefore the constraint $f_i(x_i, u_i) \in RY_{i+1}$ is not active, and the remaining conditions hold by the LM rule (p. 74) as necessary optimality conditions for a local maximum. □

The cost of having the generalized maximum principle hold without differentiability and convexity assumptions is that the adjoint *equation* is weakened to an adjoint *relation* or *inclusion*.

Under some additional assumptions the result may be strengthened as follows:

Proposition 3.4.6 *Assume that the adjoint relations hold as in the above Propositions 3.4.4 and 3.4.5. Then under additional assumptions the adjoint relation may be strengthened, viz.:*

- *If π_i^* is continuously differentiable at x_i^* then*

$$\nabla\pi_i^*(x_i^*) \in \partial_x(H_i(x_i^*, u_i^*, \pi_{i+1}^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*))$$

- *If g_i and h_i are continuously differentiable at (x_i^*, u_i^*) then*

$$\nabla_x(\lambda_i^* g_i(x_i^*, u_i^*) + \mu_i^* h_i(x_i^*, u_i^*)) \in \partial_x(H_i(x_i^*, u_i^*, \pi_{i+1}^*) - \pi_i^*(x_i^*))$$

- *If r_i, g_i and h_i are continuously differentiable at (x_i^*, u_i^*) then*

$$\nabla_x(-r_i(x_i^*, u_i^*) + \lambda_i^* g_i(x_i^*, u_i^*) + \mu_i^* h_i(x_i^*, u_i^*))$$
$$\in \partial_x(\pi_{i+1}^*(f_i(x_i^*, u_i^*)) - \pi_i^*(x_i))$$

- *If f_i and π_{i+1}^* are continuously differentiable at (x_i^*, u_i^*) and $f_i(x_i^*, u_i^*)$, respectively, then*

$$\nabla_{x_i}\pi_{i+1}^*(f_i(x_i^*, u_i^*)) \in \partial_x(-r_i(x_i^*, u_i^*) + \pi_i^*(x_i^*) + \lambda_i^* g_i(x_i^*, u_i^*) + \mu_i^* h_i(x_i^*, u_i^*))$$

- *If r_i, f_i, g_i, h_i, π_i^* and π_{i+1}^* are continuously differentiable at (x_i^*, u_i^*), (x_i^*, u_i^*), (x_i^*, u_i^*), (x_i^*, u_i^*), x_i^* and $f_i(x_i^*, u_i^*)$, respectively, then*

$$\nabla\pi^*(x_i^*) = \nabla_x(H_i(x_i^*, u_i^*, \pi_{i+1}^*) - \lambda_i g_i(x_i^*, u_i^*) - \mu h_i(x_i^*, u_i^*))$$

Proof. This is an application of non-smooth calculus, see Clarke (1983), to Propositions 3.4.4 and 3.4.5. □

Conditions for application of a *quadratic* π were given as the *augmented Hamiltonian maximum principle*, Proposition 1.4.5. Conditions for other properties of π, e.g. continuous differentiability or linearity, are given in Chapter 2 and implicitly in Section 3.5.

All the above conditions are, under the assumptions taken, necessary optimality conditions. As long as only stationarity with respect to x is assumed then assumptions of convexity, and also of smoothness of π, must be made in order to get sufficient conditions for optimality:

Proposition 3.4.7 *Assume that r_i is concave, f_i is linear, g_i is convex and h_i is linear for all i. Assume that the necessary conditions in any of the above Propositions 3.4.4, 3.4.5 or 3.4.6 are fulfilled at a feasible solution (x^o, u^o), and that π_i^* is continuously differentiable at x_i^o for $i = 1, \ldots, N$. Then this solution is optimal.*

Proof. Select $p_i = \nabla\pi_i^*(x_i^o)$, $i = 1, \ldots, N$. Due to the assumption of concavity, linearity and convexity, the maximization and stationarity conditions with respect to u_i and x_i in Propositions 3.4.4 - 3.4.6 imply maximization of $[r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) - p_i x_i]$ with respect to (x_i, u_i). Therefore the assumptions of Proposition 3.5.2 will hold using these p_i, and by that proposition the solution is optimal. □

Singular Problems

A singular problem is in this context understood to be a problem where the condition that the optimal control at stage i, u_i^*, maximizing the Hamiltonian with linear π_{i+1} does not uniquely identify u_i^*. Observe that in the context of admitting nonlinear π_{i+1}, singularity is dependent not only on the problem but on the function π_{i+1} as well.

If f_i is linear in u_i, r_i is strictly concave and $U_i(x_i)$ convex for any x_i then the problem is non-singular with linear π_{i+1}.

Let us observe that u_i^* (the solution at stage i to the OCP) may very well be unique, even if the u_i maximizing the Hamiltonian is not. Thus for instance we may get nice smoothness results as in Sections 2.7 and 2.8 even with singular problems, provided that the optimal solution to the whole OCP problem is unique, not only the optimal solution to the local problem of maximizing the Hamiltonian. See Example 3.4.1 below. On the other hand, the OCP may have more than one solution, and yet be nonsingular.

Singular problems may exhibit a particular solution structure. Thus for instance with bang-bang control the optimal control may from one index i to the next one change from one extreme point in $U_i(x_i)$ to another, see e.g. the example on page 45.

We shall here give a brief discussion of singularity in relation to the difficulties with solution attempts, analytically or by numerical methods.

Let us distinguish between proper singularity and weak singularity in the following way. Let u_i^+ and u_i^o be any two different values that maximize the Hamiltonian $H_i(x_i^*, u_i, \pi_{i+1}^*)$ over $U_i(x_i^*)$. We say that the problem is weakly singular at (x_i^*, π_{i+1}^*) if $f_i(x_i^*, u_i^o) = f_i(x_i^*, u_i^+)$ for all u_i^+ and all u_i^o. We say that the problem is properly singular if $f_i(x_i^*, u_i^o) \neq f_i(x_i^*, u_i^+)$ for some u_i^+ and u_i^o.

In an algorithmic context, proper singularity may be a problem because with this, x_{i+1} is not uniquely defined from x_i^* and the maximizing u_i^*. Thus the problem is in the forwards direction, and linked to the dynamic equation. In this respect there is no difficulty with weak singularity. Here the problem is in the backwards direction where in this case p_i need not be uniquely defined from $\nabla_x H_i$ by the adjoint equation. - Observe that even if the problem is non-singular there may be difficulties in the backwards adjoint relations recursion if the Lagrange multipliers are not unique, cf. e.g. in relation to algorithms described at the end of Section 6.4 and Section 7.1.

There are essentially three ways to eliminate the difficulties caused by singularity: (*i*) to monitor the non-unique maximizing u_i carefully, (*ii*) to manipulate the problem of maximizing the Hamiltonian or (*iii*) to manipulate the formulation of the optimal control problem. Let us consider these possibilities.

The monitoring of u_i will normally be required to take place over a sequence of stages. The purpose is to be able to select those u_i that maximize (non-uniquely) the Hamiltonian and which at the same time permit the satisfaction of additional constraints, typically an end point condition $x_N = \underline{x}_N$. The way to do this will depend on the particular problem. See the discussion in Section 6.3 and see e.g. the treatment of linear problems in Section 9.3.

The second way is to manipulate the problem of maximization of the Hamiltonian in order

that this problem yields a unique solution. Thus, rather than using the Hamiltonian with linear π_{i+1} we may use a strictly concave π_{i+1}. This will be done in e.g. Sections 6.1 and 7.2. This may eliminate proper singularity but not weak singularity. Another way is to add to the Hamiltonian terms that are strictly concave in u_i, see e.g. Section 6.1. This may eliminate weak singularity as well.

The third way to eliminate singularity is to manipulate the problem. This was done in the isotonic regression problem treated in Section 1.7. There, the criterion function (1.127) is strictly concave at stage i with respect to u_i for any fixed x_i and hence non-singular even with a linear π_{i+1}. However, (1.125) is an equivalent criterion function, but the Hamiltonian relative to this function is singular with a linear π_{i+1}.

The linear optimal control problem is singular with a linear π_{i+1}, and is therefore not so easily handled by the usual maximum principle. We conclude these observations on singularity by showing that the linear optimal control problem may be reformulated to a nonsingular problem which is (almost) equivalent.

Consider the OCP with linear criterion function:

$$\max[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N] \tag{3.9}$$

and where the dynamics and local constraints are linear; cf. e.g. (1.82) - (1.86). We substitute the criterion function by one which is strictly concave in u:

$$\max[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i - \alpha(\sum_{j=1}^{m}(u_i^j - \underline{u}_i^j)^2) + R_N^x x_N] \tag{3.10}$$

Here, α is a positive scalar and $\underline{u} \in R^{Nm}$ is arbitrary. This criterion function is strictly concave at stage i with respect to u_i for any fixed x_i and hence non-singular with a linear π_{i+1}.

The following shows that the relations between the two problems are such that if only one optimal solution is desired then the latter formulation may be used, provided α is chosen sufficiently small:

Proposition 3.4.8 *Consider the linear OCP with criterion function (3.9) and fixed initial point, and assume that it has an optimal solution. Obtain the quadratic-linear OCP from the linear OCP by using criterion function (3.10) and the same dynamics and local constraints. Then there is an $\alpha > 0$ such that the unique optimal solution to the quadratic-linear OCP is also an optimal solution to the linear OCP.*

Proof. Consider the following linear programming problem:

$$\max[c'u]$$
$$Au \leq a$$

where $c \in R^n$, $u \in R^n$, $a \in R^m$ and A is a $m \times n$ matrix. We propose to solve this problem by solving the quadratic linear problem

$$\max[c'u - \alpha\|u - \underline{u}\|^2]$$
$$Au \leq a$$

where $\alpha > 0$ is a scalar and $\underline{u}$ is an arbitrary point in R^n. This problem has a unique optimal solution if it has a feasible solution i.e. if $\{u \mid Au \leq a\}$ is nonempty.

Lemma *Assume that the linear problem has an optimal solution. Then there is an $\alpha > 0$ such that the unique solution to the quadratic linear problem is also a solution to the linear problem.*

Proof. Let U^* be the set of optimal solution to the LP problem and let $u^* \in U^*$. The necessary and sufficient KKT conditions for optimality of the LP problem are for some $\lambda \in R^m$:

$$c' = \lambda^* A,\ \lambda^* \geq 0,\ \lambda^*(Au^* - a) = 0 \tag{3.11}$$

If $\lambda = 0$ then $c' = 0$ and the Lemma obviously holds. Therefore assume that $\lambda^j > 0$ for at least one index j.

Let $\overline{u}^*$ be the point in U^* which is closest to $\underline{u}$ i.e., $\overline{u}^*$ satisfies

$$\| \underline{u} - \overline{u}^* \|^2 = \min_y [\| y - \underline{u} \|^2, y \in U^*]$$

This problem can also be written

$$\max_y [- \| y - \underline{u} \|^2] \tag{3.12}$$

$$Ay \leq a \tag{3.13}$$

$$-c'y \leq -c'u^* \tag{3.14}$$

For this problem the KKT multipliers $\underline{\lambda} \in R^m$, $\underline{\lambda}^c \in R$ satisfy

$$-2(\overline{u}^* - \underline{u})' = \underline{\lambda} A - \underline{\lambda}^c c,\ \underline{\lambda} \geq 0,\ \underline{\lambda}^c \geq 0 \tag{3.15}$$

$$\underline{\lambda}(A\overline{u}^* - a) + \underline{\lambda}^c(-c'\overline{u}^* + c'u^*) = 0 \tag{3.16}$$

We show that $\underline{\lambda}^c$ is not unique and can be chosen positive. To see this observe that $\overline{u}^*$ is a solution to the LP problem and therefore in this we can have the same part of the constraints $(Au)^j \leq a^j$ active as in (3.12) - (3.14). In addition, (3.14) will be active. Now assume $\underline{\lambda}^c = 0$; then increase all positive $\underline{\lambda}^j$ by $\delta\lambda^j$ in (3.15) - (3.16) and select $\delta\lambda^c$ so that $\delta\lambda A - \delta\lambda^c c = 0$. This is possible in view of (3.11) and it will not violate the complementary slackness conditions in (3.15) - (3.16). Hence (3.15) - (3.16) still holds, but now with $\underline{\lambda}^c > 0$.

Therefore let $\underline{\lambda}^c > 0$. This, and the observation that (3.14) is binding (implying $-c'\overline{u}_i^* + c'u^* = 0$), permit that we in (3.15) - (3.16) divide by $\underline{\lambda}^c$ and rearrange to get

$$-(2/\underline{\lambda}^c)(\overline{u}^* - \underline{u})' + c' = (\underline{\lambda}/\underline{\lambda}^c)A,\ (\underline{\lambda}/\underline{\lambda}^c) \geq 0$$

$$(\underline{\lambda}/\underline{\lambda}^c)(A\overline{u}^* - a) = 0$$

These sufficient KKT conditions show that $\overline{u}^*$ is a solution to the quadratic linear problem with $\alpha = 1/\underline{\lambda}^c$. This ends the proof of the Lemma.

For the linear OCP it is sufficient to add terms quadratic in u and the proof of the Proposition follows by elimination of x (see e.g. (6.8) - (6.9)) and considering the resulting linear problem in the variables u as a special case of the linear problem considered in the Lemma. □

Example 3.4.1 *Consider the following problem*

$$\max[\sum_{i=0}^{4} iu_i]$$

$$x_{i+1} = x_i + u_i$$

$$0 \leq u_i \leq 2$$

$$x_0 = 0$$

$$x_5 = 5$$

One may verify that p_i^* is uniquely given as $p_i^* = -2$. We find that maximization of the Hamiltonian yields the following solution:

$$u_i^* = \begin{cases} 0 & for \quad i = 0,1 \\ ? & for \quad i = 2 \\ 2 & for \quad i = 3,4 \end{cases}$$

For $i = 2$ we see that any u_2 is optimal because $H_2(x_2^*, u_2, p_3^*) = -x_2^*$, i.e., H_2 is independent of u_2. Therefore the problem is properly singular. One may verify that the solution to the problem despite this is unique, and has $u_2^* = 1$. □

3.5 The Extended Maximum Principle

Now we turn to an optimality principle which we shall call the extended maximum principle. This distinguishes itself from the previous ones by the fact that the maximization at stage i is with respect to (x_i, u_i), not only with respect to u_i. The gain in return for this complication is that we get sufficient optimality conditions *without* assumptions of concavity, convexity and linearity. As before functions π_i are central to the analysis. The results of Propositions 3.5.1 - 3.5.3 were first obtained in Krotov (1967), cf. Krotov (1988).

The results are intimately related to duality, which is dealt with towards the end of the section.

Consider the optimal control problem (OCP):

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{3.17}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{3.18}$$

$$(x_i, u_i) \in V_i \tag{3.19}$$

$$x_N \in V_N \tag{3.20}$$

For this problem there are two obvious types of relaxation.

One is to relax the local constraints $(x_i, u_i) \in V_i$. This idea will not be followed here, since we go for the second type of relaxation viz., the stagewise decomposition. For this

We therefore introduce the functions $\pi_i : R^n \to R$, $i = 0, \ldots, N$ and relax the dynamic equations. We add to the criterion (3.17) the following expressions, that cancel whenever the dynamic equations (3.18) are fulfilled:

$$\sum_{i=1}^{N} \pi_i(f_{i-1}(x_{i-1}, u_{i-1})) - \pi_i(x_i) \tag{3.21}$$

A notational variant adds also $(\pi_0(x_0) - \pi_0(x_0))$.

We may therefore define the *relaxed and price coordinated problem* as follows:

$$\max_{(x,u)}[\sum_{i=0}^{N-1}(r_i(x_i,u_i)-\pi_i(x_i)+\pi_{i+1}(f_i(x_i,u_i)))+r_N(x_N) \tag{3.22}$$
$$+\pi_0(x_0)-\pi_N(x_N)]$$
$$(x_i,u_i)\in V_i \tag{3.23}$$
$$x_N\in V_N \tag{3.24}$$

The criterion in (3.22) may alternatively be written

$$\sum_{i=0}^{N-1} r_i(x_i,u_i)+r_N(x_N) \tag{3.25}$$
$$+\sum_{i=1}^{N}(\pi_i(f_{i-1}(x_{i-1},u_{i-1}))-\pi_i(x_i))$$

Observe, that now this problems decomposes into $(N+1)$ independent problems, one for each i:

$$\max_{(x_0,u_0)}[r_0(x_0,u_0)+\pi_1(f_0(x_0,u_0))] \tag{3.26}$$
$$(x_0,u_0)\in V_0 \tag{3.27}$$
$$\max_{(x_i,u_i)}[r_i(x_i,u_i)-\pi_i(x_i)+\pi_{i+1}(f_i(x_i,u_i))] \tag{3.28}$$
$$(x_i,u_i)\in V_i,\ i=1,\ldots,N-1 \tag{3.29}$$
$$\max_{x_N}[r_N(x_N)-\pi_N(x_N)] \tag{3.30}$$
$$x_N\in V_N \tag{3.31}$$

Proposition 3.5.1 *The optimal criterion value in (3.22) - (3.24) is an upper bound on the optimal criterion value in the OCP (3.17) - (3.20).*

Proof. The set of feasible solutions to (3.17) - (3.20) are given by (3.18) - (3.20). This is a subset of the feasible solutions to (3.22) - (3.24) which are given by (3.23) - (3.24). For all (x,u) satisfying (3.18) the terms in (3.22) or (3.25) involving π cancel and therefore for these (x,u) (3.17) and (3.22) take the same value. Therefore the optimal criterion value of (3.17) - (3.20) cannot be bigger than the optimal criterion value in (3.22) - (3.24). □

Proposition 3.5.2 *Let there be given the OCP and some π. Suppose $(x^\diamond,u^\diamond)$ is an optimal solution to (3.22) - (3.24). If the dynamic equations hold for these $(x^\diamond,u^\diamond)$ for all i then $(x^\diamond,u^\diamond)$ is an optimal solution to the OCP (3.17) - (3.20).*

Proof. $(x^\diamond,u^\diamond)$ is a feasible solution in (3.17) - (3.20). By the previous Proposition 3.5.1 $(x^\diamond,u^\diamond)$ gives an upper bound on the optimal value in (3.17) - (3.20). Since the upper bound is attained by a feasible solution this solution is optimal. □

We have given above some conditions which guarantee that a solution found by relaxation and price coordination is indeed optimal in the original problem. If the conditions are not fulfilled, we get in any case an upper bound on the optimal criterion value. As seen the main point is the stagewise decomposition of the optimization, i.e. solution of (3.26) - (3.31). The subproblems are coordinated by price functions π_i. Observe that the implicit restrictions we had to take into account in relation to dynamic programming (see for instance $f(x_i, u_i) \in RY_{i+1}$ in Proposition 3.2.1), are not present here. This makes the decomposition more convenient than that of dynamic programming.

We formulated above the extended maximum principle as sufficient conditions for optimality, Proposition 3.5.2. We proceed to show under which circumstances this maximum principle is necessary, i.e., when the functions π_i exist, such that the problem may be solved this way.

Proposition 3.5.3 *Assume that r_i is bounded on V_i for all i and that there exists a solution to the OCP (3.17) - (3.20). Then there exist π such that the solution to the relaxed problem (3.22) - (3.24) satisfies the dynamic equation (and therefore also by Proposition 3.5.2 solves (3.17) - (3.20)).*

Proof. Let (x^*, u^*) be an optimal solution to OCP. Choose $\pi_N(x_N) = RUB_N(x_N) = r_N(x_N)$, $\forall x \in V_N$. Choose $\pi_N(x_N) = k_N$ for all other x_N, for which π_N need be defined, i.e. for $x_N \in (Z_N - V_N)$. Here k_N is a constant, such that for $i = N-1$ there holds for all $(x_i, u_i) \in V_i$

$$r_i(x_i, u_i) - RUB_i(x_i) + \pi_{i+1}(f_i(x_i, u_i)) \leq$$

$$r_i(x_i^*, u_i^*) - RUB_i(x_i^*) + \pi_{i+1}(f_i(x_i^*, u_i^*))$$

Such constant exists since r_N and r_{N-1} were assumed bounded.

Now define recursively backwards, $i = N-1, \ldots, 0$, $\pi_i(x_i) = RUB_i(x_i)$ on RX_i and $\pi_i(x_i) = k_i$ on $Z_i - RX_i$. Here k_i is a constant, such that the above relationship holds for all $(x_i, u_i) \in X_i \cap RX_i$. Such constants exist, since r_i were assumed bounded on V_i.

With π defined this way we see that there holds for all i that (x_i^*, u_i^*) maximizes $[r_i(x_i, u_i) - \pi_i(x_i) + \pi_{i+1}(f_i(x_i, u_i))]$ over V_i. Since (x^*, u^*) was assumed optimal, the dynamic equation holds for (x^*, u^*).

Remark: the idea in the proof is that π_i can be selected equal to the backwards dynamic programming greater upper boundaries where these are defined, and sufficiently small elsewhere. We might as well have worked with the forwards greater upper boundaries. □

From a theoretical point of view this result is comforting, but not from a practical point of view. If π_i must be chosen as essentially equal to one of the greater upper boundaries, there is no point in applying relaxation, since the problem is already nearly solved, when the upper boundaries are constructed. Fortunately, π_i may be chosen simpler. As an example, we know that certain problems can be solved by Lagrangian relaxation (see also Proposition 3.5.6 below). This means that these problems can be solved using linear π_i, although the greater upper boundaries are not linear.

To give conditions on π_i, we shall define some other functions $F_i : R^n \to R$, $i = 0, \ldots, N$. Specifically we define $F_0(x_0) = 0$, for all x_0, and generally

$$F_{i+1}(x_{i+1}) = \max_{(x_i, u_i)} [r_i(x_i, u_i) - \pi_i(x_i)] \tag{3.32}$$

subject to (3.18) - (3.19). F_{i+1} is defined on Z_i^{i+1} i.e., for all x_{i+1}, for which there exists a $(x_i, u_i) \in V_i$ such that $x_{i+1} = f_i(x_i, u_i)$.

Further we define functions $RF_i : R^n \to R$, $i = 0, \ldots, N-1$, as

$$RF_i(x_i) = \max_{(x_i,u_i)} [r_i(x_i,u_i) + \pi_{i+1}(f_i(x_i,u_i))] \tag{3.33}$$

subject to (3.19). RF_i is defined on the set X_i of locally admissible states, i.e., for all x_i, for which there is a u_i, such that $(x_i,u_i) \in V_i$. Specifically we define $RF_N(x_N) = 0$.

Consider the following relations:

$$\begin{aligned} UB_i(x_i) - UB_i(x_i^*) &\leq F_i(x_i) - F_i(x_i^*) && (3.34)\\ &\quad \forall x_i \in Y_i \cap X_i \\ F_i(x_i) - F_i(x_i^*) &\leq -\pi_i(x_i) + \pi_i(x_i^*) && (3.35)\\ &\quad \forall x_i \in Z_{i-1}^i \cap X_i \\ -\pi_i(x_i) + \pi_i(x_i^*) &\leq -RF_i(x_i) + RF_i(x_i^*) && (3.36)\\ &\quad \forall x_i \in X_i \\ -RF_i(x_i) + RF_i(x^*) &\leq -RUB_i(x_i) + RUB_i(x_i^*) && (3.37)\\ &\quad \forall x_i \in X_i \cap RY_i \end{aligned}$$

They link π_i to the dynamic programming upper boundaries UB_i and RUB_i, and they link π_i to π_{i-1} and π_{i+1}. The inequalities do not hold for the same arguments x_i, since they are not defined on the same sets. But we see that all the relations are defined at x_i^* , the optimal state at stage i, if it exists, and that for this state there holds equality in all the relations.

Proposition 3.5.4 *Assume that the problem has a solution (x^*,u^*). Then (x^*,u^*) is a solution to (3.26) - (3.31) satisfying the dynamic equation if and only if the relations (3.34) - (3.37) hold for all i.*

Proof. Assume that (x^*,u^*) is a solution to (3.26) - (3.31) satisfying the dynamic equation. Clearly (3.34) and (3.35) hold for $i = 0$. By the definition (3.32), (3.34) holds for $i = 1$. By the definition (3.32) we see that (3.35) holds for $i = 1$, since otherwise (x_1^*,u_1^*), which by assumption satisfies $x_2^* = f_1(x_1^*,u_1^*)$, would not be maximizing in (3.26) - (3.31). Since (3.35) holds for $i = 1$, we see that (3.34) holds for $i = 2$, due to the relations (3.23) and (3.26). Continuing this way we see that (3.34) and (3.35) hold for all i. Similarly, (3.36) and (3.37) hold for $i = N$. Moreover, (3.37) holds for $i = N-1$ by definition (3.33). (3.36) holds for $i = N-1$, because otherwise (x_{N-2}^*,u_{N-2}^*), which satisfies $x_{N-1}^* = f_{N-2}(x_{N-2}^*,u_{N-2}^*)$, would not be maximizing in (3.26) - (3.31). Since (3.36) holds for $i = N-1$ we see that (3.30) holds for $i = N-2$, due to the definition of RUB_i and (3.33). Continuing this way, we see that also (3.36) and (3.37) hold for all i.

Now assume that (3.34) and (3.35) hold for all i. Then by the definition of UB_i and (3.32) for any i there will be a maximizing (x_i^*,u_i^*) in (3.26) - (3.31) such that (3.18) and (3.20) hold. □

Actually, we see from the proof of Proposition 3.5.4 that we can strengthen the conditions a little bit:

Proposition 3.5.5 *Assume that the problem (3.17) - (3.20) has a solution. Then (3.34) - (3.35) hold if and only if (3.36) - (3.37) hold.*

Proof. If either (3.34) - (3.35) or (3.36) - (3.37) hold then there will be a (x^*,u^*) maximizing (3.26) - (3.31) and satisfying (3.18) and (3.20), according to the definition of UB_i and (3.32) or

the definition of RUB_i and (3.27), respectively. But according to Proposition 3.5.4 this implies that (3.34), (3.35), (3.36) and (3.37) hold. □

We see that there are intimate relations between the functions π_i and the upper boundaries, as well as between π_i and π_{i+1}. This may for instance be exploited in relation to Proposition 3.4.6 by showing that if RUB_i and UB_i are continuously differentiable then also π_i may be chosen so.

As shown in the previous section there are similarly strong relationships between the upper boundaries, as well as between upper boundaries and the price functions π_i used in the various optimality and maximum principles. Therefore the characterization of the previous two propositions have equally strong implications for the other optimality and maximum principles.

Lagrangian Relaxation

We can now give a restatement in our terminology of the well-known result that a convex mathematical programming problem satisfying constraint qualifications, can be solved by Lagrangian relaxation. That is, we can use linear π_i, i.e., $\pi_i(x_i) = p_i x_i$, where $p_i \in R^n$ is a row vector. See the formulation in (1.66) - (1.70) on page 28, and cf. also Proposition 1.4.2.

In the Proposition we refer to a constraint qualification. This may take various forms, depending on the structure of the restrictions (3.18) - (3.20), as discussed in Section 2.5.

Proposition 3.5.6 *Assume that in the OCP (3.17) - (3.20) r_i is concave, f_i is affine, V_i is convex (g_i is convex and h_i is affine if V_i is given as $\{(x_i, u_i) \mid g_i(x_i, u_i) \leq 0,\ h_i(x_i, u_i) = 0\}$) and that a constraint qualification holds. Then, if the problem has a solution, there exist linear π such that a solution to (3.26) - (3.31) satisfies the dynamic equation (and therefore also by Proposition 3.5.2 solves (3.17) - (3.20)).*

Proof. The result can be found in standard textbooks on mathematical programming, see e.g. Bazaraa and Shetty (1979), Theorems 3.2.5. and 6.2.4. □

The sufficiency counterpart to this result on necessary conditions is again Proposition 3.5.2 and for the particular case of linear π_i see Vidal (1987) and Proposition 1.4.1.

If the problem does not fulfill the assumptions of Proposition 3.5.6 one may have to use supports that are non-linear, if we shall solve the problem by decomposition. In specific cases one may be able to describe the supports qualitatively (but maybe leaving some parameters unspecified) by a close analysis of the problem. This, indeed, is the case under the assumptions of Proposition 3.5.6, where the general form of the supports can be linear, but where the question of what value some parameters (in this case, the slopes p_i) must take is left open. The previous section of this chapter gives hints on the selection of appropriate forms, and so do the relations of the previous two propositions.

Example 3.5.1 *Consider the following (OCP) where all variables are scalars:*

$$\max[-(u_0)^2 + (u_1)^2 - 2(x_1 - 1)^2]$$

$$x_1 = x_0 + u_0$$

$$x_2 = x_1 + u_1$$

$$x_0 = \underline{x}_0 = 0$$

$$x_2 = \underline{x}_N = 2$$

We cannot solve this problem by Lagrangian relaxation of the dynamic equation due to the term $(u_1)^2$. *Let us define* π_i *as follows*

$$\pi_i(x_i) = p_i x_i + x_i' A_i x_i$$

where $\pi_i \in R^n$ *and* A_i *are* $n \times n$ *matrices (in this case scalars, since* $n = 1$*).*

The expression to be maximized in (3.22) is

$$-(u_0)^2 + p_1(x_0 + u_0) + (x_0 + u_0)A_1(x_0 + u_0) + (u_1)^2 - 2(x_1 - 1)^2$$

$$-p_1 x_1 - x_1 A_1 x_1 + p_2(x_1 + u_1) + (x_1 + u_1)A_2(x_1 + u_1) - p_2 x_2 - x_2 A_2 x_2$$

x_0 *and* x_N *are fixed. We find that the second derivative with respect to* u_0 *is* $2(A_1 - 1)$. *This is negative for* $A_1 < 1$. *Let us tentatively choose* $A_1 = 0$. *We now find the Hessian with respect to* (x_1, u_1) *as* $\begin{pmatrix} 2A_2 - 4 & 2A_2 \\ 2A_2 & 2 + 2A_2 \end{pmatrix}$ *The diagonal elements are negative for* $A_2 < -2$, *and the determinant is positive also for these values. The Hessian is therefore negative definite for* $A_2 < -2$. *We may then choose e.g.* $A_1 = 0$, $A_2 = -3$, *and then there are finite, unique maximizing values in (3.22).*

Letting $p_1 = 0$ *and* $p_2 = 8$ *permits the solution* $(x_0^*, u_0^*) = (0, 0)$ *and* $(x_1^*, u_1^*) = (0, 2)$. *Since the dynamic equations are fulfilled for these values, they are optimal.*

What makes this simple solution procedure applicable here is the term $-2(x_i - 1)^2$ *in the criterion. As seen, the introduction of the terms* $\pi_{i+1}(f_i(x_i, u_i))$ *give a linkage between* x_i *and* u_i. *Therefore the strict convexity of the criterion with respect to* u_1 *is eliminated by the coupling to* x_1. *The criterion is sufficiently concave with respect to* x_1 *to provide a strictly concave expression with respect to* (x_1, u_1). □

Applying the Dual Gap

Now consider the deviation between the upper boundaries and the supports. A relation was given in Proposition 3.5.1. We give a result which expresses necessary conditions for optimality, in terms of values that an optimal x_i^* or (x_i^*, u_i^*) must necessarily attain.

Let there be given functions π_i^o, and let (x_i^o, u_i^o) be solutions in the relaxed problem (3.26) - (3.31). Let (x_i', u_i') denote any feasible solution in the OCP (3.17) - (3.20). Define now the *empirical gap*, G^e, as follows

$$G^e = \sum_{i=0}^{N-1} (r_i(x_i^o, u_i^o) - \pi_i(x_i^o) + \pi_{i+1}(f_i(x_i^o, u_i^o))) + r_N(x_N^o) + \pi_0(x_0^o) - \pi_N(x_N^o) \quad (3.38)$$
$$-(\sum_{i=0}^{N-1} r_i(x_i', u_i') + r_N(x_N'))$$

An estimate of G^e (with application of linear π_i) was given i Aubin and Ekeland (1976). Application of G^e is indicated in the following result.

Proposition 3.5.7 *Let* (x^*, u^*) *be an optimal solution to OCP. Let there be given a* π^o *and the corresponding relaxed solution* (x^o, u^o) *to (3.26) - (3.31). Let there be given an empirical gap* G^e. *Then for* $i = 0, \ldots, N-1$ *there holds*

$$UB_i(x_i^*) \geq -\pi_i^o(x_i^*)$$

$$+\sum_{j=0}^{i-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))+\pi_0^o(x_0^o)-G^e,$$

there holds for $i=N$

$$UB_N(x_N^*)\geq -\pi_N^o(x_i^*)$$

$$+\sum_{j=0}^{N-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))+\pi_0^o(x_0^o)+r_N(x_N^o)-G^e$$

and for $i=0,\ldots,N$ *there holds*

$$RUB_i(x_i^*)\geq \pi_i^o(x_i^*)$$

$$+\sum_{j=i}^{N-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))+\pi_N^o(x_N^o)+r_N(x_N^o)-G^e$$

Proof. By considering the truncated problem starting at stage i with x_i^* we get from Proposition 3.5.1

$$\sum_{j=i}^{N-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))+r_N(x_N^o)-\pi_N^o(x_N^o)+\pi_i^o(x_i^*)$$
$$\geq \sum_{j=i}^{N-1}r_j(x_j^*,u_j^*)+r_N(x_N^*)$$

Add to both sides of this the expression

$$UB_i(x_i^*)+\sum_{j=0}^{i-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))+\pi_0^o(x_0^o)$$

Observing that

$$UB_i(x_i^*)+\sum_{j=0}^{i-1}r_j(x_j^*,u_j^*)+r_N(x_N^*)=UB_N(x_N^*)$$

and

$$\sum_{j=0}^{N-1}(r_j(x_j^o,u_j^o)+r_N(x_N^o)-\pi_i^o(x_i^o)+\pi_{i+1}^o(f_i(x_i^o,u_i^o)))$$

$$-\pi_N(x_N^o)+\pi_0^o(x_0^o)-UB_N(x_N^*)\geq G^e$$

this can be rearranged to the expression above.

Similar holds for the proof for $i=N$.

For the proof concerning RUB_i we have by considering the truncated problem ending at x_i^*

$$\sum_{j=0}^{i-1}(r_j(x_j^o,u_j^o)-\pi_j^o(x_j^o)+\pi_{j+1}^o(f_j(x_j^o,u_j^o)))-\pi_0^o(x_0^o)-\pi_i^o(x_i^*)\geq\sum_{j=0}^{i-1}r_j(x_j^*,u_j^*)$$

Add to both sides of this

$$RUB_i(x_i^*) + \sum_{j=i}^{N-1}(r_j(x_j^o, u_j^o) + -\pi_j^o(x_j^o) + \pi_{j+1}^o(f_j(x_j^o, u_j^o))) + r_N(x_N^o) - \pi_N^o(x_N^o)$$

Then after rearrangement the result is obtained. □

We have the following corollary which shows the application to any subsequence of stages:

Corollary. *With the notation of the above Proposition we have for any integers s and t satisfying $0 \leq s \leq t \leq N-1$:*

$$\sum_{j=s}^{t} r_j(x_j^*, u_j^*) \geq$$

$$\sum_{j=s}^{t}(r_j(x_j^o, u_j^o) - \pi_j^o(x_j^o) + \pi_{j+1}^o(f_j(x_j^o, u_j^o))) + \pi_s^o(x_s) - \pi_{t+1}^o(x_{t+1}^o) - G^e$$

Proof. Consider the problem for the OCP with start index $i = s$ and final index $i = t+1$, with start point x_s^* and end point x_{t+1}^*. For this the empirical gap G^e found for the whole problem is also valid. The Proposition then applies directly to this. □

This kind of results can be used in a way similar to the way Branch-and-Bound is used. Often Branch-and-Bound is applied within an integer-linear context, where the relaxation is linear programming relaxation of the integer constraints, or Lagrangian relaxation of some constraints. As the results above show, also non-linear supports may be applied.

3.6 Duality

We have in Proposition 1.4.8 given the classical results for linear problems. Duality in relation to convex (i.e., concave criterion function for maximization, linear dynamics and convex local constraints) discrete time optimal control problems was treated in Outrata (1978) and Rockafellar (1988).

For convex problems linear dual functions work well, and part of the formulations is similar to the Lagrangian relaxation, see (1.66) - (1.70) and Proposition 3.5.6. An advantage of the duality framework is that it permits derivation of guidelines for search for the appropriate values of the multipliers p_i that make the sufficient optimality conditions be fulfilled. Also in terms of interpretation of the solutions, of alternative problem formulations and in other ways the duality framework is rich.

The formulation with linear dual functions permits sufficient optimality conditions to be formulated, however, the necessary optimality conditions are dependent on convexity. As already seen in Propositions 3.5.2 and 3.5.3, we may attain sufficient optimality conditions, even for nonconvex problems. This is obtained at the expense of working with nonlinear functions π_i. We shall now develop the essentials of the duality framework in order to take advantage of this.

The results parallel those of mathematical programming duality theory. It may be expressed in different forms, for instance in terms of conjugate functions or superdifferentiability (see e.g. Tind and Wolsey (1981), Outrata (1978), Outrata and Jarusek (1984/85), Stoer and Witzgall (1970)).

The essential point in our formulation is that we give conditions that permit the relaxed problem to be solved stagewise, such that the maximization with respect to (x,u) is decomposed into N independent problems, cf. (3.26) - (3.31). Thus, the claim that application of nonlinear price functions for decomposition destroys decomposability is not valid in this formulation.

For convenience, define the functions C_i as

$$C_0(x_0,u_0,\pi_0,\pi_1) = r_0(x_0,u_0) + \pi_1(f_0(x_0,u_0)) \tag{3.39}$$

$$C_i(x_i,u_i,\pi_i,\pi_{i+1}) = r_i(x_i,u_i) - \pi_i(x_i) + \pi_{i+1}(f_i(x_i,u_i)) \tag{3.40}$$

$$C_N(x_N,\pi_N) = R_N(x_N) - \pi_N(x_N) \tag{3.41}$$

Further define $V = V_0 \times \ldots \times V_N$, $\pi = (\pi_0,\ldots,\pi_N)$, $C(x,u,\pi) = \sum_{i=0}^{N-1} C_i(x_i,u_i,\pi_i,\pi_{i+1}) + C_N(x_N,\pi_N)$ and $R(x,u) = \sum_{i=0}^{N-1} r_i(x_i,u_i) + r_N(x_N)$. As before, OCP is the problem (3.17) - (3.20).

It is seen that the relaxed problem (3.26) - (3.31) in this notation may be written as

$$\max_{(x,u)\in V} [C(x,u,\pi)] \tag{3.42}$$

Thus, $C(x,u,\pi)$ is the criterion function (3.22) and V correspond to the constraints (3.23) - (3.24). Further, the criterion functions in the decomposed problems (3.26) - (3.31) correspond to C_i.

The weak duality result of Proposition 3.5.1, and the sufficient optimality condition of Proposition 3.5.2 may in this notation be formulated as

Proposition 3.6.1 *If OCP has a feasible solution $(x^\diamond,u^\diamond)$ then for any π*

$$R(x^\diamond,u^\diamond) \leq \sup_{(x,u)\in V} [C(x,u,\pi)]$$

If (x^o,u^o) is a solution to the relaxed problem (3.42) and satisfies the dynamic equation (3.18) then $C(x^o,u^o,\pi) = R(x^o,u^o)$, and (x^o,u^o) solves the OCP.

The functions π_i may be selected according to various criteria. We want to select them such that the sufficient optimality conditions hold, however, we may have to restrict them to a class of functions that are manageable, such as for instance linear. For all functions π_i, we require the following: if $a \in R^n$, $b \in R^n$, then

$$\inf_{\pi\in\Pi} [\pi_i(a) - \pi_i(b)] = -\infty \text{ if } a \neq b \tag{3.43}$$

Apart from this requirement, we may on particular circumstances restrict the functions considered to a particular class, e.g. linear. Thus, Π denotes a subset of the functions $\pi_i : R^N \to R$ satisfying (3.43).

With this, we may write the OCP as follows: Find a $(x^o,u^o,\pi^o) \in V \times \Pi$ such that

$$\sup_{(x,u)\in V} [\inf_{\pi\in\Pi} [C(x,u,\pi)]] = C(x^o,u^o,\pi^o) \tag{3.44}$$

We denote this POCP, the primal optimal control problem, which then leads to the following definition of DOCP, the dual optimal control problem: Find (x^D,u^D,π^D) such that

$$\inf_{\pi\in\Pi} [\sup_{(x,u)\in V} [C(x,u,\pi)]] = C(x^D,u^D,\pi^D) \tag{3.45}$$

Note that neither POCP nor DOCP is decomposable in the sense that they reduce to N independent subproblems like (3.26) - (3.31).

Proposition 3.6.2 *OCP and POCP are the same problem in the sense that (1) if OCP has a solution* (x^*, u^*) *then* $(x^*, u^*, \pi^\diamond)$ *solves POCP for any* $\pi^\diamond \in \Pi$*; (2) if POCP has a solution* (x^o, u^o, π^o) *then* (x^o, u^o) *is a solution to OCP.*

In either case, $R(x^*, u^*) = R(x^o, u^o) = C(x^o, u^o, \pi^o) = C(x^*, u^*, \pi^\diamond)$.

Proof. Since $R(x^*, u^*) = C(x^*, u^*, \pi^\diamond) \geq R(x, u) \geq C(x, u, \pi)$ for any $(x, u) \in V$ satisfying the dynamic constraint (3.18) and for any $\pi \in \Pi$; and since by the assumption (3.43) $\inf[C(x, u, \pi)] = -\infty$ for any $(x, u) \in V$ not satisfying the dynamic constraint, (x^*, u^*) together with any $\pi^o \in \Pi$ solves POCP. Conversely, if (x^o, u^o, π^o) solves POCP then (x^o, u^o) satisfies the dynamic equations due to (3.43). Moreover, it is optimal, i.e., a solution to OCP, due to the "sup" in (3.44), i.e., $C(x^o, u^o, \pi^o) = R(x^o, u^o) \geq R(x, u)$ for any $(x, u) \in V$ satisfying the dynamic equation. □

The weak duality result of Proposition 3.5.1 and the sufficiency result of Proposition 3.5.2 are included in the following result.

Proposition 3.6.3

$$\sup_{(x,u)\in V}[\inf_{\pi\in\Pi}[C(x, u, \pi)]] \leq \inf_{\pi\in\Pi}[\sup_{(x,u)\in V}[C(x, u, \pi)]]$$

Moreover,

$$\sup_{(x,u)\in V}[\inf_{\pi\in\Pi}[C(x, u, \pi)]] = \inf_{\pi\in\Pi}[\sup_{(x,u)\in V}[C(x, u, \pi)]] = C(x^o, u^o, \pi^o)$$

for some (x^o, u^o, π^o) *if and only if* (x^o, u^o, π^o) *is a solution to both POCP and DOCP.*

Proof. $\inf_{\pi\in\Pi}[C(x, u, \pi)] \leq C(x, u, \pi)$ for all $\pi \in \Pi$; hence

$$\sup_{(x,u)\in V}[\inf_{\pi\in\Pi}[C(x, u, \pi)]] \leq \sup_{(x,u)\in V}[C(x, u, \pi)]$$

for all $\pi \in \Pi$. As a consequence, the first part of the Proposition holds. The second part follows directly from the definition of the problems POCP and DOCP. □

If equality does not hold in Proposition 3.6.3 we have a positive dual gap, where the dual gap $G(\Pi)$ is defined as

$$G(\Pi) = \inf_{\pi\in\Pi}[\sup_{(x,u)\in V}[C(x, u, \pi)]] - \sup_{(x,u)\in V}[\inf_{\pi\in\Pi}[C(x, u, \pi)]] \tag{3.46}$$

Observe that $G(\Pi)$ depends on the class Π, as well as on the particular OCP.

In Proposition 3.5.7 we used the empirical gap G^e. The relation between the two gaps obviously is $G(\Pi) \leq G^e$, (provided $\pi \in \Pi$, where π is used in the definition of G^e) since in the definition of G^e a particular π and an arbitrary feasible solution to the OCP are used, while the definition of $G(\Pi)$ depends on the optimal $\pi \in \Pi$ as well as on the optimal solution to the OCP.

By Proposition 3.6.3, $G(\Pi)$ is nonnegative for any Π'. If $G(\Pi')$ is positive for the class Π' chosen, the gap may be reduced by widening Π':

Proposition 3.6.4 *If* $\Pi' \subset \Pi$ *then* $G(\Pi) \leq G(\Pi')$.

Proof.

$$\sup_{(x,u)\in V}[\inf_{\pi\in\Pi}[C(x,u,\pi)]] = \sup_{(x,u)\in V}[\inf_{\pi\in\Pi'}[C(x,u,\pi)]]$$

by Proposition 3.6.2 and

$$\inf_{\pi\in\Pi}[\sup_{(x,u)\in V}[C(x,u,\pi)]] \leq \inf_{\pi\in\Pi'}[\sup_{(x,u)\in V}[C(x,u,\pi)]]$$

since $\Pi' \subset \Pi$. Combining the results yields the desired conclusion. □

Under the lenient assumptions of Proposition 3.5.3 the gap may be reduced to zero by letting Π be the class of all functions satisfying (3.43): *Assume that r_i is bounded on V_i for all i and that there exists a solution to the OCP. Then there exist π such that the solution to the relaxed problem satisfies the dynamic equation and the dual gap is zero.*

We give further a formulation in terms of saddle points. Thus, define a saddle point as a triple (x^S, u^S, π^S) satisfying

$$C(x,u,\pi^S) \leq C(x^S,u^S,\pi^S) \leq C(x^S,u^S,\pi) \tag{3.47}$$

for all $(x,u) \in V$ and all $\pi \in \Pi$.

Proposition 3.6.5 *If (x^S,u^S,π^S) is a saddle point then (x^S,u^S,π^S) solves both POCP and DOCP.*

If (x^*,u^*,π^*) solves POCP, if (x^D,u^D,π^D) solves DOCP and if $C(x^*,u^*,\pi^*) = C(x^D,u^D,\pi^D)$ then (x^*,u^*,π^D) is a saddle point.

Proof. Assume that $C(x^S,u^S,\pi^S)$ is a saddle point. Then by definition, $C(x,u,\pi^*) \leq C(x^S,u^S,\pi^S)$ $\leq C(x^S,u^S,\pi)$, implying

$$\min_{\pi\in\Pi}[C(x^S,u^S,\pi)] = C(x^S,u^S,\pi^S) = \max_{(x,u)\in V}[C(x,u,\pi)]$$

and therefore (x^S,u^S,π^S) solves both POCP and DOCP. If on the other hand (x^*,u^*,π^*) solves POCP and (x^D,u^D,π^D) solves DOCP and the two optimal criterion values are equal then (with $(x,u) \in V$ and $\pi \in \Pi$ and inf and sup taken with respect to the relevant variables and functions)

$$\begin{aligned}\inf[C(x,u,\pi)] \leq C(x^*,u^*,\pi^*) &= \inf[C(x^*,u^*,\pi)] = C(x^D,u^D,\pi^D)\\ &= \sup[(x,u,\pi^D)] \leq \sup[C(x,u,\pi)]\end{aligned}$$

which implies

$$\begin{aligned}C(x^*,u^*,\pi^*) = \inf[C(x^*,u^*,\pi)] &\leq \inf[C(x^*,u^*,\pi^o)]\\ &\leq \sup[C(x,u,\pi^o)] = C(x^D,u^D,\pi^o)\end{aligned}$$

and therefore

$$C(x^*,u^*,\pi^*) = C(x^*,u^*,\pi^D) = C(x^D,u^D,\pi^D)$$

Together with the relations $C(x^*,u^*,\pi^*) \leq C(x^*,u^*,\pi)$ and $C(x,u,\pi^D) \leq C(x^D,u^D,\pi^D)$ this gives the saddle point relation for (x^*,u^*,π^D):

$$C(x,u,\pi^*) \leq C(x^*,u^*,\pi^D) \leq C(x^*,u^*,\pi)$$

□

Corollary *(x^S, u^S, π^S) is a saddle point if and only if (x^S, u^S, π^S) solves both POCP and DOCP.*

Proof. The saddle point provides a solution to both POCP and DOCP. If (x^S, u^S, π^S) solves both POCP and DOCP then the optimal values in both cases is $C(x^S, u^S, \pi^S)$, and therefore (x^S, u^S, π^S) is a saddle point. □

We have in this section shown how the duality perspective may be applied to the OCP, with emphasis on a general approach not limited to application of linear price functions. The formulations permit the relaxed problem to be solved independently stagewise. As expected, some of the result are restatements of results from the previous section in relation to relaxation with price functions. Therefore also the characterization of the functions π_i given in that section, see Propositions 3.5.4 and 3.5.5, carry over to the duality formulations here. In fact, the duality relations are intimately related to all the previous optimality and maximum principles through Propositions 3.5.4 and 3.5.5.

The duality perspective adds to the previous ones in various ways. What is added specifically in this section is the demonstration that a systematic search for π_i that permits solution of the OCP using relaxation may be performed. This is done by formulating the DOCP and showing its relations to POCP and relaxation of the POCP. Thus, search for appropriate functions π_i is itself an optimization problem - it may even have the structure of an OCP, as is the case with linear problems, cf. Proposition 1.4.8.

3.7 Conclusions

The best known necessary and sufficient optimality conditions in relation to the OCP is the classical maximum principle. This is an elegant and useful condition, however dependent on smoothness and convexity assumptions.

In this chapter we have shown that a number of alternative optimality conditions are available for the OCP. While the stagewise formulation of the maximum principle is conserved, the optimality conditions differ according to the optimization variable and with respect to the auxiliary functions π_i used.

One generalization of the classical maximum principle is to abandon the differentiability assumptions. In this case the adjoint equation - a stationarity condition similar to the KKT condition - will be substituted by the adjoint relation, using the concept of subdifferential.

In case of non-convex upper boundaries the maximization of the Hamiltonian may of course be substituted by such stationarity condition. However, another approach is to apply non-linear π_{i+1} in the definition of the Hamiltonian. This will permit that maximization of the Hamiltonian by the optimal control is always possible; if π_{i+1} is nonsmooth then again the adjoint equation must be substituted by the adjoint relation.

As has been shown, these considerations on the maximum principle are heavily related to the properties of the upper boundaries, treated in Chapter 2. As this in turn is inseparable from the dynamic programming (including optimal evolution) perspective, it follows that maximum principle and dynamic programming approaches are in turn closely related.

Without suitable convexity assumptions the maximum principles of Section 3.4 only provide necessary optimality conditions, and in this respect they are fundamentally different from the

dynamic programming perspective. The explanation is that those maximum principles relate to the smaller upper boundaries while dynamic programming relate to the greater upper boundaries.

This is changed in the perspectives of Section 3.5. Here, maximization at stage i is performed simultaneously with respect to x_i and u_i, and π_i as well as π_{i+1} enter in the expression to be maximized. Now, sufficient optimality conditions are obtained without convexity assumptions. Further, necessary optimality conditions are obtained under lenient assumptions. Again, there are strong relations to the upper boundaries.

Finally, in Section 3.6 another perspective is applied to the characterization of π_i. By development of the duality framework with nonlinear dual functions and stagewise decomposition it is shown that the appropriate π_i's may be characterized as optimal solutions to the dual optimal control problem. The π_i's have strong relationships to the upper boundaries in the sense that the properties of these latter ones impose limitations on the possible choices of π_i. This in turn implies that the different optimality conditions have strong relations between them, and that the key to the understanding is the concept of upper boundary.

Chapter 4

Dynamic Programming

Dynamic programming is probably the most popular of the optimization techniques for optimal control in discrete time. It is usually used in the backwards direction. It consists in recursively backwards, $N-1,\ldots,0$, solving the problems

$$RUB_i(x_i) = \max_{u_i}[r_i(x_i,u_i) + RUB_{i+1}(f(x_i,u_i))] \tag{4.1}$$

$$u_i \in U_i(x_i) \tag{4.2}$$

This determines the upper boundary functions RUB_i and the optimal control functions $u_i^*(.)$. Knowing x_0^* (or computing it by optimization) the optimal controls and states are then calculated recursively forwards, $0,\ldots,N-1$, as $x_{i+1}^* = f_i(x_i^*,u_i^*(x_i))$. Dynamic programming is treated in a number of books, see. e.g. Bellman (1957), Bellman and Dreyfus (1962), Nemhauser (1966), Dreyfus and Law (1977), Denardo (1982), Bertsekas (1987), Sniedovich (1992).

A main problem in the application of dynamic programming is how to represent the upper boundaries RUB_i. With this given, the way of solving (4.1) - (4.2) is often obvious.

Solution techniques therefore fall in two broad classes depending on whether the state space for definition of RUB_i (or UB_i) is discrete or continuous.

If the state space is discrete and finite the simplest technique of solving (4.1) - (4.2) is to compute all relevant combinations of x_i and u_i and for each x_i select the u_i that maximizes in (4.1); this determines the value $RUB_i(x_i)$ and the optimal control $u_i^*(x_i)$.

The most advantageous way to organize the enumeration depends on the structure of the problem (4.1) - (4.2), in particular how much information that may be reused. Thus for instance if organizing the enumeration in nested loops ('for $a := 1$ to A do for $b := 1$ to B do ...'), x_i, u_i or x_{i+1} may be in the outer loop. See e.g. Denardo (1982), Ch. 2.

Rather than using complete enumeration, various techniques such as implicit enumeration, bounding, fathoming and others may be applied. As this will necessarily be highly problem specific, a multitude of techniques has been used, see e.g. Morin (1978) for a survey.

If the state space is continuous there are basically two possibilities. Either the problem is solved with full exploitation of this, or the problem is approximated by a problem with discrete state space, and solved as just described. In Section 4.1 we discuss the errors encountered from discretization.

Solving a problem with continuous state space is possible only if the functions RUB_i and u_i^* can be easily represented and analyzed in terms of the optimization (4.1) - (4.2).

The best known example of this is the problem with quadratic strictly concave criterion function and linear dynamics. Here, RUB_i is given as a quadratic function $x_i'Q_ix_i + P_ix_i + \rho_i$ and $u_i^*(.)$ is given as a linear function $u_i^*(x_i) = K_i + L_ix_i$, where Q_i, P_i, K_i, L_i and ρ_i are matrices of appropriate dimensions. We develop details of this in Section 4.2 and also treat the case with local linear equality constraints in Sections 4.3 (the backwards direction) and 4.7 (the forwards direction).

If inequality constraints are present, then RUB_i in the quadratic case is *piecewise* quadratic, and u_i^* is *piecewise* linear. This greatly increases the difficulties. Bannister and Kaye (1991) treated the linear case and Jørgensen (1993), Jørgensen and Ravn (1997) treated the linear and the quadratic cases. In Section 4.4 we show how to treat this case when $n = 1$ and $m = 1$, and we show in Section 4.6 that the technique may be extended to problems with $m \geq 1$, although still with $n = 1$.

If the problem has continuous state space but is not solvable explicitly, approximations may be made that preserve the continuous state space, using for instance polynomials or splines. This has the advantage over the discretization that the approximation is better (for comparable complexity of the representation), but the optimization in (4.1) - (4.2) may be more difficult, e.g. not being performed in a finite number of arithmetic operations. Recent development is reported in Johnson, Stedinger, Shoemaker, Li and Tejada-Guibert (1993).

The forwards direction may also be used in dynamic programming. This is based on Proposition 3.2.2 and recursively determines UB_i and $u_i^*(.)$ (depending on x_{i+1}) from

$$UB_{i+1}(x_{i+1}) = \max_{x_i,u_i}[r_i(x_i,u_i) + UB_i(x_i)] \tag{4.3}$$

$$(x_i,u_i) \in V_i \tag{4.4}$$

$$x_{i+1} = f_i(x_i,u_i) \tag{4.5}$$

Similar observations as above hold for the computational aspects. Thus, from a computational point of view it depends on the specific problem, whether the forwards direction or the backwards direction is preferable. For the quadratic-linear problem the backwards direction is from this point of view preferable, cf. Section 4.7.

If the purpose of the OCP is not the determination of the optimal control, or not this alone, the situation may be different. If for instance the final x_N is not given but choice of x_N is to be evaluated, then it may be desirable to have information on how the selection of x_N influences the optimal criterion value on the OCP. This is precisely the kind of information that is represented in the function UB_N, and in this case it may be desirable to use the forwards direction in order to find UB_N. If both x_0 and x_N are parameters this may be handled, in particular if the problem is QLE then the DP formulation may be exploited, cf. Dreyfus and Kan (1973) and Section 5.2.

If on the other hand the problem is stochastic, then the backwards direction is preferable, since this permits a solution on feedback form.

4.1 Discretization of the State Space

For many problems the state space is discrete by nature. However, for problems that naturally have a continuous state space, it may be chosen to discretize this as a means to employ the discrete dynamic programming technique. This in fact is a very popular technique.

The attractiveness of this lies in the ease by which the functions RUB_i are represented. Moreover, as a consequence, the optimization with respect to u_i in (4.1) - (4.2) may be simple in theory,

since only those $u_i \in U_i(x_i)$ for which $f_i(x_i, u_i)$ coincides with a predetermined, discretized value of x_{i+1} need be evaluated (we disregard the practical difficulties in actually determining these u_i).

In relation to this technique the question arises how the discretization should actually be chosen, and how well the solution to the discretized problem approaches the solution to the non-discretized problem.

Various suggestions have been made as to how to choose the discretization, see e.g. Denardo (1982) Ch. 4 or Luus (1990), however, in most cases this is probably chosen with an equal distance between the chosen discrete values of the states.

The question of convergence has been posed in Bellman and Dreyfus (1962), and results are given in Fox (1973) and Bertsekas (1975).

We give below a result similar in spirit to that of Bertsekas (1975), for the case of discretization of the state space. As seen, the essential assumptions are Lipschitz continuity of the involved functions and satisfaction of a constraint qualification. Under these assumptions it is known from Section 2.6 that the upper boundaries are Lipschitz continuous, and the linear convergence result is therefore somehow as expected.

We introduce at each stage i a set $G_i \subset R^n$ of grid-points. Let δ_i denote a scalar such that if x_i^* is optimal in the original continuous OCP then there holds

$$\| x_i^* - x_i^o \| \leq \delta_i \text{ for some } x_i^o \in G_i \cap Y_i^g \cap RY_i^g \tag{4.6}$$

In this, Y_i^g and RY_i^g are defined the same way as Y_i and RY_i in Section 2.1 but in relation to the discretized problem. The relation implies that by moving at each stage i the state from x_i^* by a distance of at most δ_i it will be possible to find $x^g = (x_0^{g\prime}, x_1^{g\prime}, \ldots, x_N^{g\prime})'$ which is a grid-point and which together with some u permits a feasible solution. Observe that it is not necessary to know x_i^* in order to estimate δ_i; if (4.6) holds for any $x_i \in Y_i \cap RY_i$ (Y_i and RY_i in the original non-discretized problem) and not only for $x_i = x_i^*$ then it also holds for x_i^*. The difficulty in establishing (4.6) is therefore the determination of the grid-points - i.e. the discretization details - such that $Y_i^g \cap RY_i^g$ "covers" $Y_i \cap RY_i$ sufficiently well. If a discretization with equal distance between gridpoints is used then it is, however, easy to determine δ_i.

Proposition 4.1.1 *Assume that r_i, f_i, g_i and h_i for all i are Lipschitz continuous and that at each stage the problem defining ub_i^{i+1} is tame and satisfies a constraint qualification (as in Proposition 2.6.3) at all points. Let (x^*, u^*) be optimal in the original problem. Then there is a positive constant $\overline{L}$ such that the following hold. Let there be given a set of grid points G_i and corresponding δ_i such that (4.6) holds for all i. Let $(\tilde{x}^g, \tilde{u}^g)$ be optimal in the discretized problem. Then*

$$\sum_{i=0}^{N-1} r_i(\tilde{x}_i^g, \tilde{u}_i^g) + r_N(\tilde{x}_N^g) \geq$$
$$\sum_{i=0}^{N-1} r_i(x_i^*, u_i^*) + r_N(x_N^*) - \overline{L}(\delta_0 + \delta_N + 2\sum_{i=1}^{N-1} \delta_i)$$

Proof. The assumption on the problems defining ub_i^{i+1} implies that ub_i^{i+1} are Lipschitz continuous in the interior of W_i^{i+1}, cf. Proposition 2.6.3, and therefore there is a positive $\overline{L}$ such that

$$ub_i^{i+1}(x_i^g, x_{i+1}^g) \geq ub_i^{i+1}(x_i^*, x_{i+1}^*) - \overline{L}(\delta_i + \delta_{i+1})$$

where x_i^g and x_{i+1}^g are chosen to satisfy (4.6). This holds for all i. Now $x^g = (x_0^{g\prime}, x_1^{g\prime}, \ldots, x_N^{g\prime})'$ and a corresponding u^g is feasible because by (4.6) $x_i^g \in Y_i^g \cap RY_i^g$ for all i. By Proposition 3.3.5

there holds at (x^g, u^g) $\sum_{i=0}^{N} r_i = \sum_{i=0}^{N-1} ub_i^{i+1}$ and therefore we get

$$\begin{aligned}
&\sum_{i=0}^{N-1} r_i(\tilde{x}_i^g, \tilde{u}_i^g) + r_N(\tilde{x}_N^g) \\
&\geq \sum_{i=0}^{N-1} r_i(x_i^g, u_i^g) + r_N(x_N^g) = \sum_{i=0}^{N-1} ub_i^{i+1}(x_i^g, x_{i+1}^g) \\
&\geq \sum_{i=0}^{N-1} ub_i^{i+1}(x_i^*, x_{i+1}^*) - \overline{L}(\delta_0 + \delta_N + 2\sum_{i=1}^{N-1} \delta_i) \\
&= \sum_{i=0}^{N-1} r_i(x_i^*, u_i^*) + r_N(x_N^*) - \overline{L}(\delta_0 + \delta_N + 2\sum_{i=1}^{N-1} \delta_i)
\end{aligned}$$

□

This result is rather disappointing in the sense that convergence of the optimal criterion value of the approximated problem towards the true optimal criterion value as $\delta_i \to 0$ may be expected to be relatively slow.

4.2 The Quadratic Linear Problem

The quadratic linear problem - i.e. the problem with quadratic criterion function, linear dynamics and (if present) linear local constraints - is an important example of a type of problems where dynamic programming may be used to find the solution analytically. We shall in this section consider the unconstrained problem (QL), while the problem QLEI with local linear constraints will be treated in Section 4.3. Special cases with one-dimensional state vector ($n = 1$) will be treated in Section 4.4.

The unconstrained problem QL

We define the unconstrained problem QL as

$$\begin{aligned}
&\max[\sum_{i=0}^{N-1} \tfrac{1}{2}x_i' R_i^{xx} x_i + x_i' R_i^{xu} u_i + \tfrac{1}{2}u_i' R_i^{uu} u_i + R_i^x x_i + R_i^u u_i && (4.7)\\
&+\tfrac{1}{2}x_N' R_N^{xx} x_N + R_N^x x_N] \\
&x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i && (4.8)\\
&x_0 = \underline{x}_0 && (4.9)
\end{aligned}$$

where R_i^{xx}, R_i^{xu}, R_i^{uu}, R_i^x, R_i^u, F_i^x, F_i^u and $\overline{f}_i$ are matrices of dimensions $n \times n$, $n \times m$, $m \times m$, $1 \times n$, $1 \times m$, $n \times n$, $n \times m$, and $n \times 1$, respectively, with R_i^{xx} and R_i^{uu} symmetric.

Eliminating x_N by using the dynamic equation we find that RUB_{N-1} at the point x_{N-1} is defined as

$$\begin{aligned}
&\max_{u_{N-1}}[\tfrac{1}{2}x_{N-1}' R_{N-1}^{xx} x_{N-1} + x_{N-1}' R_{N-1}^{xu} u_{N-1} && (4.10)\\
&+\tfrac{1}{2}u_{N-1}' R_{N-1}^{uu} u_{N-1} + R_{N-1}^x x_{N-1} + R_{N-1}^u u_{N-1}
\end{aligned}$$

$$
\begin{aligned}
&+\tfrac{1}{2}(F^x_{N-1}x_{N-1}+F^u_{N-1}u_{N-1}+\overline{f}_{N-1})'R^{xx}_N\\
&(F^x_{N-1}x_{N-1}+F^u_{N-1}u_{N-1}+\overline{f}_{N-1})\\
&+R^x_N(F^x_{N-1}x_{N-1}+F^u_{N-1}u_{N-1}+\overline{f}_{N-1})]
\end{aligned}
$$

Assuming that the criterion in this is strictly concave we may find the unique maximizing u_{N-1} by differentiating and equating to zero because $U_{N-1}(x_{N-1}) = R^n$ and $RY_N = R^n$. We shall therefore solve the following with respect to u_{N-1}:

$$
\begin{aligned}
&x'_{N-1}R^{xu}_{N-1}+u'_{N-1}R^{uu}_{N-1}+R^u_{N-1}\\
&+(F^x_{N-1}x_{N-1}+F^u_{N-1}u_{N-1}+\overline{f}_{N-1})'R^{xx}_NF^u_{N-1}+R^x_NF^u_{N-1}=0
\end{aligned}
\tag{4.11}
$$

The solution may under the above assumptions be written as

$$
\begin{aligned}
&u_{N-1}=\\
&-(R^{uu}_{N-1}+F^{u\prime}_{N-1}R^{xx}_NF^u_{N-1})^{-1}(R^{xu\prime}_{N-1}+F^{u\prime}_{N-1}R^{xx}_NF^x_{N-1})x_{N-1}\\
&-(R^{uu}_{N-1}+F^{u\prime}_{N-1}R^{xx}_NF^u_{N-1})^{-1}(R^{u\prime}_{N-1}+F^{u\prime}_{N-1}R^{xx}_N\overline{f}_{N-1}+F^{u\prime}_{N-1}R^{x\prime}_N)
\end{aligned}
\tag{4.12}
$$

We observe that the solution u_{N-1} is expressed as a linear function of x_{N-1}. We may therefore insert this into the expression (4.10) for RUB_{N-1} and obtain RUB_{N-1} as a quadratic function defined on $RY_{N-1} = R^n$, cf. also Proposition 2.4.6.

By suitable definition of the matrices Q_{N-1} and P_{N-1} and the constant ρ_{N-1} we may therefore write RUB_{N-1} as

$$
RUB_{N-1}(x_{N-1}) = x'_{N-1}Q_{N-1}x_{N-1}+P_{N-1}x_{N-1}+\rho_{N-1} \tag{4.13}
$$

We see that for the maximization of the backwards dynamic programming criterion $[r_{N-2} + RUB_{N-1}]$ at stage $(N-2)$ we have obtained exactly the same structure as we had in (4.10) at stage $(N-1)$ if we in (4.7) let $Q_N = \frac{1}{2}R^{xx}_N$ and $P_N = R^x_N$. With $i = N-2$ we therefore have that $RUB_i(x_i)$ is defined by:

$$
\begin{aligned}
RUB_i(x_i) \quad &= \max_{u_i}[\tfrac{1}{2}x'_iR^{xx}_ix_i+x'_iR^{xu}_iu_i\\
&+\tfrac{1}{2}u'_iR^{uu}_iu_i+R^x_ix_i+R^u_iu_i\\
&+(F^x_ix_i+F^u_iu_i+\overline{f}_i)'Q_{i+1}(F^x_ix_i+F^u_iu_i+\overline{f}_i)\\
&+P_{i+1}(F^x_ix_i+F^u_iu_i+\overline{f}_i)]
\end{aligned}
\tag{4.14}
$$

(the constant ρ_{i+1} is omitted as it has no influence on the optimization) and the optimal $u_i(x_i)$ is defined by

$$
\begin{aligned}
&u_i=\\
&-(R^{uu}_i+2F^{u\prime}_iQ_{i+1}F^u_i)^{-1}(R^{xu\prime}_i+2F^{u\prime}_iQ_{i+1}F^x_i)x_i\\
&-(R^{uu}_i+2F^{u\prime}_iQ_{i+1}F^u_i)^{-1}(R^{u\prime}_i+2F^{u\prime}_iQ_{i+1}\overline{f}_i+F^{u\prime}_iP'_{i+1})
\end{aligned}
\tag{4.15}
$$

Therefore we can continue this backwards solution procedure to stage 0. Then exploiting the linear relationships found between x_i and u_i we can construct the optimal strategy and trajectory recursively forwards from $\underline{x}_0$.

Let us formalize this. We define $Q_N = \frac{1}{2}R_N^{xx}$, $P_N = R_N^x$ and the following matrices recursively backwards, $i = N-1,\ldots,0$ (dimensions indicated as (rows $\times$ columns)):

$$\begin{aligned}
A_i &= \tfrac{1}{2}(R_i^{xx} + 2F_i^{x\prime}Q_{i+1}F_i^x) \quad (n \times n) && (4.16)\\
B_i &= R_i^{xu} + 2F_i^{x\prime}Q_{i+1}F_i^u \quad (n \times m) && (4.17)\\
C_i &= \tfrac{1}{2}(R_i^{uu} + 2F_i^{u\prime}Q_{i+1}F_i^u) \quad (m \times m) && (4.18)\\
D_i &= R_i^u + 2\overline{f}_i'Q_{i+1}F_i^u + P_{i+1}F_i^u \quad (1 \times m) && (4.19)\\
E_i &= R_i^x + 2\overline{f}_i'Q_{i+1}F_i^x + P_{i+1}F_i^x \quad (1 \times n) && (4.20)\\
K_i &= -\tfrac{1}{2}C_i^{-1}D_i' \quad (m \times 1) && (4.21)\\
L_i &= -\tfrac{1}{2}C_i^{-1}B_i' \quad (m \times n) && (4.22)\\
Q_i &= A_i + \tfrac{1}{2}(B_iL_i + L_i'B_i') + L_i'C_iL_i \quad (n \times n) && (4.23)\\
P_i &= K_i'B_i' + 2K_i'C_iL_i + D_iL_i + E_i \quad (1 \times n) && (4.24)
\end{aligned}$$

Observe that for this QL problem the matrices Q_i and P_i may also be written directly from A_i, B_i, C_i, D_i and E_i as

$$\begin{aligned}
Q_i &= A_i - \tfrac{1}{4}B_iC_i^{-1}B_i' && (4.25)\\
P_i &= -\tfrac{1}{2}D_iC_i^{-1}B_i' + E_i && (4.26)
\end{aligned}$$

If desired, the constant ρ_i may be calculated as

$$\rho_i = \overline{f}_i'Q_{i+1}\overline{f}_i + P_{i+1}\overline{f}_i \tag{4.27}$$

but as observed above, it is not needed.

Further observe that (4.23) and (4.25) may be seen as versions of the Riccati equation, cf. eg. Bertsekas (1987) p. 58, Bertsekas (1976) p. 73. Thus, assuming $R_i^{xu} = 0$, we may reformulate (4.23) and (4.25) as

$$\begin{aligned}
Q_i \quad &= && (4.28)\\
&F_i^{x\prime}(Q_{i+1} - Q_{i+1}F_i^u(F_i^{u\prime}Q_{i+1}F^u + \tfrac{1}{2}R_i^{uu})^{-1}F_i^{u\prime}Q_{i+1})F_i^x + \tfrac{1}{2}R_i^{xx}
\end{aligned}$$

With the definitions (4.16) - (4.24) the backwards dynamic programming criterion at stage i, $[r_i(x_i,u_i) + RUB_{i+1}(f_i(x_i,u_i))]$, cf. (4.14), can be written

$$[x_i'A_ix_i + x_i'B_iu_i + u_i'C_iu_i + D_iu_i + E_ix_i] \tag{4.29}$$

except for a constant which has no influence on the optimization and therefore is omitted. The solution of (4.29) corresponding to (4.15) is then, using (4.21) - (4.22),

$$u_i = K_i + L_ix_i \tag{4.30}$$

By insertion into (4.29) it is verified that by the definitions (4.16) - (4.24) the reverse greater upper boundary criterion at stage i is of the form (4.13), and when also the constant added at this stage, $-D_iC_i^{-1}D_i'/4$, is omitted we therefore have, using (4.23) - (4.24),

$$RUB_i(x_i) = x_i'Q_ix_i + P_ix_i \tag{4.31}$$

The backwards dynamic programming solution can then be formulated as follows: Calculate recursively backwards the above matrices (4.16) - (4.24) starting with $Q_N = \frac{1}{2}R_N^{xx}$ and $P_N = R_N^x$. In the forwards direction we start with the initial point $x_0 = \underline{x}_0$. Then we calculate u_i and x_{i+1} recursively from (4.30) and the dynamic equation (4.8), respectively.

Proposition 4.2.1 *If $C_i < 0$ for all i then RUB_i and the unique optimal solution to the QL problem (4.7) - (4.9) can be found as described above. The computational complexity in $O(Nm^3)$.*

Proof. If $C_i < 0$ for all i then the DP procedure described above is well defined and yields a unique solution to (4.15) because the unique $(C_i)^{-1}$ exists. The solution to (4.15) is actually the maximizing u_i since $C_i < 0$. Therefore the procedure actually determines RUB_i as required in (4.14). By Proposition 3.2.1 the solution is optimal. At each stage the dominating calculations are the derivation of C_i^{-1}, which has a complexity of $O(m^3)$ since C_i is $m \times m$. There are N stages, and consequently the computational complexity of the algorithm is $O(Nm^3)$. □

Optimality was here linked to the conditions $C_i < 0$ for all i. This is related to the concavity of $\sum_{i=0}^{N} r_i$ on the feasible subspace, i.e. on the space satisfying (4.8) - (4.9). If for all i r_i is concave on R^{n+m} and r_N is concave on R^n then clearly $\sum_{i=0}^{N} r_i$ is concave on this subspace. As r_i is quadratic, concavity of r_i is resolved by analysis of the matrices

$$\nabla^2 r_i = \begin{pmatrix} R_i^{xx} & R_i^{ux} \\ R_i^{xu} & R_i^{uu} \end{pmatrix} \tag{4.32}$$

and R_N^{xx}. Clearly, if $\nabla^2 r_i \leq 0$ for all i then $\nabla^2 \sum_{i=0}^{N} r_i \leq 0$ and $\sum_{i=0}^{N} r_i$ is therefore concave, and strictly concave if strict inequality holds, cf. Proposition 2.4.6. Specifically, if $R_i^{xx} \leq 0$, $R_i^{xu} = 0$ and $R_i^{uu} \leq 0$ ($R_i^{uu} < 0$) then $\nabla^2 \sum_{i=0}^{N} r_i \leq 0$ (< 0, respectively).

As we shall now see, weaker conditions may be sufficient. For subsequent reference we observe that we may eliminate the state variables and express the problem exclusively in terms of the control variables. Let $r : R^{Nm} \to R$ denote this criterion:

$$\begin{aligned} r(u) = \; & \\ & r_0(\underline{x}_0, u_0) + r_1(f_0(\underline{x}_0, u_0), u_1) + \dots \\ & + r_N(f_{N-1}(\dots(f_0(\underline{x}_0, u_0), u_1)\dots), u_{N-1}) \end{aligned} \tag{4.33}$$

Clearly $\nabla^2 r < 0$ if and only if $\nabla^2 \sum_{i=0}^{N} r_i < 0$ on the feasible subspace. By the last expression we mean that $(x', u')\nabla^2(\sum_{i=0}^{N} r_i)(x', u')' < 0$ for all $(x', u')' \neq 0$ satisfying (4.8) - (4.9).

The requirement of $C_i < 0$ for all i in the above Proposition 4.2.1 is intimately related to the conditions $\nabla^2 \sum_{i=0}^{N} r_i < 0$ on the feasible subspace:

Proposition 4.2.2 *$C_i < 0$ for all i in the above DP solution if and only if $\nabla^2 \sum_{i=0}^{N} r_i < 0$ on the feasible subspace (4.8) - (4.9).*

Proof. This is a corollary to Proposition 4.3.2. □

If x_0 is free - or constrained to $X_0 \subset R^n$ - rather than given as in (4.9) then the above DP backwards recursion must be followed by finding the optimal x_0^*. This is done by solving the problem

$$\max_{x_0 \in X_0} [RUB_0(x_0)] \tag{4.34}$$

This may be done by defining Q_0 and P_0 as in (4.23) - (4.24), and then $RUB_0(x_0) = x_0' Q_0 x_0 + P_0 x_0$. (4.34) is then solved by any appropriate method. If $Q_0 < 0$ and $X_0 = R^n$ then x_0^* is found as

$$x_0^* = -\tfrac{1}{2} Q_0^{-1} P_0' \tag{4.35}$$

Example 4.2.1 *Let $n = 1$, $m = 1$ and consider the problem*

$$\max[\sum_{i=0}^{2} -(u_i)^2 - (x_3)^2 + 2x_3]$$

$$x_{i+1} = x_i + u_i$$

$$x_0 = 0$$

With $Q_3 = (-1)$ and $P_3 = (2)$ we find from (4.16) - (4.20) at stage 2 that $A_2 = (-1)$, $B_2 = (-2)$, $C_2 = (-2)$, $D_2 = (2)$ and $E_2 = (2)$. This implies by (4.21) - (4.24) that $K_2 = (1/2)$, $L_2 = (-1/2)$, $Q_2 = (-1/2)$ and $P_2 = (1)$.

Then we find $A_1 = (-1/2)$, $B_1 = (-1)$, $C_1 = (-3/2)$, $D_1 = (1)$, $E_1 = (1)$, $K_1 = (1/3)$, $L_1 = (-1/3)$, $Q_1 = (-1/3)$ and $P_1 = (2/3)$.

Finally $B_0 = (-2/3)$, $C_0 = (-4/3)$, $D_0 = (2/3)$, $K_0 = (1/4)$ (and $A_0 = (-1/3)$, $E_0 = (2/3)$, $L_0 = (-1/4)$, but these are not necessary).

With $x_0^* = 0$ we find $u_0^* = 1/4$, $x_1^* = f_0(x_0^*, u_0^*) = 1/4$, $u_1^* = K_1 + L_1 x_1^* = 1/4$, $x_2^* = 1/2$, $u_2^* = 1/4$ and $x_3^* = 3/4$.

This is globally optimal because $C_i < 0$ for $i = 0, 1, 2$.

If x_0 is free then we may calculate $Q_0 = -1/4$ and $P_0 = \frac{1}{2}$ and then from (4.35) find $x_0^* = 1$.

□

4.3 Local Linear Constraints

Now we consider the case QLEI where there are local linear constraints. If the constraints are equality constraints only, the QLE problem, then essentially this reduces to the unconstrained case QL. This is therefore relatively easy.

If there are linear inequality constraints, the QLEI problem, then in general it will be difficult or impossible to solve the problem by dynamic programming. In Section 7.1 we describe how the QLEI problem may be solved iteratively as a sequence of QLE problems by an active set method. In Section 4.4 we show that if $n = 1$ then dynamic programming may be applied.

Linear local equality constraints: QLE

Now suppose that we have local linear equality constraints for $i = 0, \ldots, N-1$

$$H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \tag{4.36}$$

in addition to (4.7) - (4.9). Here H_i^x, H_i^u and $\overline{h}_i$ are matrices of dimensions $(\ell \times n)$, $(\ell \times m)$ and $(\ell \times 1)$, respectively.

We assume a free end point. However, if x_N is fixed at $\underline{x}_N$ this may be represented in the form (4.36) as

$$F_{N-1}^x x_{N-1} + F_{N-1}^u u_{N-1} + \overline{f}_{N-1} - \underline{x}_N = 0 \tag{4.37}$$

Let the local backwards dynamic programming criterion $[r_i + RUB_{i+1}(f_i)]$ be written in the form (4.29). Then the KKT conditions for the maximization of the Hamiltonian can by introducing

the row vector $\mu_i \in R^\ell$ be written as:

$$B_i'x_i + 2C_iu_i + D_i' - H_i^{u\prime}\mu_i = 0 \tag{4.38}$$

$$H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \tag{4.39}$$

Multiplying (4.38) by -1 we can write this system as

$$\begin{pmatrix} -2C_i & H_i^{u\prime} \\ H_i^u & 0 \end{pmatrix} \begin{pmatrix} u_i \\ \mu_i' \end{pmatrix} = \begin{pmatrix} D_i' \\ \overline{h}_i \end{pmatrix} + \begin{pmatrix} B_i' \\ -H_i^x \end{pmatrix} x_i \tag{4.40}$$

Under suitable regularity assumptions the solution is then given as

$$\begin{pmatrix} u_i \\ \mu_i' \end{pmatrix} = \tag{4.41}$$

$$\begin{pmatrix} -2C_i & H_i^{u\prime} \\ H_i^u & 0 \end{pmatrix}^{-1} \begin{pmatrix} D_i' \\ \overline{h}_i \end{pmatrix} + \begin{pmatrix} -2C_i & H_i^{u\prime} \\ H_i^u & 0 \end{pmatrix}^{-1} \begin{pmatrix} B_i' \\ -H_i^x \end{pmatrix} x_i$$

As seen, (u_i, μ_i) is given as a linear expression in x_i; we may indicate the dependency as $(u_i(x_i), \mu_i(x_i))$. By suitable definition of matrices $\underline{K}_i$ and $\underline{L}_i$ of dimensions $((m+\ell) \times 1)$ and $((m+\ell) \times n)$, respectively, we may write the solution (4.41) as

$$\begin{pmatrix} u_i \\ \mu_i' \end{pmatrix} = \underline{K}_i + \underline{L}_i x_i \tag{4.42}$$

Now we denote the upper m rows of $\underline{K}_i$ and $\underline{L}_i$ by K_i and L_i, respectively, and therefore have the expression similar in form to (4.30):

$$u_i = K_i + L_i x_i \tag{4.43}$$

Inserting the expression (4.43) for u_i in $[r_i + RUB_{i+1}]$ given in (4.29) we find RUB_i expressed as a quadratic function in x_i defined on $RY_i = R^n$. By definition of the matrices Q_i and P_i as in (4.23) - (4.24) (using (4.42) - (4.43), not (4.21) - (4.22)) RUB_i may be written as in (4.31) and by definition of the matrices A_i, B_i, C_i, D_i and E_i as in (4.16) - (4.20) the backwards dynamic programming criterion at stage $(i-1)$ can then be written in the form (4.29).

The QLE problem can therefore be solved by one backwards and one forwards recursion as in the case of the QL problem.

If the initial point x_0 is free or partially constrained then the optimal x_0^* must be found as in (4.34).

Proposition 4.3.1 *If for all i the rows of H_i^u are linearly independent and $C_i < 0$ on the subspace $\{z \in R^m \mid H_i^u z = 0\}$ then RUB_i and the unique optimal solution to the QLE problem (4.7) - (4.9), (4.36) can be found as described above. The computational complexity in $O(N(m+\ell)^3)$.*

Proof. Under the assumptions, the solution u_i in (4.41) is unique and the $u_i(x_i)$ found is the optimal solution to the optimization (4.14) constrained by (4.36) cf. Luenberger (1989) p. 425. For the same reasons as in Proposition 4.2.1 the solution is optimal in the problem. At each stage the dominating calculation is the inversion in (4.41), which has a complexity of $O((m+\ell)^3)$. Total computational complexity is therefore $O(N(m+\ell)^3)$. □

Proposition 4.3.2 *Assume that the local constraints are linearly independent with respect to u_i. Then $C_i < 0$ on the subspace $\{z_i \in R^m \mid H_i^u z_i = 0\}$ for all i if and only if $\nabla^2 \sum_{i=0}^N r_i < 0$ on the feasible subspace defined by (4.8) - (4.9) and (4.36).*

Proof. The implication $C_i < 0 \Rightarrow \nabla^2 r < 0$ is shown in Jonson (1983) p. 63 (and in Pantoya (1988) for the locally unconstrained case). Now consider the other implication.

The condition $\nabla^2 \sum_{i=0}^N r_i < 0$ on the feasible subspace means $\delta u' \nabla^2 r \delta u < 0$ for all $(\delta x, \delta u)$, where $\delta u \neq 0$ on the subspace $\{\delta u_0 \in R^m \mid H_0^u \delta u_0 = 0\} \times \ldots \times \{\delta u_{N-1} \in R^m \mid H_{N-1}^u \delta u_{N-1} = 0\}$ (here $\delta u = (\delta u_0', \ldots, \delta u_{N-1}')'$); and $(\delta x, \delta u)$ is feasible in (4.8) - (4.9). Since this holds for arbitrary $\delta u \neq 0$, it also holds for any truncated problem starting at x_i. Therefore the condition implies $\sum_{j=i}^N \delta u_j' \nabla^2 r_j \delta u_j < 0$ on the feasible subspace for all i.

Now, $\sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N)$ is equal to (4.29) under the assumption that optimal controls relative to x_{i+1} are chosen for $j = i+1, \ldots, N-1$ and therefore $\sum_{j=i}^N \delta u_j' \nabla^2 r_j \delta u_j < 0$ on the feasible subspace implies $\delta u_i' C_i \delta u_i < 0$ or $C_i < 0$ on the subspace $\{z_i \in R^m \mid H_i^u z_i = 0\}$, since the optimal controls are a subset of all feasible controls. □

Example 4.3.1 *Let $n = 1$, $m = 1$ and consider the problem*

$$\max[\sum_{i=0}^{3} -(u_i)^2]$$

$$x_{i+1} = x_i + u_i$$

$$x_0 = 0$$

$$h_2(x_2, u_2) = x_2 - u_2 + 1 = 0$$

$$h_3(x_3, u_3) = x_3 + u_3 - 1 = 0$$

The last constraint may be seen as a reformulation of the end point condition $x_4 = 1$, cf. (4.37). With $Q_4 = (0)$, $P_4 = (0)$ we find $A_3 = (0)$, $B_3 = (0)$, $C_3 = (-1)$, $D_3 = (0)$ and $E_3 = (0)$. Then at stage 3 we find the solution corresponding to (4.41)

$$\begin{pmatrix} u_3 \\ \mu_3 \end{pmatrix} = \underline{K}_3 + \underline{L}_3 x_3 = \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ 1 \end{pmatrix} + \begin{pmatrix} 2 & 1 \\ 1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ -1 \end{pmatrix} x_3$$

i.e.

$$\begin{pmatrix} u_3 \\ \mu_3 \end{pmatrix} = \begin{pmatrix} 1 \\ -2 \end{pmatrix} + \begin{pmatrix} -1 \\ 2 \end{pmatrix} x_3$$

The upper row of the two columns identifies $K_3 = (1)$ and $L_3 = (-1)$, cf. (4.42) - (4.43).

We find $Q_3 = (-1)(-1)(-1) = (-1)$ and $P_3 = 2(1)(-1)(-1) = (2)$ from (4.23) - (4.24) and then $A_2 = (-1)$, $B_2 = (-2)$, $C_2 = (-2)$, $D_2 = (2)$, and $E_2 = (2)$ from (4.16) - (4.20). Now the solution corresponding to (4.41) is

$$\begin{pmatrix} u_2 \\ \mu_2 \end{pmatrix} = \underline{K}_2 + \underline{L}_2 x_2 =$$

$$\begin{pmatrix} 4 & -1 \\ -1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} 2 \\ -1 \end{pmatrix} + \begin{pmatrix} 4 & -1 \\ -1 & 0 \end{pmatrix}^{-1} \begin{pmatrix} -2 \\ -1 \end{pmatrix} x_2$$

$$= \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} 1 \\ 6 \end{pmatrix} x_2$$

We find $Q_2 = (-5)$, $P_2 = (-2)$ *and then* $A_1 = (-5)$, $B_1 = (-10)$, $C_1 = (-6)$, $D_1 = (-2)$ *and* $E_1 = (-2)$. *This gives* $u_1 = K_1 + L_1 x_1 = (-1/6) + (-5/6)x_1$. *Finally* $Q_1 = (-5/6)$, $P_1 = (-1/3)$, $B_0 = (-5/3)$, $C_0 = (-11/6)$, $D_0 = (-1/3)$, $K_0 = (-1/11)$ *and* $L_0 = (5/11)$.

Starting with $x_0^* = 0$ *we then find* $u_0^* = -1/11$, $x_1^* = -1/11$, $u_1^* = -1/11$, $x_2 = -2/11$, $u_2^* = 9/11$, $x_3^* = 7/11$, $u_3^* = 4/11$ *and* $x_4^* = 1$. □

Local linear equality and inequality constraints: QLEI

Now introduce the local inequality constraints

$$G_i^x x_i + G_i^u u_i - \overline{g}_i \leq 0 \tag{4.44}$$

in addition to the problem definition (4.7) - (4.9) and (4.36). Here G_i^x, G_i^u and $\overline{g}_i$ are matrices of dimensions $(k \times n)$, $(k \times m)$ and $(k \times 1)$, respectively. This greatly increases the difficulties in application of dynamic programming. This is attributed to the fact that now RUB_i is *piecewise* quadratic, cf. Proposition 2.4.6. Although each piece can be described analytically the difficulties of specifying the set of x_i for which each piece is valid are considerable.

In Section 4.4 we describe how it may be done for $n = 1$. We shall in Section 7.1 describe how the QLEI problem for $n \geq 1$ may be solved iteratively as a sequence of QLE problems by an active set method.

4.4 The QLI Problem with n=1

When $n = 1$ the set RY_i is an interval on the real axis, and this makes a simple explicit specification of RUB_i possible. We shall in the sequel describe backwards dynamic programming under the assumption that $n = 1$, $m = 1$, $\ell = 0$, and for $i = 0, \ldots, N-1$ (4.44) given in the form $\underline{u}_i \leq u_i \leq \overline{u}_i$ and $\underline{x}_i \leq x_i \leq \overline{x}_i$, $F_i^x > 0$, $F_i^u > 0$, and r_i concave with respect to (x_i, u_i) and strictly concave with respect to u_i. Assuming also r_N strictly concave then this implies that RUB_i is strictly concave for all i, cf. Proposition 2.4.6. More general problems, but still with $n = 1$, will be treated in Section 4.6.

The idea in the following it that RUB_i will be piecewise quadratic on the interval RY_i and this permits a simple representation of RUB_i and an analytical treatment of the dynamic programming recursive optimization (4.1) - (4.2). We proceed as follows.

As RUB_i is piecewise quadratic and concave it can be specified as follows. There is given a set of breakpoints $\tilde{x}_i^j$, $j = 1, \ldots, \omega_i$. RUB_i is here defined on $RY_i = \{x_i \mid \tilde{x}_i^1 \leq x_i \leq \tilde{x}_i^{\omega_i}\}$. From $\tilde{x}_i^j$ to $\tilde{x}_i^{j+1}$, $j = 1, \ldots, \omega_i - 1$, RUB_i is specified by (Q_i^j, P_i^j, ρ_i^j) such that

$$RUB_i(x_i) = x_i' Q_i^j x_i + P_i^j x_i + \rho_i^j \tag{4.45}$$

As RUB_i is continuous only one ρ_i^j is needed at each stage. Moreover, as we shall see below only the slope of RUB_i is needed in order to find the optimal solution. Therefore we shall not need any ρ_i^j, and they will not be specified.

At the points x_i where RUB_i is differentiable we have $\nabla RUB_i(x_i) = 2x_i' Q_i^j + P_i^j$. At the breakpoints $\tilde{x}_i^j$ RUB_i need not be differentiable. There the directional derivatives to the right and to the left, ∇^+ and ∇^-, respectively, exist and satisfy

$$\begin{aligned} &\nabla^+ RUB_i(\tilde{x}_i^j) = 2\tilde{x}_i^{j\prime} Q_i^j + P_i^j \leq \\ &2\tilde{x}_i^{j\prime} Q_i^{j-1} + P_i^{j-1} = \nabla^- RUB_i(\tilde{x}_i^j) \end{aligned} \tag{4.46}$$

with equality if and only if RUB_i is indeed differentiable at this point. At $\tilde{x}_i^1$ and $\tilde{x}_i^{\omega_i}$ only the left and the right part, respectively, of (4.46) applies.

We shall now specify how to find RUB_i assuming RUB_{i+1} known. The points $\tilde{x}_i^1$ and $\tilde{x}_i^{\omega_i}$ are defined as follows, assuming $F_i^x > 0$ and $F_i^u > 0$:

$\tilde{x}_i^1 = \max\{\underline{x}_i, (F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u\overline{u}_i - \overline{f}_i)\}$

$\tilde{x}_i^{\omega_i} = \min\{\overline{x}_i, (F_i^x)^{-1}(\tilde{x}_{i+1}^{\omega_{i+1}} - F_i^u\underline{u}_i - \overline{f}_i)\}$.

If the problem has a feasible solution then $\tilde{x}_i^1 \leq \tilde{x}_i^{\omega_i}$. Observe that we do not know the value of ω_i yet (except at stage N where $\omega_N = 2$ if $\underline{x}_N < \overline{x}_N$ and $\omega_N = 1$ if $\underline{x}_N = \overline{x}_N$).

We shall temporarily identify the remaining breakpoints of RUB_i as x_-, x_+, x_-^j and x_+^j, where $x_- \leq x_+$ and $x_-^j \leq x_+^j \leq x_-^{j+1} \leq x_+^{j+1}$; some points may coincide.

For a given x_i and RUB_{i+1} the optimal $u_i(x_i)$ is found as the solution to

$$\max_{u_i}[\tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^u u_i + R_i^x x_i \qquad (4.47)$$
$$+RUB_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i)]$$

subject to $\underline{u}_i \leq u_i \leq \overline{u}_i$ and $x_{i+1}^1 \leq F_i^x x_i + F_i^u u_i + \overline{f}_i \leq x_{i+1}^{\omega_i}$.

We are now interested in the point x_+ where the restriction $\underline{u}_i \leq u_i(x_i)$ changes from begin inactive to being active as x_i is increased, disregarding for the moment the state constraints. If RUB_{i+1} is differentiable then the derivative of the expression (4.47) with respect to u_i is

$$x_i'R_i^{xu} + u_i'R_i^{uu} + R_i^u + F_i^u\nabla RUB_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) \qquad (4.48)$$

and $u_i(x_i)$ is found as the value which makes this vanish. As RUB_{i+1} is in general not differentiable we adapt to this situation by using (4.46). If $R_i^x \leq 0$, $F_i^x > 0$, $F_i^u > 0$ and RUB_{i+1} is strictly concave the expressions in (4.48) (cf. also (4.46)) are strictly decreasing with respect to x_i and we may therefore find the unique x_+ such that

$$x_+'R_i^{xu} + \underline{u}_i'R_i^{uu} + R_i^u + F_i^u\nabla^+ RUB_{i+1}(F_i^x x_+ + F_i^u\underline{u}_i + \overline{f}_i) \qquad (4.49)$$
$$\leq 0 \leq x_+'R_i^{xu} + \underline{u}_i'R_i^{uu} + R_i^u + F_i^u\nabla^- RUB_{i+1}(F_i^x x_+ + F_i^u\underline{u}_i + \overline{f}_i)$$

If no solution x_+ to (4.49) exists with $(F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u\underline{u}_i - \overline{f}_i) \leq x_+ \leq \tilde{x}_i^{\omega_i}$ because both expressions in (4.49) are positive then let $x_+ = \tilde{x}_i^{\omega_i}$. If no solution x_+ to (4.49) exists with $(F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u\underline{u}_i - \overline{f}_i) \leq x_+ \leq \tilde{x}_i^{\omega_i}$ because both expressions in (4.49) are negative then let $x_+ = (F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u\underline{u}_i - \overline{f}_i)$.

We also find the point x_- where the restriction $u_i(x_i) \leq \overline{u}_i$ changes from being active to being inactive as x_i is increased. Similarly to (4.49) above this is found by solving

$$x_-'R_i^{xu} + \overline{u}_i'R_i^{uu} + R_i^u + F_i^u\nabla^+ RUB_{i+1}(F_i^x x_- + F_i^u\overline{u}_i + \overline{f}_i) \qquad (4.50)$$
$$\leq 0 \leq x_-'R_i^{xu} + \overline{u}_i'R_i^{uu} + R_i^u + F_i^u\nabla^- RUB_{i+1}(F_i^x x_- + F_i^u\overline{u}_i + \overline{f}_i)$$

If no solution x_- to (4.50) exists with $\tilde{x}_i^1 \leq x_- \leq (F_i^x)^{-1}(\tilde{x}_{i+1}^{\omega_{i+1}} - F_i^u\overline{u}_i - \overline{f}_i)$ because both expressions in (4.50) are positive then let $x_- = (F_i^x)^{-1}(\tilde{x}_{i+1}^{\omega_{i+1}} - F_i^u\overline{u}_i - \overline{f}_i)$. If no solution x_- to (4.50) exists with $\tilde{x}_i^1 \leq x_- \leq (F_i^x)^{-1}(\tilde{x}_{i+1}^{\omega_{i+1}} - F_i^u\overline{u}_i - \overline{f}_i)$ because both expressions in (4.50) are negative then let $x_- = \tilde{x}_i^1$.

Because RUB_{i+1} is concave and piecewise quadratic the expressions for the directional derivatives are piecewise linear and monotone functions of x_i. In the sequel we shall assume that $R_i^{xu} \leq 0$, $F_i^x > 0$ and $F_i^u > 0$ in which case the expressions are decreasing with increasing x_i. Then the

solution of (4.49) - (4.50) can be found by a monotone, finite search in x_i. Moreover, it may be exploited that $x_- \leq x_+$.

We can now specify two parts of RUB_i. For $\tilde{x}_i^1 \leq x_i \leq x_-$ we have

$$\begin{aligned} &RUB_i(x_i) = \\ &\tfrac{1}{2}x_i' R_i^{xx} x_i + x_i' R_i^{xu} \overline{u}_i + \tfrac{1}{2}\overline{u}_i' R_i^{uu} \overline{u}_i + R_i^u \overline{u}_i + R_i^x x_i \\ &+RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) \end{aligned} \tag{4.51}$$

Any breakpoint $\tilde{x}_{i+1}^j$ for which $\tilde{x}_{i+1}^j \leq F_i^x x_- + F_i^u \overline{u}_i + \overline{f}_i$ implies a breakpoint $\tilde{x}_i^j = (F_i^x)^{-1}(\tilde{x}_{i+1}^j - F_i^u \overline{u}_i - \overline{f}_i)$.

For $x_+ \leq x_i \leq \tilde{x}_i^{\omega_i}$ we have

$$\begin{aligned} &RUB_i(x_i) = \\ &\tfrac{1}{2}x_i' R_i^{xx} x_i + x_i' R_i^{xu} \underline{u}_i + \tfrac{1}{2}\underline{u}_i' R_i^{uu} \underline{u}_i + R_i^u \underline{u}_i + R_i^x x_i \\ &+RUB_{i+1}(F_i^x x_i + F_i^u \underline{u}_i + \overline{f}_i) \end{aligned} \tag{4.52}$$

Any breakpoint $\tilde{x}_{i+1}^j$ for which $F_i^x x_+ + F_i^u \overline{u}_i + \overline{f}_i \leq \tilde{x}_{i+1}^j$ implies a breakpoint $\tilde{x}_i^k = (F_i^x)^{-1}(\tilde{x}_{i+1}^j - F_i^u \underline{u}_i - \overline{f}_i)$ for some appropriate index k. (The index k may be identified after we have determined x_-^j for all relevant j, see below.)

Next we find the solution $u_i(x_i)$ to (4.47) for all x_i for which $(F_i^x x_i + F_i^u u_i(x_i) + \overline{f}_i)$ is at one of the remaining breakpoints of RUB_{i+1}. Therefore let $\underline{j}$ and $\overline{j}$ be the smallest and the largest index j, respectively, for which breakpoint $\tilde{x}_{i+1}^j$ satisfies

$$F_i^x x_- + F_i^u \overline{u}_i + \overline{f}_i \leq \tilde{x}_{i+1}^j \leq F_i^x x_+ + F_i^u \underline{u}_i + \overline{f}_i \tag{4.53}$$

If $x_- = \tilde{x}_i^1$ then let $\underline{j} = 1$; if $x_+ = \tilde{x}_i^{\omega_{i+1}}$ then let $\overline{j} = \omega_{i+1}$. Now identify x_-^j and x_+^j the following way, starting with $j = \underline{j}$ and then incrementing j by one until $j = \overline{j}$.

Suppose $j < \omega_{i+1}$ (i.e. $\tilde{x}_{i+1}^j < \tilde{x}_{i+1}^{\omega_{i+1}}$) and that the unconstrained optimum $u_i(x_i)$ to (4.47) is at the piece of RUB_{i+1} immediately to the right of $\tilde{x}_{i+1}^j$. Then $u_i = u_i(x_i)$ satisfies the stationarity condition

$$\begin{aligned} &x_i' R_i^{xu} + u_i' R_i^{uu} + R_i^u \\ &+2(F_i^x x_i + F_i^u u_i + \overline{f}_i)' Q_{i+1}^j F_i^u + P_{i+1}^j F_i^u = 0 \end{aligned} \tag{4.54}$$

cf. (4.11). If we also want $F_i^x x_i + F_i^u u_i + \overline{f}_i = \tilde{x}_{i+1}^j$ then this gives us two linear equations for the two variables (x_i, u_i). We call the solution with respect to x_i for x_+^j. The relation (4.54) implies

$$u_i(x_i) = -(R_i^{uu})^{-1}(R_i^{xu\prime} x_i + R_i^{u\prime} + 2F_i^{u\prime} Q_{i+1}^j \tilde{x}_{i+1}^j + F_i^u P_{i+1}^j) \tag{4.55}$$

and therefore we find by insertion of (4.55) into (4.8) with $x_{i+1} = \tilde{x}_{i+1}^j$ that

$$\begin{aligned} &x_+^j = \\ &(F_i^x - F_i^u (R_i^{uu})^{-1} R_i^{xu\prime})^{-1} \\ &(\tilde{x}_{i+1}^j + F_i^u (R_i^{uu})^{-1}(R_i^{u\prime} + 2F_i^u Q_{i+1}^j \tilde{x}_{i+1}^j + F_i^{u\prime} P_{i+1}^j) - \overline{f}_i) \end{aligned} \tag{4.56}$$

If $x_+^j \leq x_-$ then let $x_+^j = x_-$. If $x_+^j \geq x_+$ then let $x_+^j = x_+$. For the case $j = \omega_{i+1}$ we let $x_+^j = x_+$.

Similarly if we want to find x_-^j such that the unconstrained optimum u_i lies at the piece of RUB_{i+1} immediately to the left of $\tilde{x}_{i+1}^j$ and exactly at $\tilde{x}_{i+1}^j$, where $1 < j$ (i.e. $\tilde{x}_{i+1}^1 < \tilde{x}_{i+1}^j$), then x_-^j may be verified to be given as

$$\begin{aligned} & x_-^j = \\ & (F_i^x - F_i^u(R_i^{uu})^{-1}R_i^{xu\prime})^{-1} \\ & (\tilde{x}_{i+1}^j + F_i^u(R_i^{uu})^{-1}(R_i^{u\prime} + 2F_i^u Q_{i+1}^{j-1}\tilde{x}_{i+1}^j + F_i^{u\prime}P_{i+1}^{j-1}) - \overline{f}_i) \end{aligned} \tag{4.57}$$

If $x_-^j \leq x_-$ then let $x_-^j = x_-$ (therefore it is only necessary to find x_-^j if $x_+^j > x_-$ because $x_-^j \leq x_+^j$). If $x_-^j \geq x_+$ then let $x_-^j = x_+$ and there is no need to increase j further. For the case $j = 1$ we let $x_-^j = x_-$.

We have now for all relevant j found x_-^j and x_+^j such that for $x_-^j \leq x_i \leq x_+^j$ the optimal x_{i+1} resulting from solution of (4.47) is $\tilde{x}_{i+1}^j$. This means that for $x_-^j \leq x_i \leq x_+^j$

$$u_i(x_i) = (F_i^u)^{-1}(\tilde{x}_{i+1}^j - F_i^x x_i - \overline{f}_i) \tag{4.58}$$

If RUB_{i+1} is differentiable at $\tilde{x}_{i+1}^j$ then $x_-^j = x_+^j$. In case $x_-^j < x_+^j$ we can specify RUB_i for $x_-^j \leq x_i \leq x_+^j$ using (4.58):

$$\begin{aligned} & RUB_i(x_i) = \\ & \tfrac{1}{2}x_i'R_i^{xx}x_i + (x_i'R_i^{xu} + \tfrac{1}{2}u_i(x_i)'R_i^{uu} + R_i^u)u_i(x_i) + R_i^x x_i + RUB_{i+1}(\tilde{x}_{i+1}^j) \end{aligned} \tag{4.59}$$

Finally we determine RUB_i for $x_+^j \leq x_i \leq x_-^{j+1}$. For these x_i, $\tilde{x}_{i+1}^j \leq F_i^x x_i + F_i^u u_i(x_i) + \overline{f}_i \leq \tilde{x}_{i+1}^{j+1}$. The solution to (4.47) is therefore the stationary point implied by (4.54). To determine the relevant Q_i^j, P_i^j and $u_i(x_i)$ we may use (4.16) - (4.22).

To complete the description of RUB_i in the form (4.45) we must now identify $\tilde{x}_i^j$, Q_i^j, P_i^j and ω_i (ρ_i is not necessary as explained above).

For $\tilde{x}_i^1 \leq x_i \leq x_-$, $\tilde{x}_i^j$ are determined from $F_i^x\tilde{x}_i^j + F_i^u\overline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^j$ i.e. $\tilde{x}_i^j = (F_i^x)^{-1}(\tilde{x}_{i+1}^j - F_i^u\overline{u}_i - \overline{f}_i)$. For $x_- \leq x_i \leq x_+$ the values of $\tilde{x}_i^j$ were found in connection with determination of RUB_i in (4.57) and (4.56) as x_-^j and x_+^j. It only remains to give the proper indexes j. Finally for $x_+ \leq x_i \leq \tilde{x}_i^{\omega_i}$, $\tilde{x}_i^j$ are determined from $F_i^x\tilde{x}_i^j + F_i^u\underline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^{j+k}$ i.e. $\tilde{x}_i^j = (F_i^x)^{-1}(\tilde{x}_{i+1}^{j+k} - F_i^u\underline{u}_i - \overline{f}_i)$ for appropriate indexes j and for an appropriate integer k, where $k = \omega_{i+1} - \omega_i$. ω_i is identified as the number of (different) breakpoints.

It is seen that for $\tilde{x}_i^1 \leq x_i \leq x_-$ and $x_+ \leq x_i \leq \tilde{x}_i^{\omega_i}$ the breakpoints of RUB_i correspond in a one-to-one way to the breakpoints of RUB_{i+1} and here RUB_i is non-smooth at a breakpoint if and only if RUB_{i+1} is non-smooth at the corresponding breakpoint. Each breakpoint $\tilde{x}_{i+1}^j$ of RUB_{i+1} between $(F_i^x x_- + F_i^u\overline{u}_i + \overline{f}_i)$ and $(F_i^x x_+ + F_i^u\underline{u}_i + \overline{f}_i)$ generates one or two breakpoints of RUB_i, viz., x_-^j and x_+^j, for some j; if RUB_{i+1} is smooth at $\tilde{x}_{i+1}^j$ then $x_-^j = x_+^j$, i.e., only one breakpoint is generated, otherwise two. The points x_- and x_+ are two additional breakpoints at RUB_i (they may coincide with a x_-^j, a x_+^j, $\tilde{x}_i^1$ or $\tilde{x}_i^{\omega_i}$). Here RUB_i is smooth if $F_i^x x_- + F_i^u\overline{u}_i + \overline{f}_i$ (or $F_i^x x_+ + F_i^u\underline{u}_i + \overline{f}_i$) does not coincide with a breakpoint of RUB_{i+1}, or if it coincides with a point where RUB_{i+1} is smooth. Otherwise RUB_i is non-smooth at x_- (or x_+, respectively). In conclusion we see that the number ω_i of breakpoints of RUB_i is at most $2(N+1-i)$ (if $\underline{x}_N = \overline{x}_N$ at most $2(N+1-i)-1$).

The values for Q_i^j and P_i^j are identified from (4.51), (4.52), (4.59) and (4.16) - (4.24) as follows:

$$Q_i^j = \left\{ \begin{array}{l} \frac{1}{2}R_i^{xx} + (F_i^x)^2 Q_{i+1}^k \\ \frac{1}{2}R_i^{xx} - R_i^{xu}(F_i^u)^{-1}F_i^x + \frac{1}{2}R_i^{uu}(F_i^u)^{-2}(F_i^x)^2 \\ \text{see (4.23)} \\ \frac{1}{2}R_i^{xx} + (F_i^x)^2 Q_{i+1}^k \end{array} \right\} \ldots$$

$$\ldots \left\{ \begin{array}{ll} \text{for} & \tilde{x}_i^1 \le x_i \le x_- \\ \text{for} & x_-^j \le x_i \le x_+^j \\ \text{for} & x_+^j \le x_i \le x_-^{j+1} \\ \text{for} & x_+ \le x_i \le \tilde{x}_i^{\omega_i} \end{array} \right. \tag{4.60}$$

$$P_i^j = \left\{ \begin{array}{l} R_i^{xu}\overline{u}_i + R_i^x + 2F_i^x(F_i^u\overline{u}_i + \overline{f}_i)Q_{i+1}^k + F_i^x P_{i+1}^k R_i^{xu}(F_i^u)^{-1}(\tilde{x}_{i+1}^j - \overline{f}_i) \\ -F_i^x(F_i^u)^{-2}R_i^{uu}(\tilde{x}_{i+1}^j - \overline{f}_i) + R_i^{xu}(F_i^u)^{-1}(\tilde{x}_{i+1}^j - \overline{f}_i) - R_i^u(F_i^u)^{-1}F_i^x + R_i^x \\ \text{see (4.24)} \\ R_i^{xu}\underline{u}_i + R_i^x + 2F_i^x(F_i^u\underline{u}_i + \overline{f}_i)Q_{i+1}^k + F_i^x P_{i+1}^k \end{array} \right\} \ldots$$

$$\ldots \left\{ \begin{array}{ll} \text{for} & \tilde{x}_i^1 \le x_i \le x_- \\ \text{for} & x_-^j \le x_i \le x_+^j \\ \text{for} & x_+^j \le x_i \le x_-^{j+1} \\ \text{for} & x_+ \le x_i \le \tilde{x}_i^{\omega_i} \end{array} \right. \tag{4.61}$$

The indexes j and k are identified as discussed above in connection with $\tilde{x}_i^j$. As noted, the values ρ_i^j are not needed for determination of the optimal strategy and trajectory.

In summary we therefore see that this QLI problem may be solved by one backwards and one forwards recursion. We have $\omega_N = 2$, $\tilde{x}_N^1 = \underline{x}_N$, $\tilde{x}_N^2 = \overline{x}_N$, $Q_N = \frac{1}{2}R_N^{xx}$ and $P_N = R_N^x$ (if $\underline{x}_N = \overline{x}_N$ then $\omega_N = 1$, $\tilde{x}_N^1 = \underline{x}_N = \overline{x}_N$ and R_N^{xx}, R_N^x, Q_N and P_N are not needed). In the backwards recursion at each stage $\tilde{x}_i^j$ are determined and then RUB_i are constructed according to (4.60) - (4.61). The corresponding feedback control $u_i(x_i)$ which was identified above can be summarized as

$$u_i(x_i) = \left\{ \begin{array}{ll} \overline{u}_i & \text{for } \tilde{x}_i^1 \le x_i \le x_- \\ \text{see (4.58)} & \text{for } x_-^j \le x_i \le x_+^j \\ \text{see (4.17)-(4.22), (4.30)} & \text{for } x_+^j \le x_i \le x_-^{j+1} \\ \underline{u}_i & \text{for } x_+ \le x_i \le \tilde{x}_i^{\omega_i} \end{array} \right. \tag{4.62}$$

In the forwards recursion we start with $\underline{x}_0$ and then find the optimal strategy and trajectory from (4.62) and the dynamic equation.

It is seen that the number of arithmetic operations is proportional to the number of breakpoints. The number of breakpoints increases at most linearly with the number of stages. It follows that the computational complexity of the algorithm is $O(N^2)$.

Proposition 4.4.1 *Assume the problem formulation (4.7) - (4.9), (4.44) with $n = 1$, $m = 1$, (4.44) given as $\underline{u}_i \le u_i \le \overline{u}_i$ and $\underline{x}_i \le x_i \le \overline{x}_i$, $R_i^{xx} \le 0$, $R_i^{uu} < 0$, $R_i^{xx}R_i^{uu} - (R_i^{xu})^2 \le 0$, $F_i^x > 0$, $F_i^u > 0$ and $R_N^{xx} < 0$. Then RUB_i and the unique optimal solution can be found as described above. The computational complexity of the algorithm is $O(N^2)$.*

Proof. The argumentation was given above. □

Also the case with $F_i^u < 0$ can be handled. The simplest way is to change the sign of F_i^u, R_i^{xu} and R_i^u, replace $\underline{u}_i \leq u_i \leq \overline{u}_i$ by $-\overline{u}_i \leq u_i \leq -\underline{u}_i$, solve the problem as described above and finally change the sign of the u^* found.

If some or all of the constraints $\underline{x}_i \leq x_i$, $x_i \leq \overline{x}_i$, $\underline{u}_i \leq u_i$ or $u_i \leq \overline{u}_i$ are missing this situation can be handled for instance by formally letting $\underline{x}_i = -\infty$, $\overline{x}_i = \infty$, $\underline{u}_i = -\infty$ or $\overline{u}_i = \infty$, respectively.

If x_0 is free (or possibly constrained to an interval) then the optimal value x_0^* is found as in (4.34) (possibly constrained to the interval) after completion of the backwards but before the start of the forwards recursion.

Similar ideas as above may be applied if the problem formulation is slightly expanded, but still with $n = 1$, cf. Section 4.6.

Implementation aspects, including approximation in order to reduce the number of breakpoints in the representation of RUB_i, are discussed in Jørgensen (1993), Jørgensen and Ravn (1997).

4.5 The Linear Problem with n=1

Consider now the linear problem

$$\max[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N] \tag{4.63}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{4.64}$$

$$G_i^x x_i + G_i^u u_i - \overline{g}_i \leq 0 \tag{4.65}$$

$$H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \tag{4.66}$$

$$x_0 = \underline{x}_0 \tag{4.67}$$

This problem can also be solved by dynamic programming. Y_i and RY_i are polyhedral and the greater upper boundaries are piecewise linear and concave, cf. Proposition 2.4.5.

As in the case with the QLI it is, however, difficult to specify them explicitly, except when $n = 1$.

We shall now describe an algorithm based on backwards dynamic programming for the case with $n = 1$, $m = 1$, $\ell = 0$ (i.e., no constraints (4.66)) and the inequality constraints (4.65) given as $\underline{u}_i \leq u_i \leq \overline{u}_i$ and $\underline{x}_i \leq x_i \leq \overline{x}_i$. Bannister and Kaye (1991) derived a similar algorithm.

As RUB_i is piecewise linear it may be described by breakpoints $\tilde{x}_i^j$ and parameters P_i^j and ρ_i^j such that $RY_i = \{x_i \mid \tilde{x}_i^1 \leq x_i \leq \tilde{x}_i^{\omega_i}\}$ and for $\tilde{x}_i^j \leq x_i \leq \tilde{x}_i^{j+1}$, $j = 1, \ldots, \omega_i - 1$,

$$RUB_i(x_i) = P_i^j x_i + \rho_i^j \tag{4.68}$$

As in the QLI case RUB_i is continuous. Therefore only one ρ_i^j is needed at each stage and as only the slopes are used in the sequel, all ρ_i^j may be omitted.

Let RUB_{i+1} be given is this form. We now specify how to find RUB_i.

The maximizing $u_i = u_i(x_i)$ of the backwards dynamic programming criterion at stage i,

$$R_i^x x_i + R_i^u u_i + RUB_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) \tag{4.69}$$

subject to $\underline{u}_i \leq u_i \leq \overline{u}_i$ will be a u_i satisfying

$$\left\{\begin{array}{lll} u_i = \overline{u}_i & \text{if} & R_i^u + F_i^u \nabla^- RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) > 0 \\ \underline{u}_i \leq u_i \leq \overline{u}_i & \text{if} & R_i^u + F_i^u \nabla^+ RUB_{i+1}(F_i^x x_i + F_i^u \underline{u}_i + \overline{f}_i) \\ & & \leq 0 \leq R_i^u + F_i^u \nabla^- RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) \\ u_i = \underline{u}_i & \text{if} & R_i^u + F_i^u \nabla^+ RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) < 0 \end{array}\right. \tag{4.70}$$

where ∇^+ and ∇^- are the directional derivatives to the right and to the left, respectively.

We can now identify breakpoints as in the QLI case. The points $\tilde{x}_1^1$ and $\tilde{x}_i^{\omega_i}$ are defined as in the QLI case. Assuming $F_i^x > 0$ and $F_i^u > 0$ we have $\tilde{x}_i^1 = \max\{\underline{x}_i, (F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u \overline{u}_i - \overline{f}_i)\}$ and $\tilde{x}_i^{\omega_i} = \min\{\overline{x}_i, (F_i^x)^{-1}(\tilde{x}_{i+1}^1 - F_i^u \underline{u}_i - \overline{f}_i)\}$.

Then define x_- as follows. We assume that $F_i^u > 0$ and find x_- as the smallest x_i such that

$$\begin{aligned} R_i^u + F_i^u \nabla^+ RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) \leq 0 \leq \\ R_i^u + F_i^u \nabla^- RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) \end{aligned} \tag{4.71}$$

This corresponds to finding the smallest index j such that

$$R_i^u + F_i^u P_{i+1}^j \leq 0 \leq R_i^u + F_i^u P_{i+1}^{j-1} \tag{4.72}$$

and then finding x_- from $F_i^x x_- + F_i^u \overline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^j$, i.e. $x_- = (F_i^x)^{-1}(\tilde{x}_{i+1}^j - F_i^u \overline{u}_i - \overline{f}_i)$.

If no solution exists for x_- because for all j the expressions in (4.72) are positive then let $x_- = (F_i^x)^{-1}(\chi_{i+1}^{\omega_{i+1}} - F_i^u \overline{u}_i - \overline{f}_i)$. If no solution exists for x_- because for all j the expressions in (4.72) are negative then let $x_- = (F_i^x)^{-1}(\chi_{i+1}^1 - F_i^u \underline{u}_i - \overline{f}_i)$.

Then define x_+ as follows

$$x_+ = \left\{\begin{array}{l} x_- + (F_i^u)^{-1}(\overline{u}_i - \underline{u}_i) \\ x_- + (F_i^u)^{-1}(\overline{u}_i - \underline{u}_i) + (F_i^x)^{-1}(\tilde{x}_{i+1}^{j+1} - \tilde{x}_{i+1}^j) \end{array}\right\} \cdots$$

$$\cdots \left\{\begin{array}{ll} \text{if} & R_i^u + F_i^u P_{i+1}^j < 0 \\ \text{if} & R_i^u + F_i^u P_{i+1}^j = 0 \end{array}\right. \tag{4.73}$$

In the latter case there is a non-unique solution for $u_i(x_i)$.

Now RUB_i is defined as follows:

$$RUB_i(x_i) = \left\{\begin{array}{l} R_i^x x_i + R_i^u \overline{u}_i + RUB_{i+1}(F_i^x x_i + F_i^u \overline{u}_i + \overline{f}_i) \\ R_i^x x_i + R_i^u u_i + RUB_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) \\ R_i^x x_i + R_i^u \underline{u}_i + RUB_{i+1}(F_i^x x_i + F_i^u \underline{u}_i + \overline{f}_i) \end{array}\right\} \cdots$$

$$\cdots \left\{\begin{array}{l} \text{for } \tilde{x}_i^1 \leq x_i \leq x_- \\ \text{for } x_- \leq x_i \leq x_+ \\ \text{for } x_+ \leq x_i \leq \tilde{x}_i^{\omega_i} \end{array}\right. \tag{4.74}$$

and the optimal feedback control $u_i(x_i)$ is

$$u_i(x_i) = \left\{\begin{array}{lll} \overline{u}_i & \text{for} & \tilde{x}_i^1 \leq x_i \leq x_- \\ u_i & \text{for} & x_- \leq x_i \leq x_+ \\ \underline{u}_i & \text{for} & x_+ \leq x_i \leq \tilde{x}_i^{\omega_i} \end{array}\right. \tag{4.75}$$

In the middle option of (4.74) and (4.75) $u_i = (F_i^u)^{-1}(\tilde{x}_i^j - F_i^x x_i - \overline{f}_i)$ if the first case in (4.73) applies and u_i is any value such that $\tilde{x}_{i+1}^j \leq F_i^x x_i + F_i^u u_i + \overline{f}_i \leq \tilde{x}_{i+1}^{j+1}$ if the second case applies.

To complete the description of RUB_i in the form (4.68) we need identify $\tilde{x}_i^j$, P_i^j and ω_i. This can be done immediately from the above specification. For $\tilde{x}_i^1 \leq x_i \leq x_-$, $\tilde{x}_i^j$ are determined from $F_i^x \tilde{x}_i^j + F_i^u \overline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^j$, i.e. $\tilde{x}_i^j = (F_i^x)^{-1}(\tilde{x}_{i+1}^j - F_i^u \overline{u}_i - \overline{f}_i)$. The next breakpoint is x_+. For $x_+ \leq x_i \leq \tilde{x}_i^{\omega_i}$ the final breakpoints are determined from $F_i^x \tilde{x}_i^j + F_i^u \underline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^{j+1}$ or $F_i^x \tilde{x}_i^j + F_i^u \underline{u}_i + \overline{f}_i = \tilde{x}_{i+1}^j$, according whether the first or the second specification of x_+ in (4.73) applies.

In contrast to the QLI case the number of breakpoints increases from stage $(i+1)$ to stage i by only zero or one, according to whether the second or the first specification, respectively, applies in (4.73), and the maximum number of breakpoints at stage i is therefore $(N+2-i)$.

Finally P_i^j can be read off (4.74); the middle option gives $P_i^j = R_i^x - R_i^u (F_i^u)^{-1} F_i^x$.

In summary we therefore see that this problem may be solved by one backwards and one forwards recursion. We have $\omega_N = 2$, $\tilde{x}_N^1 = \underline{x}_N$, $\tilde{x}_N^2 = \overline{x}_N$ and from (4.63) $P_N = R_N^x$ (if $\underline{x}_N = \overline{x}_N$ then $\omega_N = 1$, $\tilde{x}_N^1 = \underline{x}_N = \overline{x}_N$ and R_N^x and P_N are not needed). In the backwards recursion RUB_i are constructed according to (4.74). In the forwards direction we start with $\underline{x}_0$ and then find the optimal strategy and trajectory from (4.75) and the dynamic equation.

It is noteworthy that the solution procedure provides the complete set of solutions, also in the case where the solution is non-unique (c.f. e.g. (4.75)). The complications in this case are somewhat similar to those in relation to application of the maximum principle to singular problems.

If x_0 is free (or possibly constrained to an interval) then the optimal value x_0^* is found as in (4.34) (possibly constrained to the interval) after completion of the backwards but before the start of the forwards recursion.

It is seen that the number of arithmetic operations is proportional to the number of breakpoints. The number of breakpoints increases at most linearly with the number of stages treated. It follows that the computational complexity of the algorithm is $O(N^2)$.

Proposition 4.5.1 *Assume the problem (4.63) - (4.65), (4.67) with $n = 1$, $m = 1$, (4.65) given as $\underline{u}_i \leq u_i \leq \overline{u}_i$ and $\underline{x}_i \leq x_i \leq \overline{x}_i$, $F_i^u > 0$ and $F_i^x > 0$. Then RUB_i and the optimal solution can be found as described above. The computational complexity of the algorithm is $O(N^2)$.*

Proof. The argumentation was given above. □

If $F_i^u < 0$ then the problem may be solve by changing the sign of F_i^u and R_i^u, replacing $\underline{u}_i \leq u_i \leq \overline{u}_i$ by $-\overline{u}_i \leq u_i \leq -\underline{u}_i$, solving the problem as described above and finally changing the sign of the u^* found. As in the QLI case the ideas used may be applied to a problem which is somewhat expanded, provided $n = 1$.

4.6 Other Problems with n=1

Similar ideas as above may be applied if the problem formulation is slightly expanded, but still with $n = 1$. If $m \geq 1$, $k \geq 0$ and/or $\ell \geq 0$ it may be possible to give an explicit solution to (4.47) and therefore also an explicit description (4.45) of RUB_i along the lines used above.

This in particular may be the case if there are no inequality constraints ($k = 0$). Thus, simple solutions were found in problems with logarithmic and exponential functions in the context of optimal rate of resource exploitation (Beckmann(1974)) and in the context of adiabatic compression of a gas (Aris, Bellman and Kalaba (1960)).

If r_i is only concave but not necessarily strictly concave with respect to u_i, i.e. $R_i^{uu} \leq 0$, $m = 1$, the algorithm may be extended by combining the ideas given below for quadratic and linear problems. This may for instance be applied to problems with a criterion function consisting of piecewise linear and quadratic segments, as in the 'dead-zone' model of Parlar and Vickson (1980), Parlar (1982).

If $m > 1$ it may be possible to reformulate the problem to an equivalent one with $m = 1$ as follows. First identify all ub_i^{i+1}. Then define a new control $z_i \in R$, and reformulate the problem as

$$\max[\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, z_i)] \tag{4.76}$$

$$x_{i+1} = z_i \tag{4.77}$$

$$z_i \in U_i(x_i) \cap RY_{i+1} \tag{4.78}$$

Here, $U_i(x_i)$ is defined such that $z_i \in U_i(x_i)$ if and only if $(x_i, z_i) \in W_i^{i+1}$.

The decisive element is, of course, if ub_i^{i+1} is easily represented and analyzed. Whether this is the case is problem specific. If it is indeed the case, we may use the following.

Proposition 4.6.1 *Consider a problem with $n = 1$ and assume that (1) $ub_i^{i+1}(x_i, .)$ for any x_i is a continuous, piecewise explicitly given function, continuously differentiable on each piece, with at most K breakpoints, (2) the position of the breakpoints on $ub_i^{i+1}(x_i, .)$ are independent of x_i, (3) $ub_i^{i+1}(., z_i)$ for any z_i is a continuous, piecewise explicitly given function, continuously differentiable on each piece, with at most K breakpoints, and (4) the position of the breakpoints on $ub_i^{i+1}(., z_i)$ are independent of z_i.*

Assume further that the DP backwards recursion is solvable, such that $z_i(x_i)$ is a continuous, piecewise explicitly given function, continuously differentiable on each piece, and either non-decreasing or non-increasing. Assume that $[ub_i^{i+1}(x_i, z_i(x_i)) + RUB_{i+1}(z_i(x_i))]$ is a continuous, piecewise explicitly given function, continuously differentiable on each piece if RUB_{i+1} is so. Assume that the computational effort for solving the DP recursion at any stage is proportional to the sum of the number of breakpoints on ub_i^{i+1} and on RUB_{i+1}.

Then the problem may be solved by DP and the computational complexity of the algorithm is $O(KN^2)$.

Proof. Consider the movement of x_i from lower to upper bound of the interval RY_i. Here, at most K breakpoint of ub_i^{i+1} will be encountered by x_i. As x_i moves, $z_i^*(x_i)$ moves, either non-decreasingly or non-increasingly, and $z_i^*(x_i)$ will encounter at most K breakpoint of ub_i^{i+1}. In addition, $z_i^*(x_i)$ will encounter all or a part of the breakpoint on RUB_{i+1}. As seen in the argumentation in relation to Proposition 4.4.1, a non-smooth breakpoint on RUB_{i+1} results in either one non-smooth breakpoint or two smooth breakpoints on RUB_i. A smooth breakpoint on RUB_{i+1} results in one smooth breakpoint on RUB_i. Thus, for each stage, the breakpoints on ub_i^{i+1} will result in at most $4K$ breakpoints on any RUB_j, $j < i$. Formally, RUB_N may be defined as $RUB_N(x_N) = 0$, and therefore the number of breakpoints on RUB_N is zero. Consequently, the number of breakpoints on RUB_i is at most $4K(N-i)+K$. N stages have to be treated, and the complexity of the treatment of each stage is proportional to the number of breakpoints. Consequently the computational complexity of the algorithm is $O(KN^2)$. □

The previously treated cases of quadratic-linear problems satisfy the assumptions of the Proposition. The assumption of non-decreasing or non-increasing z_i may be illuminated by reference to

the concept of supermodularity (see Section 9.2 for more on this). We say that $g : R^2 \rightarrow R$ is supermodular if

$$g(x_1, y_1) + g(x_2, y_2) \geq g(x_1, y_2) + g(x_2, y_1) \tag{4.79}$$

for all $x_1 > x_2$, $y_1 > y_2$.

If g is twice continuously differentiable then this is tantamount to

$$\nabla^2_{xy} g \geq 0 \tag{4.80}$$

An important consequence of submodularity is that

$$\arg\max_y [g(x, y)] \tag{4.81}$$

is nondecreasing with x. Optimization problems of the form $\max[\sum_{i=1}^{N-1} g(x_i, x_{i+1})]$ display a nice solution structure linked to monotonicity conditions.

Our optimization problem is $\max[\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})]$. These solution structures will therefore be satisfied for our problem if all ub_i^{i+1} are identical and ub_i^{i+1} supermodular. This is fulfilled if all ub_i^{i+1} are identical and if ub_i^{i+1} satisfies (4.79) or (4.80). As seen, our assumption in Proposition 4.6.1 are less strict than required for the supermodularity approach in the sense that we do not require the criterion functions at all stages to be identical.

4.7 Forwards DP on the QLE Problem

The previous results were derived using DP in the backwards direction. However, also the forwards direction may be used. We shall indicate how this may be done for the QLE problem, and observe the computational differences.

The basis is that we may represent UB_i as a quadratic, strictly concave function

$$x_i' Q_i x_i + P_i x_i + \rho_i \tag{4.82}$$

where Q_i is an $n \times n$ matrix, P_i is an $1 \times n$ matrix and ρ_i is a constant. As before, the constant ρ_i is not important and will be disregarded. Then UB_{i+1} is determined recursively forwards as

$$\begin{aligned} UB_{i+1}(x_{i+1}) = \quad & \max_{x_i, u_i} [\tfrac{1}{2} x_i' R_i^{xx} x_i + x_i' R_i^{xu} u_i + \tfrac{1}{2} u_i' R_i^{uu} u_i \qquad (4.83) \\ & + R_i^x x_i + R_i^u u_i + x_i' Q_i x_i + P_i x_i] \\ & x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \qquad (4.84) \\ & H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \qquad (4.85) \end{aligned}$$

Similarly to the development in Section 2.8 we define matrices $M$and N as follows:

$$M = \begin{pmatrix} -R_i^{xx} & -R_i^{xu} & F_i^{x\prime} & H_i^{x\prime} \\ -R_i^{ux} & -R_i^{uu} & F_i^{u\prime} & H_i^{u\prime} \\ F_i^x & F_i^u & 0 & 0 \\ H_i^x & H_i^u & 0 & 0 \end{pmatrix} \tag{4.86}$$

$$N = \begin{pmatrix} 0 \\ 0 \\ I \\ 0 \end{pmatrix} \tag{4.87}$$

Here, M is of dimension $(2n+m+\ell) \times (2n+m+\ell)$, N is of dimensions $(2n+m+\ell) \times n$, 0 are zero matrices of suitable dimensions and I is the $n \times n$ identity matrix.

The solution $(x_i^*, u_i^*, p_{i+1}^*, \mu_i^*)$ is given as a function of x_{i+1}:

$$\begin{pmatrix} x_i^*(x_{i+1}) \\ u_i^*(x_{i+1}) \\ p_{i+1}^*(x_{i+1}) \\ \mu_i^*(x_{i+1}) \end{pmatrix} = M^{-1} \begin{pmatrix} R_i^x + P_i \\ R_i^u \\ -\overline{f}_i \\ \overline{h}_i \end{pmatrix} + M^{-1} N x_{i+1} \tag{4.88}$$

Inserting $(x_i(x_{i+1}), u_i(x_{i+1}))$ from (4.88) into [ ] in (4.83) we obtain UB_{i+1} as a quadratic function. This continues recursively with the next stage, until UB_N has been found. Then the optimal (x_i, u_{i-1}) are constructed recursively backwards.

In the backwards as well as in the forwards direction the computational effort is proportional to N. The main difference therefore is in relation to the determination of M^{-1}. This has computational efforts which in the forwards direction are proportional to $(2n+m+\ell)^3$ and in the backwards direction proportional to $(m+\ell)^3$.

In Section 9.4 we derive a forwards maximum principle algorithm with computational complexity of $O(N(m+n+\ell)^3)$, i.e., better than forwards DP but worse than backwards DP.

It is seen that for this reason the backwards direction DP should be preferred. Other specific reasons may call for the application of the forwards direction, cf. the Introduction to this chapter, and in such cases the forwards maximum principle should be preferred.

We summarize as follows

Proposition 4.7.1 *Assume for the QLE problem that the criterion function is concave and that the matrix M in (4.86) is nonsingular for all i. Then the forwards DP algorithm solves the problem. The algorithm has a computational complexity of $O(N(2n+m+\ell)^3)$.*

Proof. The argumentation was given above. □

4.8 Conclusions

In this chapter we have discussed aspects of implementation of dynamic programming. The emphasis has been on applications where the upper boundaries are defined on a continuous state space. This is in part motivated by the results of Section 4.1. Here it was shown that an attempt to apply dynamic programming by discretization of the state space is a fairly bad idea in the sense that convergence of the optimal criterion value of the approximated problem towards the true optimal criterion value may be expected to be relatively slow.

The remaining parts of the chapter has dealt with problems that are concave and where an explicit representation of the upper boundaries and the feedback strategies is possible.

The quadratic-linear problem has a priority position, both due to its large applicability and due to the fact that it is ideally suited for analytical treatment by DP. This was presented in Section 4.2 for the locally unconstrained problem QL and in Section 4.3 and Section 4.7 linear local equality constraints were treated as well (the QLE problem). The basic working mechanisms of DP on the QL or QLE problems are that RUB_i and UB_i are quadratic functions and the feedback strategy $u_i(.)$ is linear such that matrices may be used to represent the functions. The basic calculations are solution of linear systems of equations and matrix multiplications. This is closely related to

solution of the Riccati equation. The computational complexity is $O(N(m+\ell)^3)$ for backwards DP and $O(N(2n+m+\ell)^3)$ for forwards DP.

When there are local inequality constraints (the QLI or QLEI problem) the difficulties arise. This is attributable to the fact that now RUB_i and UB_i are piecewise quadratic functions and the feedback strategy $u_i(.)$ is piecewise linear. We showed in Section 4.4 that it is possible to solve also in this case, provided $n = 1$ and $m = 1$ and we developed an algorithm for this with computational complexity $O(N^2)$. The linear problem which traditionally poses difficulties for the stagewise approach was in Section 4.5 shown to be also solvable by DP. In Section 4.6 this was generalized, although it is still required that $n = 1$.

In conclusion we see that application of dynamic programming with exact representation of the upper boundaries is attractive from a computational point of view, and should be adopted whenever the problem formulation permits this.

Chapter 5

Smaller Upper Boundaries

In this chapter we consider application of smaller upper boundaries ub_i^{i+1} in algorithms. Here $ub_i^{i+1}(x_i, x_{i+1})$ must be determined as optimal value in the problem

$$ub_i^{i+1}(x_i, x_{i+1}) = \max_{u_i}[r_i(x_i, u_i)] \tag{5.1}$$

$$u_i \in U_i(x_i) \tag{5.2}$$

$$f_i(x_i, u_i) = x_{i+1} \tag{5.3}$$

which (for $i = 0, \ldots, N-2$; see (2.19) for $i = N-1$) is derived from the OCP

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{5.4}$$

$$x_{i+1} = f_i(x_i, u_i),\ i = 0, \ldots, N-1 \tag{5.5}$$

$$(x_i, u_i) \in V_i,\ i = 0, \ldots, N-1 \tag{5.6}$$

$$x_N \in V_N \tag{5.7}$$

By Proposition 3.3.5 we see that x^* is optimal if and only if it maximizes the function $F : R^{n(N+1)} \to R$, defined as

$$F(x) = \sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1}) \tag{5.8}$$

The set of states for which the OCP has feasible solutions is $W \subseteq R^{(N+1)n}$.

The problem (5.4) - (5.7) may therefore be written as

$$\max[F(x)] \tag{5.9}$$

$$x \in W \tag{5.10}$$

In the perspective of decomposition, the application of smaller upper boundaries falls into the class of resource decomposition or primal decomposition. It shares with other methods within this class the advantage of maintaining feasible solutions throughout the iterations, once a feasible solution has been found. The methods were advocated for the OCP in e.g. Howson and Sancho

(1975), Chang, Chang and Luh (1989) and Zuo (1991). In more general terms this may be referred to as time axis decomposition, cf. Edmunds and Bard (1990).

In Chapter 4 we discussed the solution strategy based on complete determination of the greater upper boundaries either forwards or backwards, viz., dynamic programming. As seen, this is applicable essentially only if the state space is discrete and of small size, or if the upper boundaries can be given a simple analytical expression.

In the spirit of dynamic programming we might give a complete determination of all ub_i^{i+1} if the state space is discrete and finite. Even if this could be done relatively fast (e.g. on parallel processors) the difficulties of solving the combinatorial problem of *simultaneous* optimization with respect to all x_i - i.e., the problem (5.9) - (5.10) - would certainly be prohibitive in most applications of modest to large scale. Only in certain cases, notably the quadratic-linear case, the functions ub_i^{i+1} are of such simple structure that it may be possible to work directly with them.

We shall therefore here mainly consider iterative methods.

In Section 5.1 we derive the gradient ∇F in terms of ∇ub_i^{i+1} in order to specify methods like steepest ascent. We also review in Section 5.1 two popular methods, the Progressive Optimality Principle and the State Increment Dynamic Programming. We show that convergence will occur under assumptions that imply continuous differentiability of F, i.e. of all ub_i^{i+1}. The rate of convergence will be worse than that of steepest ascent, which is not attractive.

In Section 5.2 we discuss the quadratic-linear problem, and the application of Newton like methods to non QL problems in order to achieve superlinear rate of convergence.

As noted in Example 3.3.1 stagewise optimality conditions are too weak to assure global optimality unless all ub_i^{i+1} are continuously differentiable. We discuss at length in Section 5.3 the difficulties involved when these assumptions are not fulfilled; this is typically the case if there are local inequality constraints implicit in (5.6) - (5.7). It seems difficult to take advantage of smaller upper boundaries in algorithms if it cannot be guaranteed that they are continuously differentiable.

The potential advantages of using smaller upper boundaries therefore lie in the cases where they are once or twice continuously differentiable, in which case application of parallel processing may take place. For this to be efficient it should be relatively time consuming to solve the local problems (5.1) - (5.3) relative to the global problem (5.9) - (5.10). This may be the case if the control dimension m is large relative to the state dimension n.

It should be observed that the discussion in the present chapter also applies to the case where several stages have been joined in the definition of (5.1) - (5.3). I.e., the problem (5.4) - (5.7) is partitioned into fewer than N stages. Thus, let $0 = I_0 < I_1 < \ldots < I_j < \ldots < I_K = N$, and then (5.1) - (5.3) may be replaced by

$$ub_j^{j+1}(x_{I_j}, x_{I_{j+1}}) = \max_{\{u_i\}} \Big[\sum_{i=I_j}^{I_{j+1}-1} r_i(x_i, u_i) \Big] \tag{5.11}$$

$$x_{i+1} = f_i(x_i, u_i),\ i = I_j, \ldots, I_{j+1} - 1 \tag{5.12}$$

$$(x_i, u_i) \in V_i,\ i = I_j, \ldots, I_{j+1} - 1 \tag{5.13}$$

$$x_{I_j} \text{ and } x_{I_{j+1}} \text{ given} \tag{5.14}$$

with obvious modifications for ub_{K-1}^K.

5.1 Gradient and Related Algorithms

Under the assumption that F in (5.9) is continuously differentiable, one may apply a gradient method such as steepest ascent.

Assume the local problem (5.1) - (5.3) given on the form

$$ub_i^{i+1}(x_i, x_{i+1}) = \max_{u_i}[r_i(x_i, u_i)] \tag{5.15}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{5.16}$$

$$h_i(x_i, u_i) = 0 \tag{5.17}$$

As seen in Proposition 2.7.4 it is under suitable assumptions possible to identify the gradients $\nabla_{(x_i x_{i+1})} ub_i^{i+1}(x_i, x_{i+1})$ from the Lagrange multipliers (p_{i+1}, μ_i) relative to the constraints, such that

$$\nabla_{x_i} ub_i^{i+1}(x_i, x_{i+1}) = \nabla_{x_i} H_i \tag{5.18}$$

$$\nabla_{x_{i+1}} ub_i^{i+1}(x_i, x_{i+1}) = -p_{i+1} \tag{5.19}$$

Therefore we find

$$\nabla_{x_i} F(x) = \nabla_{x_i} H_i - p_i \tag{5.20}$$

This leads immediately to the following result

Proposition 5.1.1 *Assume for the problems (5.15) - (5.17) that they are LMR regular, that all functions are continuously differentiable and that the solutions u_i and multipliers μ_i are unique. Then F is continuously differentiable and given as*

$$\nabla F(x) = (\nabla_{x_0} H_0, \nabla_{x_1} H_1 - p_1, \ldots, \nabla_{x_{N-1}} H_{N-1} - p_{N-1}, -p_N)$$

Proof. See Proposition 2.7.4. □

From this we may apply any gradient type algorithm, in particular the steepest ascent algorithm, see e.g. Luenberger (1989) p. 214. Under assumptions guaranteeing twice continuous differentiability also methods like quasi Newton or conjugate gradient methods may be applied, see Luenberger (1989).

Alternatives to the gradient type algorithms have received some attention under the names of the Progressive Optimality Principle and State Increment Dynamic Programming.

The Progressive Optimality Algorithm

In the progressive optimality algorithm, Howson and Sancho (1975), we solve the problem (5.9) - (5.10) by a coordinate, i.e. stagewise, search in x_i, with x_j kept fixed, $j \neq i$. For convenience of exposition we assume in the sequel a problem with fixed initial and end points.

We can then describe one iteration in the algorithm as follows. Move x_0 such that $[ub_0^1(x_0, x_1)]$ is maximized. Move x_1 such that $[ub_0^1(x_0, x_1) + ub_1^2(x_1, x_2)]$ is maximized. Then move x_2 in such a way that $[ub_1^2(x_1, x_2) + ub_2^3(x_2, x_3)]$ is maximized. Continue this way for all stages up to $i = N-1$. Then finally move x_N in such a way that $[ub_{N-1}^N(x_{N-1}, x_N)]$ is maximized. Then the next iteration starts again with x_0.

The algorithm terminates if no improvement is attained in the criterion in an iteration.

Proposition 5.1.2 *Consider the OCP where all functions are assumed continuously differentiable. Let a feasible initial trajectory be given. Assume that in all iterations for all i the u_i^* involved in the determination of ub_i^{i+1} is unique and that the corresponding KKT multipliers are unique. Then either the algorithm stops at a point satisfying the weak maximum principle, or any accumulation point satisfies this.*

Proof. The assumption of continuous differentiability, and uniqueness of u_i^* and the corresponding $(p_{i+1}, \lambda_i, \mu_i)$ assure that ub_i^{i+1} is continuously differentiable for all i, and therefore also F is so. If the algorithm stops then $\partial F/\partial x_i = 0$, since otherwise we could get an improvement at stage i. By the continuous differentiability of F we have a stationary point of F if and only if all partial derivatives vanish. Now assume that the algorithm produces an infinite sequence. If a point $x^\diamond$ is not stationary then we will get an improvement of F as show above. Due to the continuous differentiability of F, this also holds for any x with $\|x - x^\diamond\| < \epsilon$, for an $\epsilon > 0$. The second result therefore follows from Polak (1971) p. 14. By Proposition 3.3.6 the stationary point of F satisfies the weak maximum principle. □

As an illustration of the implication of the violation of the assumptions in Proposition 5.1.2 we give the following two examples.

Example 5.1.1 *Define $f : R^2 \to R$ as $f(x) = min\{x_1, x_2\}$. Then f is continuous and concave, but not differentiable. Consider the point $x^\diamond = (0,0)'$. If we keep x_2 fixed at $x_2^\diamond$ we see that we cannot increase f by increasing x_1; and we decrease f by decreasing x_1. Similarly if we keep x_1 fixed at $x_1^\diamond$ we cannot increase f by changing x_2. It is therefore tempting to conclude that we have a local maximum. This is not true, since by increasing x_1 and changing x_2 such that $x_2 = x_1$ we get an increase in f. We see the difficulties caused by non-differentiability. Similar observations are made in Example 3.3.1.* □

Example 5.1.2 *Consider a problem with $x_i \in R^2$ and the dynamic equation at stage i given by $x_{i+1}^1 = f_i^1(x_i, u_i)$, $x_{i+1}^2 = f_i^2(x_i, u_i) = x_i^1$ for some function $f_i^1 : R^{2+m} \to R$. Such a dynamic equation may be found in problems with* time delays *and in problems with* acceleration. *In such problems it is common to introduce artificial elements in the state vector in order to remember previous values of controls and/or states. Here x_i^2 is such artificial element.*

If we maximize $[ub_i^{i+1}(x_i, x_{i+1}) + ub_{i+2}(x_{i+1}, x_{i+2})]$ with respect to x_{i+1} for fixed x_i and x_{i+2} we see that x_{i+1}^2 cannot be changed. In other words, the assumption of interiority (which is implicit in the assumption of the KKT conditions holding) is violated. Thus for this important class of problem, the algorithm does not work. □

As we see the conditions that guarantee convergence are very restrictive, but as illustrated by the examples not easily weakened. In Howson and Sancho (1975) the problem was assumed to be without local constraints and it was assumed that the dynamic equation (5.5) permitted unique identification of u_i from (x_i, x_{i+1}). The latter assumption was also used in the examples in Chang, Chang and Luh (1989). The assumption of unique $(p_{i+1}, \lambda_i, \mu_i)$, also used in Zuo (1991), is very restrictive. It implies essentially that there can be no inequality constraints in the problem definition. This may to some extent be overcome by stage aggregation (at the expense of larger dimensions of each subproblem) as discussed in Subsection 2.5. In fact this was the solution suggested in Lee (1989) who observed the non-convergence of the algorithm in the context of hydro-thermal scheduling, see also Turgeon (1982).

If convergence does occur it can be expected to be slow. This may be seen as follows. The method can be compared to a coordinate search method. The only difference is that here the

coordinate that changes, one at a time, is a vector $x_i \in R^n$, not a scalar. The coordinate search method has a slower convergence rate that the steepest ascent method. Thus, from Luenberger (1989) p. 229 we expect that one complete iteration in the algorithm will give an improvement which is comparable to the improvement obtained by one steepest ascent iteration. This is not attractive.

The methods are supposed to be used within a dynamic programming context. If this dynamic programming is performed by a discretization of state space, then in fact the assumptions of differentiability and concavity are violated. For this implementation the convergence results do not apply (see below, though).

State Increment Dynamic Programming

A variant over the above method is the socalled state increment dynamic programming, Larson and Korsak (1970). The state increment dynamic programming can also be seen as a coordinate (vector) search, but with a different cycle. One iteration in the method can be described as follows. Let $j = 1$ and solve OCP under the additional restriction that only x_i^j are changed, $i = 0, \ldots, N$, while all x_i^k remain fixed for $k \neq j$. Then increase j by one and solve again. Continue this way until $j = n$. Then the next iteration starts again with $j = 1$. The algorithm terminates if no improvement is attained during an iteration.

As seen, the algorithm solves repeatedly a sequence of n optimal control problems, each one with a one-dimensional state vector. It was suggested solved by dynamic programming discretizing the state space.

Proposition 5.1.3 *Consider the OCP. Let a feasible initial trajectory be given. Assume that in all iteration for all i the u_i^* involved in the determination of ub_i^{i+1} is unique and that the corresponding KKT multipliers are unique. Then either the algorithm stops at a point satisfying the weak maximum principle, or any accumulation point satisfies this.*

Proof. The proof is similar to the proof of Proposition 5.1.2. □

The same comments as above concerning the restrictiveness of the assumptions and the expected slow rate of convergence can be made. Yakowitz (1983) analyzed the method and derived the linear rate of convergence, relating the method to the Gauss-Seidel method for solving linear equations.

The two methods described here, the progressive optimality principle and the state increment dynamic programming, do overcome the problem with the size of the state space. The cost of this is that they are iterative methods. The assumptions that guarantee convergence are quite strong. Often the methods are used under the assumption that u_i can be determined uniquely from the dynamic equation (5.5) (Larson and Korsak (1970), Zuo and Wu (1989)). This may be the case if $m = n$ and $\nabla_{u_i} f_i$ is nonsingular.

The method as presented in Larson and Korsak (1970) is supposed to be used within a dynamic programming context. If this dynamic programming is performed by a discretization of state space, then in fact the assumptions of differentiability and concavity are violated. For this implementation the convergence rate results do not apply in a strict sense although they may do so approximately, cf. Yakowitz (1983).

5.2 QLE Problems and Newton Methods

Now assume that all ub_i^{i+1} are twice continuously differentiable such that the function F in (5.9) is twice continuously differentiable. We may then consider application of Newton or Newton-like methods in order to achieve superlinear rate of convergence. We discuss this here, and first address the quadratic-linear problem.

QLE problems

As a special case of (5.4) - (5.7) we consider now the quadratic-linear problem, possibly with local linear equality constraints:

$$\max[\sum_{i=0}^{N-1} \tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^x x_i + R_i^u u_i \tag{5.21}$$
$$+\tfrac{1}{2}x_N'R_N^{xx}x_N + R_N^x x_N]$$
$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{5.22}$$
$$H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 \tag{5.23}$$

Any end point restrictions (5.7) that might have been are assumed included in (5.6) through (5.5) for $i = N-1$ such that $V_N = R^n$.

For $i = 0, \ldots, N-1$ $ub_i^{i+1}(x_i, x_{i+1})$ is then defined as the optimal value in the problem

$$ub_i^{i+1}(x_i, x_{i+1}) = \max_{u_i}[r_i(x_i, u_i)] \tag{5.24}$$
$$= \max_{u_i}[\tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^x x_i + R_i^u u_i]$$

subject to the constraints (5.22) - (5.23).

As shown in (4.41) the solution (u_i, p_{i+1}, μ_i) may be written as

$$\begin{pmatrix} u_i \\ p_{i+1}' \\ \mu_i' \end{pmatrix} = \tag{5.25}$$
$$\begin{pmatrix} -R_i^{uu} & F_i^{u\prime} & H_i^{u\prime} \\ F_i^u & 0 & 0 \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \cdot \left(\begin{pmatrix} R_i^{u\prime} \\ x_{i+1} - \overline{f}_i \\ \overline{h}_i \end{pmatrix} + \begin{pmatrix} R_i^{xu\prime} \\ -F_i^x \\ -H_i^x \end{pmatrix} x_i \right)$$

By appropriate definition of matrices K_i, L_i and M_i the solution (5.25) for u_i may be written as, cf. also (4.43),

$$u_i = K_i + L_i x_i + M_i x_{i+1} \tag{5.26}$$

Define the following matrices

$$S_i = \tfrac{1}{2}R_i^{xx} + R_i^{xu}L_i + 2L_i'R_i^{uu}L_i \tag{5.27}$$
$$T_i = R_i^{xu}M_i + L_i'R_i^{uu}M_i \tag{5.28}$$
$$U_i = \tfrac{1}{2}M_i'R_i^{uu}M_i \tag{5.29}$$
$$V_i = K_i'R_i^{xu\prime} + K_i'R_i^{uu}L_i + R_i^u L_i + R_i^x \tag{5.30}$$
$$W_i = K_i'R_i^{uu}M_i + R_i^u M_i \tag{5.31}$$
$$\rho_i = \tfrac{1}{2}K_i'R_i^{uu}K_i + R_i^u K_i \tag{5.32}$$

Actually, ρ_i will not be needed for the QLE problems.

Inserting the solution (5.26) into the criterion $r_i(x_i, u_i)$ at stage i we get the function $ub_i^{i+1}(x_i, x_{i+1})$. Using the matrices defined we see that

$$ub_i^{i+1}(x_i, x_{i+1}) = x_i' S_i x_i + x_i' T_i x_{i+1} + x_{i+1}' U_i x_{i+1} + V_i x_i + W_i x_{i+1} + \rho_i \tag{5.33}$$

We observe that this expression, applied to the situation with $K = 1$, cf. (5.11) - (5.14), indicates how to express the upper boundary relative to (x_0, x_N), cf. the remarks towards the end of the introduction to Chapter 4 and cf. Dreyfus and Kan (1973).

From (5.33) we find

$$\nabla^2 ub_i^{i+1} = \begin{pmatrix} 2S_i & T_i \\ T_i' & 2U_i \end{pmatrix} \tag{5.34}$$

The criterion (5.9) with controls u_i eliminated may then be rewritten as

$$\begin{aligned} F(x) = & \sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1}) = \\ & \sum_{i=0}^{N-1} x_i' S_i x_i + x_i' T_i x_{i+1} + x_{i+1}' U_i x_{i+1} + V_i x_i + W_i x_{i+1} + \rho_i \end{aligned} \tag{5.35}$$

The first and second order derivatives are found as

$$\nabla_{x_i} \sum_{j=0}^{N-1} ub_j^{j+1}(x_j, x_{j+1}) = \tag{5.36}$$

$$\begin{cases} 2x_0' S_0 + x_1' T_0' + V_0 & ,i = 0 \\ 2x_i' S_i + x_{i+1}' T_i' + V_i + x_{i-1}' T_{i-1} + 2x_i' U_{i-1} + W_{i-1} & ,1 \le i \le N-1 \\ x_{N-1}' T_{N-1} + 2x_N' U_{N-1} + W_{N-1} & ,i = N \end{cases}$$

$$\nabla^2_{x_i x_j} \sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1}) = \begin{cases} 2S_0 & ,i = j = 0 \\ 2(U_{i-1} + S_i) & ,1 \le i = j \le N-1 \\ T_i' & ,j = i+1 \\ T_{i-1} & ,j = i-1 \\ 2U_{N-1} & ,i = j = N \\ 0 & ,\text{otherwise} \end{cases} \tag{5.37}$$

We therefore have the following algorithm:

Algorithm

Step 1 For $i = 0, \ldots, N-1$ find matrices (5.27) - (5.31) using (5.25) - (5.26).

Step 2 Maximize by any method the criterion function (5.35) over $R^{(N+1)n}$. This identifies x^*.

Step 3 Find u^* from (5.26).

Proposition 5.2.1 *Consider the QLE problem (5.21) - (5.23). Assume that the criterion function (5.21) is concave and that the matrices to be inverted in (5.25) are nonsingular. Then the algorithm solves the problem. The computational complexity is at least* $O(N(m+n+\ell)^3)$.

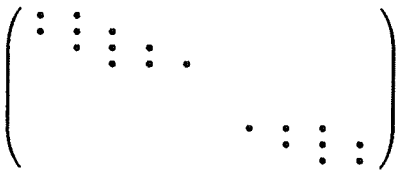


Figure 5.1: Structure of the Hessian matrix

Proof. The solution found in (5.25) is under the assumptions taken the global optimum such that indeed ub_i^{i+1} is as defined in (5.33) on the set $W_i^{i+1} = R^{2n}$. Maximization of (5.35) on $R^{(N+1)n}$ therefore solves the problem with respect to x, and u^* is derived as described. The computational complexity of the determination of ub_i^{i+1} is $O((m+n+\ell)^3)$, cf. (5.25) and therefore the computational complexity of the algorithm is at least $O(N(m+n+\ell)^3)$, irrespective of what method is used for maximizing (5.35). □

If x_0 is given with an initial value $\underline{x}_0$ rather than being constrained together with u_0 as in (5.23) then the assumption that the matrices to be inverted in (5.25) are nonsingular does not hold. However, it is not a problem to consider x_0 fixed, as discussed several times before.

As seen in Figure 5.1, $\nabla^2 F$ has a band matrix structure. This was applied in Chang, Chang and Luh (1989).

We may solve the global problem (5.9) - (5.10) as a dynamic programming problem as in Section 4.2 by formulating the following optimal control problem in state variables $x_i \in R^n$ and control variables $z_i \in R^n$:

$$\max[\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, z_i)] \tag{5.38}$$

$$x_{i+1} = z_i \tag{5.39}$$

Thus, in Step 2 of the above algorithm we may solve the problem (5.38) - (5.39) by dynamic programming with (5.38) defined in (5.35). The algorithm will solve the QLE problem in one iteration.

As shown in Proposition 4.3.1 the computational complexity of the backwards DP applied to (5.38) - (5.39) is $O(Nn^3)$ and coupling this with the result of Proposition 5.2.1 we find the total computational complexity in combination with backwards DP is $O(N(m+n+\ell)^3 + n^3)$.

The computational complexity, $O(N(m+n+\ell)^3)$, is worse than that of applying directly backwards DP, $O(N(m+\ell)^3)$, cf. Proposition 4.3.1. It follows that only if advantage can be taken of parallel processing in the determination of the ub_i^{i+1}'s the above method may be advantageous to the backwards DP in terms of computational complexity.

Non-QL Problems

In the more general case of non quadratic-linear problems, we may apply the above solution for the QLE problem in an iterative procedure in order to obtain quadratic rate of convergence.

We consider the following problem

$$\max[\sum_{i=0}^{N-1} r_i(x_i,u_i)+r_N(x_N)] \tag{5.40}$$

$$x_{i+1}=f_i(x_i,u_i) \tag{5.41}$$

$$h_i(x_i,u_i)=0 \tag{5.42}$$

Let there be given an iteration point (x^k,u^k,p^k,μ^k). Now derive the quadratic-linear approximation, similar in form to (5.33) as follows. First define the Hamiltonian as

$$H_i(x_i,u_i,p_{i+1},\mu_i)=r_i(x_i,u_i)+p_{i+1}f_i(x_i,u_i)-\mu_i h_i(x_i,u_i) \tag{5.43}$$

and then let

$$R_i^x=\nabla_x r_i(x_i^k,u_i^k) \tag{5.44}$$

$$R_i^u=\nabla_u r_i(x_i^k,u_i^k) \tag{5.45}$$

$$R_i^{xx}=\nabla_{xx}^2 H_i(x_i^k,u_i^k,p_{i+1}^k,\mu_i^k) \tag{5.46}$$

$$R_i^{uu}=\nabla_{uu}^2 H_i(x_i^k,u_i^k,p_{i+1}^k,\mu_i^k) \tag{5.47}$$

$$R_i^{xu}=\nabla_{xu}^2 H_i(x_i^k,u_i^k,p_{i+1}^k,\mu_i^k) \tag{5.48}$$

$$F_i^x=\nabla_x f_i(x_i^k,u_i^k) \tag{5.49}$$

$$F_i^u=\nabla_u f_i(x_i^k,u_i^k) \tag{5.50}$$

$$\overline{f}_i=f_i(x_i^k,u_i^k) \tag{5.51}$$

$$H_i^x=\nabla_x h_i(x_i^k,u_i^k) \tag{5.52}$$

$$H_i^u=\nabla_u h_i(x_i^k,u_i^k) \tag{5.53}$$

$$\overline{h}_i=h_i(x_i^k,u_i^k) \tag{5.54}$$

With these values we may formulate the problem (5.21) - (5.23) as the QLE approximation to (5.40) - (5.42) around the point $(x_i^k u_i^k,p_i^k,\mu_i^k)$ and we are again in the QLE case. The problem (5.21) - (5.23) is conveniently formulated in the variables $\delta x_i=x_i-x_i^k$ and $\delta u_i=u_i-u_i^k$ as

$$\max[\sum_{i=0}^{N-1} \tfrac{1}{2}\delta x_i' R_i^{xx}\delta x_i+\delta x_i' R_i^{xu}\delta u_i+\tfrac{1}{2}\delta u_i' R_i^{uu}\delta u_i+R_i^x\delta x_i+R_i^u\delta u_i \tag{5.55}$$
$$+\tfrac{1}{2}\delta x_N' R_N^{xx}\delta x_N+R_N^x\delta x_N]$$

$$\delta x_{i+1}=F_i^x\delta x_i+F_i^u\delta u_i+\overline{f}_i \tag{5.56}$$

$$H_i^x\delta x_i+H_i^u\delta u_i-\overline{h}_i=0 \tag{5.57}$$

We may formulate the algorithm as follows. Assume that we have a (x^k,u^k,p^k,μ^k) at iteration k; the algorithm may be initiated by $p^0=0$, $\mu^0=0$ and any (x^0,u^0). We use DP to solve in Step 2, but any other suitable method may be applied, cf. the discussion after the previous algorithm. Then proceed as follows:

Algorithm

Step 1 For $i=0,\ldots,N-1$ define the matrices (5.44) - (5.54); (5.25) - (5.26) and (5.27) - (5.31).

Step 2 Solve by dynamic programming the problem (5.38) - (5.39) with (5.38) defined in (5.35). This identifies x^{k+1}.

Step 3 Find $(u^{k+1}, p^{k+1}, \mu^{k+1})$ from (5.25)

Step 4 Let $k = k + 1$ and go to Step 1.

Proposition 5.2.2 *Assume that in a neighborhood around the unique optimal solution (x^*, u^*) to (5.40) - (5.42) r_i, f_i and h_i are twice continuously differentiable for all i, that at all iterations all $\nabla^2 ub_i^{i+1}$ are negative definite and that all (p_{i+1}^k, μ_i^k) are unique. Then there is a neighborhood around the optimal (x^*, u^*, p^*, μ^*) such that for any initial (x^k, u^k, p^k, μ^k) in this neighborhood the algorithm converges towards (x^*, u^*, p^*, μ^*) at a quadratic rate.*

Proof. The algorithm may be seen as Wilson's application of Newton's method. In this, Newton iterations are taken in the variables (x, u, p, μ) in order to satisfy the system of KKT equations. See Luenberger (1989) p 431. - We shall elaborate on this in Chapter 7. □

The desirable superlinear rate of convergence may therefore be achieved. However, as seen in the discussion of the QLE problem, the computational complexity of the solution of the QLE approximation is not attractive. Therefore the present algorithm seems inefficient, see also Chapter 7. The only argument that is in favor of the present approach is that parallel computations may be applied in the setting up of matrices (5.44) - (5.54).

5.3 Nonsmooth Smaller Upper Boundaries

We shall conclude this chapter with an analysis of the Progressive Optimality Principle of Section 5.1 when applied to nonsmooth functions which, at least for constrained OCP, will be involved when using the ub_i^{i+1}'s.

We take as a specific case the following problem:

$$\max[\sum_{i=1}^{N} R_i^u u_i] \tag{5.58}$$

$$x_{i+1} = x_i + F_i^u u_i \tag{5.59}$$

$$G_i^u u_i \leq \underline{g}_i, \; i = 1, \ldots, N \tag{5.60}$$

$$x_1 = 0 \tag{5.61}$$

$$x_{N+1} \leq b \tag{5.62}$$

which is seen also to be the linear programming problem

$$\max[\sum_{i=1}^{N} R_i^u u_i] \tag{5.63}$$

$$\sum_{i=1}^{N} F_i^u u_i \leq b \tag{5.64}$$

$$G_i^u u_i \leq \underline{g}_i, \; i = 1, \ldots, N \tag{5.65}$$

We shall solve the latter problem by the following decomposition scheme. The optimization takes place in an interplay between the "center" and the N "units". The center decides upon a

distribution of the common resource b between the units, i.e., the center decides upon the magnitude of b_i, $i = 1, \ldots, N$, satisfying $\sum_{i=1}^{N} b_i = b$. Knowing the b_i, each of the units finds the solution to its local problem

$$\max[R_i^u u_i] \tag{5.66}$$
$$F_i^u u_i \leq b_i \tag{5.67}$$
$$G_i^u u_i \leq \underline{g}_i \tag{5.68}$$

Having found the optimal solution and the corresponding shadow prices with respect to (5.67), the units communicate these prices to the center. The center then determines, if the distribution of b among the units is optimal (i.e., the problem is solved) or it finds another distribution, and the process is repeated.

Such decomposition is known as primal or resource decomposition, and in particular for the linear programming problem (5.63) - (5.65) it may be called Kornai-Liptak decomposition (Kornai and Liptak (1965)).

Let us define the functions $ub_i : R^n \to R$ as

$$ub_i(b_i) = \max[R_i^u u_i] \tag{5.69}$$
$$F_i^u u_i \leq b_i \tag{5.70}$$
$$G_i^u u_i \leq \underline{g}_i \tag{5.71}$$

The basic idea in the solution principle above is therefore, that if the marginal increase in $ub_i(b_i)$ is greater that the marginal decrease in $ub_j(b_j)$ for some small redistribution between b_i and b_j, then a redistribution of resources should take place between unit j and unit i.

It may be possible to make an algorithm work, based on this idea. The first problem is to get a correct analysis of the marginal increases and decreases. As Example 5.1.1 above shows, it is insufficient to analyze directional derivatives along the coordinate axes in case of non-smoothness. In the present context this means that *it is insufficient to consider redistribution of one resource b^k at a time.* This again means that the concept of shadow price, or marginal value, must be revised, if it is understood in the traditional way, related to a one-resource-at-a-time idea. Let us consider this further.

Assume a b_i given. Assume that the corresponding solution to (5.69) - (5.71) is $u_i^*(b_i)$. Then because the constraints are linear there exist Lagrange multiplier row vectors $p_i \in R^n$ and $\lambda_i \in R^m$ such that

$$R_i^u = p_i F_i^u + \lambda_i G_i^u \tag{5.72}$$
$$p_i \geq 0, \ \lambda_i \geq 0 \tag{5.73}$$
$$p_i(F_i^u u_i^* - b_i) = 0, \ \lambda_i(G_i^u u_i^* - \underline{g}_i) = 0 \tag{5.74}$$

We let $KKT_i^p(u_i^*)$ denote the set of KKT multiplies $\{p_i\}$ satisfying this, together with some λ_i. Then the directional derivative in the direction $s_i \in R^n$ is (assuming the Mangasarian-Fromowitz constraint qualification holds) given as

$$\min_{p_i}[p_i s_i] \tag{5.75}$$
$$p_i \in KKT_i^p(u_i^*) \tag{5.76}$$

Now assume the same given for unit j. We can then conclude that a redistribution of resources between unit i and unit j in the direction $s \in R^n$ would increase the criterion, if for some $s \in R^n$

the optimal criterion value in the following problem is positive:

$$\min_{(p_i,p_j)} [p_i s + p_j(-s)] \tag{5.77}$$

$$p_i \in KKT_i^p(u_i^*) \tag{5.78}$$

$$p_j \in KKT_j^p(u_j^*) \tag{5.79}$$

The criterion value in this can be written $(p_i - p_j)s$. We can therefore see that the criterion value can be increased for some s, if and only if we can *not* have $p_i = p_j$. We can also express this as follows: a redistribution can give an increase in the criterion value, if and only if $KKT_i^p(u_i^*) \cap KKT_j^p(u_j^*) = \emptyset$.

To appreciate the result, observe that the KKT conditions (5.72) - (5.74) for the two problems for i and j can be written in the following form:

$$\begin{pmatrix} R_i^u \\ R_j^u \end{pmatrix} = \tag{5.80}$$

$$p_i \begin{pmatrix} F_i^u \\ 0 \end{pmatrix} + p_j \begin{pmatrix} 0 \\ F_j^u \end{pmatrix} + \lambda_i \begin{pmatrix} G_j^u \\ 0 \end{pmatrix} + \lambda_j \begin{pmatrix} 0 \\ G_j^u \end{pmatrix}$$

$$p_i \geq 0, p_j \geq 0, \lambda_i \geq 0, \lambda_j \geq 0 \tag{5.81}$$

$$p_i(F_i^u u_i^* - b_i) = 0,\ p_j(F_j^u u_j^* - b_j) = 0 \tag{5.82}$$

$$\lambda_i(G_i^u u_i^* - \underline{g}_i) = 0, \lambda_j(G_j^u u_j^* - \underline{g}_j) = 0 \tag{5.83}$$

Consider the problem

$$\max_{(u_i,u_j)} [R_i^u u_i + R_j^u u_j] \tag{5.84}$$

$$F_i^u u_i + F_j^u u_j \leq b_i + b_j \tag{5.85}$$

$$G_i^u u_i \leq \underline{g}_i \tag{5.86}$$

$$G_j^u u_j \leq \underline{g}_j \tag{5.87}$$

From the KKT conditions we know that (u_i^*, u_j^*) is optimal in this problem if and only if there is a common multiplier $p = p_i = p_j$, satisfying (5.80) - (5.83). This is, not surprisingly, just another formulation of the previous result: a redistribution can increase the criterion value, if and only if $KKT_i^p(u_i^*) \cap KKT_j^p(u_j^*) = \emptyset$.

In case of a non-unique solution to the problem (5.69) - (5.71) defining ub_i, we get essentially the same result, if we consider all solutions. We denote by $U_i^*(b_i)$ the set of optimal solutions in (5.69) - (5.71). As before $KKT_i^p(u_i^*)$ denotes the set of the KKT multipliers p_i satisfying, with some λ_i, the KKT conditions (5.72) - (5.74) for a $u_i^* \in V_i^*(b_i)$. Then the directional derivative in the direction $s_i \in R^n$ is given as

$$\max_{u_i^*}[\min_{p_i}[p_i s_i]] \tag{5.88}$$

$$p_i \in KKT_i^p(u_i^*) \tag{5.89}$$

$$u_i^* \in U_i^*(b_i) \tag{5.90}$$

From this we can similarly as above derive the necessary and sufficient condition that a redistribution between unit i and unit j can increase the criterion:
$KKT_i^p(u_i^*) \cap KKT_j^p(u_j^*) = \emptyset$ for all $u_i^* \in U_i^*(b_i)$ and all $u_j^* \in U_j^*(b_j)$.

Example 5.3.1 *Consider a problem composed of the following three subproblems:*
Subproblem 1:

$$\max_{u_1}[u_1^1 + u_1^2]$$

$$u_1^1 + u_1^2 \leq b_1^1 = 3$$

$$\tfrac{1}{2}u_1^1 + u_1^2 \leq b_1^2 = 1.5$$

$$u_1^1 \leq 1$$

$$u_1^2 \leq 1$$

Subproblem 2:

$$\max_{u_2}[u_2^1 + u_2^2]$$

$$\tfrac{1}{2}u_2^1 + u_2^2 \leq b_2^1 = 1.5$$

$$u_2^1 + u_2^2 \leq b_2^2 = 3$$

$$u_2^1 \leq 1$$

$$u_2^2 \leq 1$$

Subproblem 3:

$$\max_{u_3}[u_3^1 + u_3^2]$$

$$u_3^1 \leq b_3^1 = 1$$

$$u_3^1 \leq b_3^2 = 1$$

$$u_3^2 \leq 1$$

The distribution of the two common resources $(b_1, b_2) = (5.5, 5.5)$ *has been made; the two first restrictions in each subproblem show that the distribution to the three subproblems is (3,1.5), (1.5,3) and (1,1), respectively. See also Figure 5.2.*

We find the following unique optimal solutions and the corresponding KKT^p *multiplier sets concerning the two first restrictions for the three problems:*

Subproblem 1: $u_1^*(b_1) = (1,1)'$,
and $KKT_1^p(u_1^*) = \{p_1^1 = 0, 0 \leq p_1^2 \leq 1\}$
Subproblem 2: $u_2^*(b_2) = (1,1)'$,
and $KKT_2^p(u_2^*) = \{0 \leq p_2^1 \leq 1, p_2^2 = 0\}$
Subproblem 3: $u_3^*(b_3) = (1,1)'$,
and $KKT_3^p(u_3^*) = \{0 \leq p_3^1, 0 \leq p_3^2, p_3^1 + p_3^2 = 1\}$
We see that $KKT_1^p \cap KKT_2^p = \{(0,0)\} \neq \emptyset$, $KKT_1^p \cap KKT_3^p = \{(0,1)\} \neq \emptyset$ *and* $KKT_2^p \cap KKT_3^p = \{(1,0)\} \neq \emptyset$

It is therefore not profitable to redistribute between any two of the subproblems.

Yet, $KKT_1^p \cap KKT_2^p \cap KKT_3^p = \emptyset$. *It may be verified that with the redistribution given by* $\Delta b_1 = (-1,0)'$, $\Delta b_2 = (0,-1)'$, $\Delta b_3 = (1,1)'$ *the optimal solution of the subproblems is* $u_1^* = (1,1)'$, $u_2^* = (1,1)'$, $u_3^* = (2,1)'$. *As this increases the criterion from 6 to 7 the former distribution was not optimal.* □

Let us now state the results in the following form.

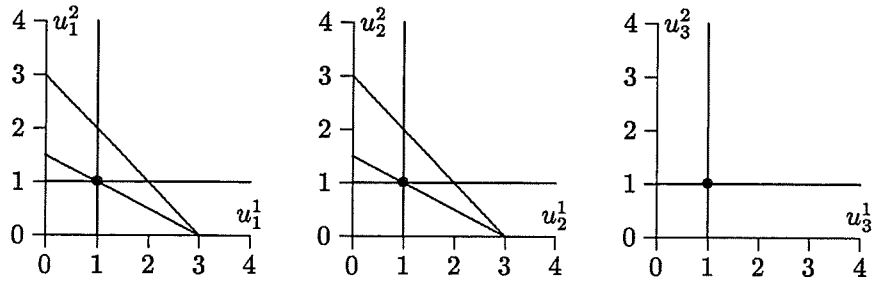


Figure 5.2: Restrictions for the three subproblems of Example 5.3.1

Proposition 5.3.1 *The resource distribution vector* $b = (b_1, b_2, \ldots, b_N)$ *is optimal if and only if* $KKT_1^p \cap KKT_2^p \cap \ldots \cap KKT_N^p \neq \emptyset$ *for all* $u_i^* \in U_i^*(b_i)$, *i=1,...,N.*

Proof. The condition is exactly the same as the KKT conditions for the whole problem. These are necessary and sufficient conditions for optimality in the linear programming problem. □

Proposition 5.3.2 *If the resource distribution vector b is not optimal, then an improving direction* $\Delta b = (\Delta b_1, \Delta b_2, \ldots, \Delta b_N)$ *is given as an optimal solution to*

$$\max_{\Delta b}[\max_{u^*}[\min_{p} \sum_{i=1}^{N} p_i \Delta b_i]]]$$

$$(p_1, p_2, \ldots, p_N) \in KKT_1^p \times KKT_2^p \times \ldots \times KKT_N^p$$

$$u^* \in U_1^*(b_1) \times U_2^*(b_2) \times \ldots \times U_N^*(b_N)$$

$$\sum_{i=1}^{N} \Delta b_i = 0$$

Proof. The inner optimization problem ($\max_{u^*} \min_p$) gives the directional derivative relative to a change in the direction Δb, cf. Gauvin and Dubeau (1982). The outer optimization ($\max_{\Delta b}$) is with respect to all directions Δb, which leave $\sum_{i=1}^{N} b_i = b$. If there is an improving direction it will therefore be found. The maximization of $\sum_{i=1}^{N} ub_i$ is a maximization of a concave function over a convex set, so if a point is not optimal it is possible to find an improving feasible direction. □

In general it is not easy to solve the optimization problem of Proposition 5.3.2. Assuming that it can be solved we can now describe a procedure to solve the problem by the decomposition procedure in several ways. For instance the center can start with comparing all pairs of KKT-sets from units (i, j), and make a redistribution between these two units, if the distribution is not already optimal, i.e. if $KKT_i^p \cap KKT_j^p = \emptyset$. Then all triples must be compared and brought to optimality. Then all sets of 4, 5, etc. subproblems, until finally all N units are compared at the same time.

Alternatively the center may first decide to bring the distribution between units 1 and 2 to optimality, then similarly for units 1, 2 and 3, then units 1, 2, 3 and 4, etc., until finally all units are compared. The important point is, that eventually all subproblems' KKT^p sets are compared simultaneously. If an improving direction is not found then stop; otherwise, take a step in the direction found of such length that a new extreme point is found.

Proposition 5.3.3 *Let a problem of the form (5.63) - (5.65) be given. Assume that it has an optimal solution. If the direction finding problem in Proposition 5.3.2 can be solved, then the problem may be solved in a finite number of steps by either of the two described procedures.*

Proof. The assumption that the Mangasarian-Fromowitz constraint qualification holds, in order that the formulas for the directional derivatives are valid, is not critical, since it can always be made to hold by introduction of a suitable number of artificial variables, cf. the discussion in Section 2.5. This also permits starting the iterations from an arbitrary point. From the previous Proposition we know how to calculate an improving direction Δb^*. For given Δb^* the parametric programming to determine the steplength is standard in linear programming theory. Each step in any of the algorithms brings us to a different extreme point, and gives an improvement. Therefore the algorithm will never twice be in the same extreme point. Since there is only a finite number of extreme points, the algorithm will terminate in a finite number of steps. □

We observe as an important consequence that *it is not possible to solve the problem by a coordinate search*, since it is in this way not possible to compare all possible redistributions simultaneously. Therefore also the state increment dynamic programming and the progressive optimality principle, cf. Section 5.1, break down in the case of inequality constraints.

If we had a nonlinear problem with smooth criterion and restrictions we would have to go through essentially the same procedure to find an improving direction. The differences would be that we would have to find the direction (Proposition 5.3.2) in a linearized version of the problem and that we would have to find the steplength by a one dimensional search.

In conclusion we see that *if the ub_i are non-smooth, then sufficient optimality conditions can* not *be given in an independent stagewise decomposable form*: Some kind of dependence between the stages is required in order to get sufficient optimality conditions. The result also holds for $n = 1$, as illustrated in Example 3.3.1.

It is noteworthy that this dependence may for instance be secured by the adjoint vector as in the classical maximum principle, as will be illustrated in the next example.

Example 5.3.2 *Let us now apply the classical maximum principle. We want to illustrate that this principle does not accept the solution indicated in Example 5.3.1 as optimal.*

We formulate the problem of Example 5.3.1 above as an OCP as follows: $x_1 = \underline{x}_1 = (0,0)'$, $x_4 = \underline{x}_4 = (5.5, 5.5)'$, $r_N \equiv 0$ *and the following local conditions:*

Stage 1:

$$r_1(x_1^1, x_1^2, u_1^1, u_1^2, u_1^3, u_1^4) = u_1^1 + u_1^2$$

$$x_2 = f_1(x_1^1, x_1^2, u_1^1, u_1^2, u_1^3, u_1^4) = \begin{pmatrix} x_1^1 + u_1^1 + u_1^2 + u_1^3 \\ x_1^2 + \frac{1}{2}u_1^1 + u_1^2 + u_1^4 \end{pmatrix}$$

$$g_1^1(x_1, u_1) = u_1^1 - 1 \leq 0$$

$$g_1^2(x_1, u_1) = u_1^2 - 1 \leq 0$$

$$g_1^3(x_1, u_1) = -u_1^3 \leq 0$$

$$g_1^4(x_1, u_1) = -u_1^4 \leq 0$$

Stage 2:

$$r_2 = u_2^1 + u_2^2$$

$$x_3 = \begin{pmatrix} x_2^1 + \frac{1}{2}u_2^1 + u_2^2 + u_2^3 \\ x_2^2 + u_2^1 + u_2^2 + u_2^4 \end{pmatrix}$$

$$g_2^1(x_2,u_2) = u_2^1 - 1 \leq 0$$
$$g_2^2(x_2,u_2) = u_2^2 - 1 \leq 0$$
$$g_2^3(x_2,u_2) = -u_2^3 \leq 0$$
$$g_2^4(x_2,u_2) = -u_2^4 \leq 0$$

Stage 3:

$$r_3 = u_3^1 + u_3^2$$
$$x_4 = \begin{pmatrix} x_3^1 + u_3^1 + u_3^3 \\ x_3^2 + u_3^1 + u_3^4 \end{pmatrix}$$
$$g_3^1(x_3,u_3) = -u_3^3 \leq 0$$
$$g_3^2(x_3,u_3) = -u_3^4 \leq 0$$
$$g_3^3(x_3,u_3) = u_3^2 - 1 \leq 0$$
$$h_3^1(x_3,u_3) = x_3^1 + u_3^1 + u_3^3 - 5.5 = 0$$
$$h_3^2(x_3,u_3) = x_3^2 + u_3^1 + u_3^4 - 5.5 = 0$$

The hypothetical solution indicated in Example 5.3.1 above is interpreted as $u_1 = (1,1,1,0)'$, $u_2 = (1,1,0,1)'$, $u_3 = (1,1,0,0)'$, $x_1 = (0,0)'$, $x_2 = (3.0,1.5)'$, $x_3 = (4.5,4.5)'$ *and* $x_4 = (5.5,5.5)'$.

We shall now see if we with this solution can have the adjoint equations and optimization of the Hamiltonian satisfied.

We have $p_4 = (0,0)$ *since* $r_4 \equiv 0$ *and the end point is free. At stage 3 we find the optimal solution to the maximization of the Hamiltonian as* $u_3 = (1,1,0,0)'$. *We determine* p_3 *by the adjoint equation*

$$p_3 = (p_3^1, p_3^2) = \nabla_x(r_3 + p_4 f_3 - \lambda_3 g_3 - \mu_3 h_3)$$

We find

$$p_3 = (\mu_3^1, \mu_3^2)$$

Here (μ_3^1, μ_3^2) *must satisfy the KKT conditions* $0 = \nabla_u(H - \lambda g - \mu h)$ *or*

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} =$$

$$\begin{pmatrix} 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \lambda_3^1 \begin{pmatrix} 0 \\ 0 \\ -1 \\ 0 \end{pmatrix} - \lambda_3^2 \begin{pmatrix} 0 \\ 0 \\ 0 \\ -1 \end{pmatrix} - \lambda_3^3 \begin{pmatrix} 0 \\ -1 \\ 0 \\ 0 \end{pmatrix} - \mu_3^1 \begin{pmatrix} 1 \\ 0 \\ 1 \\ 0 \end{pmatrix} - \mu_3^2 \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \end{pmatrix}$$

$$\lambda_3^1 \geq 0, \lambda_3^2 \geq 0, \lambda_3^3 \geq 0$$

We find that $(\mu_3^1, \mu_3^2) \in \{0 \leq \mu_3^1, 0 \leq \mu_3^2, \mu_3^1 + \mu_3^2 = 1\}$ *and therefore* $p_3 \in \{p_3^1 \leq 0, p_3^2 \leq 0, p_3^1 + p_3^2 = -1\}$ *(note the change of sign from* KKT_3^p *of Example 5.3.1!)*

At stage 2 we maximize the Hamiltonian

$$u_2^1 + u_2^2 + p_3^1(x_2^1 + \tfrac{1}{2}u_2^1 + u_2^2 + u_2^3) + p_3^2(x_2^2 + u_2^1 + u_2^2 + u_2^4)$$

subject to the four local constraints $g_2^j \leq 0$. *For any* p_3 *in the set identified above we get* $(u_2^1, u_2^2) = (1,1)$ *as optimal solution (but not necessarily unique). It is possible to get* $(u_2^3, u_2^4) = (0,1)$ *but only for* $p_3 = (-1,0)$. *If therefore* $u_2 = (1,1,0,1)'$ *shall maximize the Hamiltonian (as suggested in Example 5.3.1),* p_3 *must be* $(-1,0)$.

We find p_2 *from the adjoint equation:*

$$p_2 = \nabla_x(r_2 + p_3 f_2 - \lambda_2 g_2 - \mu_2 h_2) = p_3$$

At stage 1 we maximize the Hamiltonian

$$u_1^1 + u_1^2 + p_2^1(x_1^1 + u_1^1 + u_1^2 + u_1^3) + p_2^2(x_1^2 + \tfrac{1}{2}u_1^1 + u_1^2 + u_1^4)$$

subject to the four local constraints $g_1^j \leq 0$. *For any* p_2 *in the set identified above we get* $(u_1^1, u_1^2) = (1,1)$ *as optimal solution (but not necessarily unique). It is possible to get* $(u_1^3, u_1^4) = (1,0)$ *but only for* $p_2 = (0,-1)$. *If therefore* $u_1 = (1,1,1,0)'$ *shall maximize the Hamiltonian (as suggested in Example 5.3.1),* p_2 *must be* $(0,-1)$. *However, this is inconsistent with the previous requirements that* $p_2 = p_3$ *and* $p_3 = (-1,0)$.

We therefore see that it is not possible to have the adjoint equations, the dynamic equation and the maximization of the Hamiltonian satisfied for all i. In other words we reject the hypothesis that the optimal solution is as suggested in Example 5.3.1.

We see that the classical maximum principle provide stronger optimality conditions that the progressive optimality principle. *If fact, as the criterion is concave, the constraints and the dynamics linear, the classical maximum principle provide necessary as well as sufficient optimality conditions for this problem.* □

5.4 Conclusions

In the present chapter we have considered application of the smaller upper boundaries in algorithms.

Under assumptions of smoothness we have derived explicit formulae for first and second order derivatives. Based on this, we have stated an algorithm for the QLE problem in Section 5.2. Compared to a direct application of DP, this is not attractive. The approach also extends to non QLE problems, where an algorithm with quadratic rate of convergence has been developed. Again, compared to an application of DP, this is not attractive in view of the alternatives in Chapter 7.

If parallel computations may be applied, the approach may be attractive, since in this case the major part of the computations, with complexity $O(N(n+m+\ell)^3)$, may be performed in parallel. Smoothness of the upper boundaries is as essential prerequisite when the solutions of the N subproblems are to be coordinated.

It therefore seems that from a computational point of view two major prerequisites, smoothness of smaller upper boundaries and parallel computations, are necessary if the smaller upper boundaries shall be applied.

As discussed in Sections 2.7 and 2.8 the conditions for existence of the first and second order derivatives are quite strong. Thus, it seems difficult to guarantee smoothness if there are local inequality constraints in the problem definition. This was discussed at length in Section 5.3 where also the difficulties in handling the optimization in case of nonsmoothness was illustrated.

Finally, other justifications of using smaller upper boundaries are to be found in relation to interpretations of optimality conditions, cf. Chapter 3, as well as in relating to other approaches, in particular the mathematical programming resource decomposition idea as treated in Section 5.3.

Chapter 6

Maximum Principle Algorithms

Algorithms for the solution of the optimal control problem, based on the maximum principle, were suggested early after the appearance of the maximum principle, see e.g. Katz (1962) and Fan and Wang (1964).

They express in condensed form the stagewise approach to the OCP analysis and solution, working alternatively forwards and backwards through the stages, with u_i and x_{i+1} found in the forwards run and p_i in the backwards run. The idea may be indicated as the following prototype maximum principle algorithm:

Step 1 Find u_i maximizing the Hamiltonian and find x_{i+1} from the dynamic equation for $i = 0, \ldots, N-1$. Go to Step 2.

Step 2 Find p_i from the adjoint equations, $i = N, \ldots, 1$. Go to Step 1.

The major strength in this type of algorithm is the simplicity and the intuitive appeal. The potential efficiency should be ascribed to the simplicity of the calculations. In the forwards run, u_i will be found as the solution to a maximization problem as the following,

$$\max_{u_i}[r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i)] \tag{6.1}$$

$$u_i \in U_i(x_i) \tag{6.2}$$

As typically $m << N$, it is likely to be much simpler to solve N such small problems (size indicated by m) than to solve one large problem (size indicated by $N(n+m)$). Further, the determination of x_{i+1} in the forwards direction and p_i in the backwards direction is usually computationally simple.

If there is no dependency on the state in (6.2) then a feasible trajectory is simply constructed by using arbitrary feasible $u_i \in U_i$ and then construction of x_{i+1} from the dynamic equation.

On the other hand the shortcomings of this simple approach are severe. First, if indeed there is dependency on x in (6.2) - even if it is only representing, say, an end point constraint $x_N = \underline{x}_N$ - it is not obvious how to handle this. Second, the application of the costate vector as indicated will provide only gradient information for the selection of u_i in (6.1) - (6.2) and hence the rate of convergence will be at most linear.

Theoretical convergence analysis were often based on the transfer of results from mathematical programming, see the review in Polak (1973). In particular it was found that the simple maximum principle algorithm suggested above will not in general provide convergence, in part because the conditions of the maximum principle are not necessarily fulfilled at the optimum, cf. Section

1.3. The generalized maximum principle is more appropriate in this respect, and computational development of it was undertaken in Nahorski, Ravn and Vidal (1983), Nielsen (1985), Nielsen and Ravn (1985), Ferreira (1984), Ferreira (1990).

In this chapter we develop algorithms based on the maximum principle. Section 6.1 provides the essential formulae that permit the interpretation of the partial derivatives of the criterion function in relation to $\nabla_u H_i$, $\nabla_x H_i$ and p_i. Based on this, various gradient type algorithms are developed differing with respect to how the stepsize is controlled. Section 6.2 elaborates on the theme of the gradient and presents a simple projection algorithm.

While the algorithms of Sections 6.1 and 6.2 deal with problems where there is no dependence on the state in the local constraints (6.2), such dependency is discussed in Section 6.3. It is shown that this leads to a natural definition of feedback strategy, i.e., that u_i should be expressed as a function of x_i. This is then applied in Section 6.4 where a generalized maximum principle algorithm is developed, handling state dependent constraints. The algorithm also handles the situation where the upper boundaries are not smooth due to non-unique solutions and/or non-unique KKT multipliers, cf. Section 2.7.

6.1 Simple Maximum Principle Algorithms

In this section we consider the following OCP with local constraints that are independent of x:

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{6.3}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{6.4}$$

$$u_i \in U_i \tag{6.5}$$

$$x_0 = \underline{x}_0 \tag{6.6}$$

Whenever convenient, we shall eliminate r_N in order to express the dependence on u. In this case the terms $r_{N-1}(x_{N-1}, u_{N-1}) + r_N(x_N)$ in (6.3) will be eliminated, and substituted by the term $r_{N-1}(x_{N-1}, u_{N-1})$ redefined as

$$r_{N-1}(x_{N-1}, u_{N-1}) + r_N(f_{N-1}(x_{N-1}, u_{N-1})) \tag{6.7}$$

Interpretation of p

In the algorithms the costate vector p_i is used an as essential intermediate variable. In Section 1.3 we interpreted p_i in relation to an *optimal* trajectory. In connection with application of p_i in algorithms as here we must interpret p_i in relation to a non-optimal trajectory.

The essential observation for this section and the next one is that p_i can be interpreted as a gradient or partial derivative of the Hamiltonian. And in turn that the Hamiltonian may be interpreted as an approximation of the criterion-to-go. To see this, observe that we can eliminate the state vectors x_i and express the problem in terms of $\underline{x}_0$ and the controls u_i only:

$$\begin{aligned} x_1 &= f_0(\underline{x}_0, u_0) \\ x_2 &= f_1(f_0(\underline{x}_0, u_0), u_1) \\ x_3 &= f_2(f_1(f_0(\underline{x}_0, u_0), u_1), u_2) \\ &\vdots \end{aligned} \tag{6.8}$$

$$x_N = f_{N-1}(f_{N-2}(\dots f_0(x_0, u_0)\dots, u_{N-2}), u_{N-1}) \tag{6.9}$$

We can now recursively backwards define p_i to satisfy the following relations around a nominal trajectory (x, u), satisfying (6.8) - (6.9):

$$p_N = \nabla r_N(x_N) \tag{6.10}$$

and then recursively backwards, $i = N-1, \dots, 0$,

$$p_i = \nabla_x r_i(x_i, u_i) + p_{i+1} \nabla_x f_i(x_i, u_i) \tag{6.11}$$

With the definition of the Hamiltonian as

$$H_i(x_i, u_i, p_{i+1}) = \begin{cases} r_N(x_N) & \text{for } i = N \\ r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) & \text{for } 0 \le i < N \end{cases} \tag{6.12}$$

(6.10) - (6.11) amount to the backwards recursive definition of p_i

$$p_i = \nabla_x H_i(x_i, u_i, p_{i+1}). \tag{6.13}$$

We might start the elimination (6.8) - (6.9) of the state variables at any stage i (rather than at $i = 0$), such that for any $j > i$ we have

$$x_j = f_{j-1}(f_{j-2}(\dots f_i(x_i, u_i)\dots, u_{j-2}), u_{j-1}) \tag{6.14}$$

The following Proposition 6.1.1 interprets p_i and $\nabla_x H_i$ and Proposition 6.1.2 interprets $\nabla_u H_i$ in relation to such elimination of the state variables.

Proposition 6.1.1 *Assume that r_i and f_i are continuously differentiable for all i, that x_j is defined as in (6.14) for $j = i+1, \dots, N$ and p_i by (6.13) for $j = N, \dots, i$. Then*

$$p_i = \nabla_x H_i(x_i, u_i, p_{i+1}) = \nabla_{x_i}[\sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N)]$$

Proof. By (6.12) the formula holds for $i = N$. Now assume it holds for $i+1$. Then for i we find by using the chain rule, (6.13), (6.4) and (6.8) - (6.9)

$$\begin{aligned} &\nabla_{x_i}[\sum_{j=i}^{N-1} r_j(x_j, u_j) + r_N(x_N)] \\ &= \nabla_x r(x_i, u_i) + (\nabla_{x_{i+1}}(\sum_{j=i+1}^{N-1} r_j(x_j, u_j) + r_N(x_N)))\, \nabla_x f_i(x_i, u_i) \\ &= \nabla_x r_i(x_i, u_i) + p_{i+1} \nabla_x f_i(x_i, u_i) = \nabla_x H_i(x_i, u_i, p_{i+1}) = p_i \end{aligned}$$

The result then follows by induction. □

Proposition 6.1.2 *Assume that r_i and f_i are continuously differentiable for all i, that x_j is defined as in (6.14) for $j = i+1, \dots, N$, and p_i by (6.13) for $j = N, \dots, i$. Then*

$$\nabla_u H_i(x_i, u_i, p_{i+1}) = \nabla_{u_i}[\sum_{j=0}^{N-1} r_j(x_j, u_j) + r_N(x_N)]$$

Proof. For $i = N-1$ we find

$$\begin{aligned}
&\nabla_{u_{N-1}}(r_{N-1}(x_{N-1},u_{N-1}) + r_N(x_N)) \\
&= \nabla_u r_{N-1}(x_{N-1},u_{N-1}) + \nabla r_N(x_N)\nabla_u f_{N-1}(x_{N-1},u_{N-1}) \\
&= \nabla_u r_{N-1}(x_{N-1},u_{N-1}) + p_N \nabla_u f_{N-1}(x_{N-1},u_{N-1}) \\
&= \nabla_u H_{N-1}(x_{N-1},u_{N-1},p_N).
\end{aligned}$$

The result therefore holds for $i = N-1$. Now assume the relation holds for $i+1$. Then for i we find by using the chain rule, the relations (6.4), (6.13) and (6.8) - (6.9) and the previous Proposition 6.1.1

$$\begin{aligned}
&\nabla_{u_i}[\sum_{j=i}^{N-1} r_j(x_j,u_j) + r_N(x_N)] \\
&= \nabla_u r(x_i,u_i) + (\nabla_{x_{i+1}}(\sum_{j=i+1}^{N-1} r_j(x_j,u_j) + r_N(x_N)))\ \nabla_u f_i(x_i,u_i) \\
&= \nabla_u r_i(x_i,u_i) + p_{i+1}\nabla_u f_i(x_i,u_i) = \nabla_u H_i(x_i,u_i,p_{i+1}).
\end{aligned}$$

The result then follows by induction. □

Expressing the criterion function $[\sum_{i=0}^{N-1} r_i(x_i,u_i) + r_N(x_N)]$ in terms of u only, cf. (6.8) - (6.9), we may call it $r(u)$, i.e.

$$\begin{aligned}
r(u) = \quad & \\
& r_0(\underline{x}_0,u_0) + r_1(f_0(\underline{x}_0,u_0),u_1) + \ldots \\
& \ldots + r_N(f_{N-1}(\ldots(f_0(\underline{x}_0,u_0),u_1)\ldots),u_{N-1})
\end{aligned} \tag{6.15}$$

In this perspective the constraints (6.5) may be written as

$$u \in U_0 \times U_1 \times \ldots \times U_{N-1} \tag{6.16}$$

It follows from Proposition 6.1.2 that we may express the partial derivatives with respect to u_i of the criterion as

$$\nabla_{u_i} r(u) = \nabla_u H_i(x_i,u_i,p_{i+1}) \tag{6.17}$$

Proposition 6.1.2 is the basis for steepest ascent, conjugate gradients and quasi Newton methods in the unconstrained case ($U_i = R^m$) of (6.3) - (6.7), see e.g. Bertsekas (1974), and gradient projection method in the constrained case ($U_i \subset R^m$), see e.g. Papageorgieu (1985). It is also basis for steplength selection rules which are based on comparison with the gradient, e.g. the Armijo or the Goldstein rules.

A Maximum Principle Algorithm

Now consider the construction of an algorithm based on a stagewise maximization with respect to u_i. At stage i we might want to maximize the criterion-to-go with respect to the variable u_i. Thus

we want to solve the problem

$$\max[\sum_{j=i}^{N-1} r_j(x_j,u_j)+r_N(x_N)] \tag{6.18}$$

$$u_i \in U_i \tag{6.19}$$

where the state variables $x_j, j=i+1,\ldots,N$ are supposed eliminated by (6.8) - (6.9), or similarly that (6.4) remains fulfilled for $j=i+1,\ldots,N$.

However, we will approximate this expression by the Hamiltonian, i.e., we will solve

$$\max_{u_i}[H_i(x_i,u_i,p_{i+1})] \tag{6.20}$$

$$u_i \in U_i \tag{6.21}$$

Thus, the terms $(\sum_{j=i+1}^{N-1} r_i(x_i,u_i)+r_N(x_N))$ have been substituted by $p_{i+1}f_i(x_i,u_i)$.

By Proposition 6.1.2 we observe that the partial derivative of the criterion with respect to u_i is the same in (6.18) - (6.19) as in (6.20) - (6.21).

As we have seen in Chapter 3 this approximation is sufficiently accurate to permit identification of optimality conditions, provided some concavity, convexity and linearity assumptions are fulfilled.

Rather than using (6.20) - (6.21) directly for finding u_i we shall modify it a little bit. Thus, for a given nominal (x_i^k,u_i^k,p_{i+1}^k) from the previous iteration we solve at stage i the problem

$$\max_{u_i}[H_i(x_i^k,u_i,p_{i+1}^k)-\gamma\sum_{j=1}^{m}(u_i^j-(u_i^j)^k)^2] \tag{6.22}$$

subject to (6.21). The purpose is to assure (if possible) that the criterion in (6.22) is strictly concave with respect to u_i. Thus, if H_i is strictly concave with respect to u_i we may take $\gamma=0$, otherwise a sufficiently large γ maybe can assure the strict concavity. Observe that the partial derivatives with respect to u_i at the point u_i^k is the same in the two criteria (6.20) and (6.22).

We can now specify a Maximum Principle Algorithm the following way.

Maximum Principle Algorithm - Interpolation

Step 0 Choose $u_i^0 \in U_i$ and calculate x_{i+1}^0 from (6.4) for all i. Choose γ such that the criterion in (6.22) is strictly concave with respect to u_i for any (x_i,p_{i+1}). Let $k=0$ and $\alpha=1$

Step 1 Calculate p_i^k from (6.13) for $i=N,\ldots,1$

Step 2 Calculate u_i^* from (6.22) - (6.21) for $i=0,\ldots,N-1$

Step 3 If $u_i^*=u_i^k$ for all i then stop, else go to Step 4.

Step 4 Calculate $u_i=\alpha u_i^*+(1-\alpha)u_i^k$ and x_{i+1} from (6.4) for $i=0,\ldots,N-1$

Step 5 If $\sum_{i=0}^{N-1} r_i(x_i,u_i)+r_N(x_N) > \sum_{i=0}^{N-1} r_i(x_i^k,u_i^k)+r_N(x_N^k)$ then go to Step 6 else let $\alpha=\frac{1}{2}\alpha$ and go to Step 4.

Step 6 Let $k=k+1, u_i^k=u_i$ and $x_i^k=x_i$. Let $\alpha=1$. Go to Step 1.

Proposition 6.1.3 *Consider the problem (6.3) - (6.7). Assume that r_i and f_i are continuously differentiable for all i, and that γ can be and is selected such that the criterion in (6.22) is strictly concave with respect to u_i for any (x_i, p_{i+1}). Assume that all U_i are convex. Assume that the set $\{u \in R^{Nm} \mid \sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) \geq \sum_{i=0}^{N-1} r_i(x_i^0, u_i^0) + r_N(x_N^0), u_i \in U_i, x_{i+1} = f_i(x_i, u_i)\}$ is compact and nonempty.*

Then any accumulation point for the algorithm satisfies the weak maximum principle of Proposition 3.4.3.

If all U_i are given as $\{u_i \in R^m \mid g_i(u_i) \leq 0, h_i(u_i) = 0\}$ where g_i are convex and h_i linear and a constraint qualification holds then any accumulation point for the algorithm satisfies the weak maximum principle of Proposition 1.3.4.

If in addition for all i r_i is concave and strictly concave with respect to u_i and f_i linear then the algorithm converges to an optimal solution of the problem where the maximum principle of Proposition 3.4.3 is satisfied.

Proof. Let $F : R^{2Nm} \to R$ be the sum of the expressions in (6.22): $F(u^k; u) = \sum_{i=0}^{N-1}(H_i(x_i^k, u_i, p_{i+1}^k) - \gamma \sum_{j=1}^{m}(u_i^j - (u_i^j)^k)^2)$, where u^k is considered a parameter. The $u^* = (u_0^{*\prime}, \ldots, u_{N-1}^{*\prime})'$ from Step 2 maximizes $F(u^k; u)$ over the convex set given in (6.16). The function F is continuously differentiable and satisfies $\nabla F(u^k; u^k) = \nabla r(u^k)$, cf. (6.15), Proposition 6.1.2 and the fact that the derivative of the last terms in (6.22) vanish for $u_i^j = (u_i^j)^k$. The continuous differentiability of F implies $F(u^k; u^k + \alpha(u^* - u^k)) = F(u^k; u^k) + \alpha \nabla F(u^k; u^k)(u^* - u^k) + o(\alpha)$ where $o(\alpha)/\alpha \to 0$ as $\alpha \to 0$. Since F by assumption is strictly concave then u^* is unique and $u^* \neq u^k$ implies $F(u^k; u^*) > F(u^k; u^k)$, and also $o(\alpha) \leq 0$. From this follows $\nabla F(u^k; u^k)(u^* - u^k) > 0$, and since $\nabla F(u^k; u^k) = \nabla r(u^k)$ then also $\nabla r(u^k)(u^* - u^k) > 0$. Hence, $(u^* - u^k)$ is an improving direction for r.

We now observe that at all steps of the algorithm the values generated are unique since (6.22) is strictly concave with respect to u_i, U_i is convex and all functions are continuously differentiable. Moreover any value generated is a continuous function of the other values involved; thus, u_i from (6.22) - (6.21) is a continuous function of (x_i, p_{i+1}), x_{i+1} from (6.4) is a continuous function of u_i and p_i from (6.13) is a continuous function of (x_i, u_i, p_{i+1}).

Considering the algorithm as a mapping $(x^k, u^k) \to (x^{k+1}, u^{k+1})$ this mapping is then point-to-point and continuous and therefore closed. All iteration points are contained in a compact set. If $u^k \neq u^*$ then the criterion will increase as shown above. If $u^k = u^*$ then from Propositions 6.1.1 and 6.1.2 it follows that the weak maximum principle of Proposition 3.4.3 is fulfilled, and if the additional conditions on U_i are assumed then the weak maximum principle of Proposition 1.3.4 is fulfilled. From Luenberger (1989) pp. 187 - 188 the first part then follows. Under the assumptions of the second part the optimal solution is unique and satisfies the maximum principle, cf. Proposition 1.3.5. From Luenberger (1989) pp. 187 - 188 the second part then follows. □

Conditions to ensure that γ can in fact be selected such that the criterion in (6.22) is strictly concave with respect to u_i are discussed below, cf. (6.25) - (6.27).

Stepsize Control by a Control Penalty Approach

The previous algorithm was based on two main ideas, (1) the approximation of the criterion-to-go by a concave function with the same partial derivative and (2) a direct stepsize control.

Now we consider a different approach. Again we make an approximation of the criterion-to-go, but this time such that the approximation always underestimates the criterion value. This serves

as a penalty that will implicitly give a stepsize control. We consider application of penalties applied to the controls and to the states in turn. We start with the penalty on the control.

Although obviously there are many choices for selection such penalty function we shall here only consider quadratic functions. Thus, in one version the problem at stage i will be

$$\max_{u_i}[H_i(x_i^k, u_i, p_{i+1}^k) - \gamma \sum_{j=1}^{m}(u_i^j - (u_i^j)^k)^2] \tag{6.23}$$

$$u_i \in U_i \tag{6.24}$$

Here γ is a positive parameter and $(u_i^j)^k$ is a nominal value (the value from the previous iteration). Thus (6.23) - (6.24) is the same as (6.22) - (6.21).

In order to underestimate $(\sum_{j=1}^{N-1} r_j(x_j, u_j) + r_N(x_N))$ by this quadratic modification it suffices that the functions involved are continuously differentiable with a Lipschitz bound on the change in the gradient. Thus, there must be a constant L such that for all i

$$|\nabla r_i(x_i, u_i) - \nabla r_i(\tilde{x}_i, \tilde{u}_i)| < L|(x_i, u_i) - (\tilde{x}_i, \tilde{u}_i)| \tag{6.25}$$

$$|\nabla f_i(x_i, u_i) - \nabla f_i(\tilde{x}_i, \tilde{u}_i)| < L|(\tilde{x}_i, \tilde{u}_i) - (\tilde{x}_i, \tilde{u}_i)| \tag{6.26}$$

and

$$|\nabla r_N(x_N) - \nabla r_N(\tilde{x}_N)| < L|x_N - \tilde{x}_N| \tag{6.27}$$

We can now specify an algorithm the following way.

Maximum Principle Algorithm - Quadratic Control Penalty

Step 0 Choose $u_i^0 \in U_i$ and calculate x_{i+1}^0 from (6.4) for all i. Choose $\gamma_* > 0$. Let $k = 0$ and $\gamma = \gamma_*$

Step 1 Calculate p_i^k from (6.13) for $i = N, \ldots, 1$

Step 2 Repeat letting $\gamma = 2\gamma$ until the criterion in (6.23) is strictly concave with respect to u_i for all (x_i, p_{i+1}).

Step 3 Calculate u_i^* from (6.23) - (6.24) for $i = 0, \ldots, N-1$

Step 4 If $u_i^* = u_i^k$ for all i then stop, else go to Step 5.

Step 5 Let $u_i = u_i^*$ and calculate x_{i+1} from (6.4) for $i = 0, \ldots, N-1$

Step 6 If $\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) > \sum_{i=0}^{N-1} r_i(x_i^k, u_i^k) + r_N(x_N^k)$ then go to Step 7 else let $\gamma = 2\gamma$ and go to Step 3.

Step 7 Let $k = k + 1, u_i^k = u_i$ and $x_i^k = x_i$. Let $\gamma = \gamma_*$. Go to Step 1.

Proposition 6.1.4 *Consider the problem (6.3) - (6.7) and assume as in Proposition 6.1.3 and in addition that (6.25) - (6.27) hold. Then the conclusions of Proposition 6.1.3 hold for the algorithm above.*

Proof. We observe that the Lipschitz bound (6.25) - (6.27) and the elimination of x, see (6.8) - (6.9), assure that the function r in (6.15) may be underestimated by a quadratic function and indeed is so if γ is sufficiently large.

Define the function $F : R^{2Nm} \to R$ as in the proof of Proposition 6.1.3. If γ is sufficiently large then F underestimates r, i.e., $F(u^k; u^k) = r(u^k)$ and $F(u^k; u) \leq r(u)$ for all u feasible in (6.16). If the unique u^* maximizes $F(u^k; u)$ and $u^* \neq u^k$ then $r(u^k) = F(u^k; u^k) < F(u^k; u^*) \leq r(u^*)$, i.e., an improvement is attained at u^* relative to u^k.

The result then follows in the same way as in Proposition 6.1.3. □

Quadratic State Penalty

Now recall that the reason for using the quadratic term as above essentially is to compensate for the error introduced by approximating $\sum_{j=i}^{N} r_j$ by H_i. However, r_i is represented accurately, and only in the representation of $\sum_{j=i+1}^{N} r_j$ by $p_{i+1} f_i$ there may be an error.

In the second version we therefore relate the quadratic term more directly to where the approximation error emanates, viz., $p_{i+1} f_i$. Therefore the problem corresponding to (6.23) - (6.24) will be

$$\max_{u_i} [H_i(x_i^k, u_i, p_{i+1}^k) - \gamma \sum_{j=1}^{n} (f_i^j(x_i^k, u_i) - (x_{i+1}^j)^k)^2] \tag{6.28}$$

$$u_i \in U \tag{6.29}$$

Again a Lipschitz bound (6.25) - (6.27) will suffice to ensure that an underestimation is possible.

We can now specify an algorithm the following way.

Maximum Principle Algorithm - Quadratic State Penalty

Step 0 Choose $u_i^0 \in U_i$ and calculate x_{i+1}^0 from (6.4) for all i. Choose $\gamma_* > 0$. Let $k = 0$ and $\gamma = \gamma_*$

Step 1 Calculate p_i^k from (6.13) for $i = N, \ldots, 1$

Step 2 Repeat letting $\gamma = 2\gamma$ until the criterion in (6.23) is strictly concave with respect to u_i for all (x_i, p^{i+1}).

Step 3 Calculate u_i^* from (6.23) - (6.24) for $i = 0, \ldots, N-1$

Step 4 If $u_i^* = u_i^k$ for all i then stop, else go to Step 5.

Step 5 Let $u_i = u_i^*$ and calculate x_{i+1} from (6.4) for $i = 0, \ldots, N-1$

Step 6 If $\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) > \sum_{i=0}^{N-1} r_i(x_i^k, u_i^k) + r_N(x_N^k)$ then go to Step 7 else let $\gamma = 2\gamma$ and go to Step 3.

Step 7 Let $k = k + 1, u_i^k = u_i$ and $x_i^k = x_i$. Let $\gamma = \gamma_*$. Go to Step 1.

Proposition 6.1.5 *Consider the problem (6.3) - (6.7) and assume as in Proposition 6.1.3 and in addition that (6.25) - (6.27) hold. Then the conclusions of Proposition 6.1.3 hold for the algorithm above.*

Proof. The Lipschitz bound (6.25) - (6.27) and the elimination of x, see (6.8) - (6.9), assure that r may be underestimated using quadratic functions. The result then follows in the same way as in Proposition 6.1.3. □

In general it will be more difficult to solve (6.28) - (6.29) than (6.23) - (6.24). In particular if the Hamiltonian has some additively separable structure this is preserved in (6.23) - (6.24), while presumably lost in (6.28) - (6.29) if $n > 1$, see Example 6.1.1.

Example 6.1.1 *Let* $n = 2$, $m = 3$, $U_i = \{u_i \in R^m \mid \underline{u}_i^j \leq u_i^j \leq \overline{u}_i^j,\ j = 1,\ldots,m\}$, $x_{i+1}^1 = f_i^1(x_i, u_i) = x_i^1 - u_i^1 - u_i^3 + d_i^1$, $x_{i+1}^2 = f_i^2(x_i, u_i) = x_i^2 - u_i^2 + u_i^3 + d_i^2$, $r_i(x_i, u_i) = r_i^x(x_i) + r_i^{u1}(u_i^1) + r_i^{u2}(u_i^2) + r_i^{u3}(u_i^3)$. *This can be seen as an example of the water network case described in the introduction. With* $\gamma = 1$ *and thus* $\pi_{i+1}(x_{i+1}) = -(x_{i+1}^1)^2 - (x_{i+1}^2)^2$ *we specify (6.28) - (6.29) as follows:*

$$\begin{aligned}
&\max_{u_i}[r_i(x_i, u_i) - (x_i^1 - u_i^1 - u_i^3 + d_i^1 - (x_{i+1}^1)^k)^2 \\
&-(x_i^2 - u_i^2 + u_i^3 + d_i^2 - (x_{i+1}^2)^k)^2] \\
&\underline{u}_i^1 \leq u_i^1 \leq \overline{u}_i^1 \\
&\underline{u}_i^2 \leq u_i^2 \leq \overline{u}_i^2 \\
&\underline{u}_i^3 \leq u_i^3 \leq \overline{u}_i^3
\end{aligned}$$

As seen, this is not decomposable in u_i^j. *But if we specify* $\gamma = 0$ *and thus a linear* π_{i+1}, *viz.,* $\pi_{i+1}(x_{i+1}) = p_{i+1}x_{i+1}$, *the local problem decomposes into three independent problems:*

$$\begin{aligned}
&\max_{u_i^1}[r_i^{u1}(u_i^1) - p_{i+1}^1 u_i^1] \\
&\underline{u}_i^1 \leq u_i^1 \leq \overline{u}_i^1 \\
&\\
&\max_{u_i^2}[r_i^{u2}(u_i^2) - p_{i+1}^2 u_i^2] \\
&\underline{u}_i^2 \leq u_i^2 \leq \overline{u}_i^2 \\
&\\
&\max_{u_i^3}[r_i^{u3}(u_i^3) - p_{i+1}^1 u_i^3 + p_{i+1}^2 u_i^3] \\
&\underline{u}_i^3 \leq u_i^3 \leq \overline{u}_i^3
\end{aligned}$$

These are the subproblems to be solved in (6.22) - (6.21) in the Maximum Principle Algorithm of Section 6.1 with $\gamma = 0$. *Also the maximization in (6.22) - (6.21) with* $\gamma > 0$ *and (6.23) - (6.24) permits this decomposition, since the penalty* $\sum_{j=1}^m (u_i^j - (u_i^j)^k)^2$ *is additively separable.* □

All algorithms in this Section (in this Chapter, actually) have at most a linear rate of convergence, since they rely on the most recent gradient information only. The rate of convergence will depend on the eigenvalue structure (cf. Luenberger (1989) p. 220, Part III), in particular the relationship

$$\left(\frac{A-a}{A+a}\right) \tag{6.30}$$

where A is the largest and a is the smallest of the eigenvalues in $\nabla^2 r(u^*)$, where r is defined in (6.15). Thus, there will not in terms of rate of convergence be any essential difference between the algorithms presented here. A value close to 1 indicates rapid convergence. The matrix $\nabla^2 r(u^*)$

may in the scalar case ($n = m = 1$) be seen to be constituted of the following submatrices

$$
\begin{aligned}
\frac{\partial^2 r}{\partial u_k \partial u_k} = \quad & \nabla^2_{uu} r_k \\
+ \quad & \nabla_u f'_k \nabla^2_{xx} r_{k+1} \nabla_u f_k + \nabla_x r_{k+1} \nabla^2_{uu} f_k \\
+ \quad & \nabla_u f'_k \nabla_x f'_{k+1} \nabla^2_{xx} r_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& \quad + \nabla_u f'_k \nabla_x r_{k+2} \nabla^2_{xx} f_{k+1} \nabla_u f_k + \nabla_x r_{k+2} \nabla_x f_{k+1} \nabla^2_{uu} f_k \\
+ \quad & \nabla_u f'_k \nabla_x f'_{k+1} \nabla_x f'_{k+2} \nabla^2_{xx} r_{k+3} \nabla_x f_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& \quad + \nabla_u f'_k \nabla_x f'_{k+1} \nabla_x r_{k+3} \nabla^2_{xx} f_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& \quad + \nabla_u f'_k \nabla_x r_{k+3} \nabla_x f_{k+2} \nabla^2_{xx} f_{k+1} \nabla_u f_k \\
& \quad + \nabla_x r_{k+3} \nabla_x f_{k+2} \nabla_x f_{k+1} \nabla^2_{uu} f_k \\
+ & \\
\vdots & \\
+ \quad & \nabla_u f'_k \nabla_x f'_{k+1} \cdots \nabla_x f'_{N-1} \nabla^2_{xx} r_N \nabla_x f_{N-1} \cdots \nabla_x f_{k+1} \nabla_u f_k \\
& \quad + \\
& \quad \vdots \\
& \quad + \nabla_x r_N \nabla_x f_{N-1} \cdots \nabla_x f_{k+1} \nabla^2_{uu} f_k
\end{aligned}
\tag{6.31}
$$

and for $j < k$:

$$
\begin{aligned}
\frac{\partial^2 r}{\partial u_k \partial u_j} = \quad & \nabla_u f'_j \nabla_x f'_{j+1} \nabla_x f'_{j+2} \cdots \nabla_x f'_{k-2} \nabla_x f'_{k-1} \nabla^2_{xu} r_k \\
+ \quad & \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_k \nabla^2_{xx} r_{k+1} \nabla_u f_k \\
& \quad + \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_{k-1} \nabla^2_{xu} f_k \nabla_x r_{k+1} \\
+ \quad & \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_{k+1} \nabla^2_{xx} r_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& \quad + \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_k \nabla^2_{xx} f_{k+1} \nabla_x r_{k+2} \nabla_u f_k \\
& \quad + \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_{k-1} \nabla^2_{xu} f_k \nabla_x f_{k+1} \nabla_x r_{k+2} \\
+ & \\
\vdots & \\
+ \quad & \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla^2_{xx} r_N \nabla_x f_{N-1} \cdots \\
& \quad + \\
& \quad \vdots \\
& \quad + \nabla_u f'_j \nabla_x f'_{j+1} \cdots \nabla_x f'_{N-2} \nabla^2_{xu} f_{N-1} \nabla_x r_N
\end{aligned}
\tag{6.32}
$$

It is seen that in general the relationship (6.30) will not be close to 1, and rapid convergence cannot be expected. In particular it is seen that the $\nabla_x f_i$'s are important, as the product of up to $N-1$ such matrices are involved. In the scalar case, for instance, if $\nabla^2_{xx} r_N < 0$ and if $\nabla_x f_i > 1$ for all i this will imply decreasing speed of convergence as N increases.

6.2 Hamiltonians, Gradients and Projections

The essential working mechanisms of the previous algorithms may be seen to be three: an approximation of the gradient of $\sum_{j=i}^{N} r_j$, strict concavity and stepsize control. We here discuss the consequence of the gradient approximation.

The importance is clear from the convergence analysis. Essentially, at stage i we have the Hamiltonian given as a combination of the local criterion r_i and the gradient with respect to u_i of $\sum_{j=i+1}^{N} r_j$.

It seems that the difference between taking the Hamiltonian and taking the gradient should not be essential in terms of rate of convergence. Thus, we arrive naturally at gradient ideas.

A gradient projection idea may be specified as follows. Consider at stage i the point $(u_i^k + \nabla_u H_i(x_i^k, u_i^k, p_{i+1}^k)')$. This may be interpreted stagewise as the previous iteration point u_i^k plus the partial derivative with respect to u_i of $\sum_{j=0}^{N} r_j$. This point is to be projected onto the feasible control set U_i. The projection u_i^* satisfies

$$u_i^* = \arg\max_{u_i}[-\|(u_i^k + \nabla_u H_i(x_i^k, u_i^k, p_{i+1}^k)' - u_i)\|] \tag{6.33}$$

$$u_i \in U_i \tag{6.34}$$

If U_i is convex, compact and nonempty then u_i^* is uniquely defined from (6.33) - (6.34).

The difficulty in finding u_i^* from (6.33) - (6.34) depends on the set U_i. A particularly attractive case is where $U_i = \{u_i \in R^m \mid \underline{u}_i \leq u_i \leq \overline{u}_i\}$. Here the projection is simply found as follows. Let $(u_i^p)^j = (u_i^k + \nabla_u H_i(x_i^k, u_i^k, p_{i+1}^k)')^j$ for $j = 1, \ldots, m$. Then

$$\begin{aligned}(u_i^*)^j &= \max\{\underline{u}_i^j, \min\{\overline{u}_i^j, (u_i^p)^j\}\} = \min\{\overline{u}_i^j, \max\{\underline{u}_i^j, (u_i^p)^j\}\} \\ &= \begin{cases} \underline{u}^j & \text{if } (u_i^p)^j \leq \underline{u}^j \\ (u_i^p)^j & \text{if } \underline{u}^j \leq (u_i^p)^j \leq \overline{u}^j \\ \overline{u}^j & \text{if } \overline{u}^j \leq (u_i^p)^j \end{cases}\end{aligned} \tag{6.35}$$

This was applied in e.g. Papageorgieu (1985).

Gradient Projection Algorithm

Step 0 Choose $u_i^0 \in U_i$ and calculate x_{i+1}^0 from (6.4) for all i. Let $k = 0$ and $\alpha = 1$

Step 1 Calculate p_i^k from (6.13) for $i = N, \ldots, 1$

Step 2 Calculate u_i^* from (6.33) - (6.34) for $i = 0, \ldots, N-1$

Step 3 If $u_i^* = u_i^k$ for all i then stop, else go to Step 4.

Step 4 Calculate $u_i = \alpha u_i^* + (1-\alpha)u_i^k$ and x_{i+1} from (6.4) for $i = 0, \ldots, N-1$

Step 5 If $\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) > \sum_{i=0}^{N-1} r_i(x_i^k, u_i^k) + r_N(x_N^k)$ then go to Step 6 else let $\alpha = \frac{1}{2}\alpha$ and go to Step 4.

Step 6 Let $k = k+1, u_i^k = u_i$ and $x_i^k = x_i$. Let $\alpha = 1$. Go to Step 1.

Proposition 6.2.1 *Consider the problem (6.3) - (6.7) and assume that for all i r_i and f_i are continuously differentiable. Assume that U_i are given as $\{u_i \in R^m \mid g_i(u_i) \leq 0, h_i(u_i) = 0\}$ where g_i are convex and h_i linear and a constraint qualification holds. Assume that the set $\{u \in R^{Nm} \mid \sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) \geq \sum_{i=0}^{N-1} r_i(x_i^0, u_i^0) + r_N(x_N^0), u_i \in U_i, x_{i+1} = f_i(x_i, u_i)\}$ is compact and*

nonempty. Then any accumulation point for the algorithm satisfies the weak maximum principle of Proposition 3.4.3.

If in addition for all i r_i is concave and strictly concave with respect to u_i and f_i is linear then for any $\gamma \geq 0$ the algorithm converges to an optimal solution of the problem where the maximum principle of Proposition 3.4.3 is satisfied.

Proof. The projection (6.33) - (6.34) performed stagewise is the same as the projection of $(u^k + \nabla_u \sum_{i=0}^{N} r_i)$ on $U_0 \times \ldots \times U_{N-1}$, cf. Proposition 6.1.2. At all steps of the algorithm the values generated are unique since $U = U_0 \times \ldots \times U_{N-1}$ is convex and all functions are continuously differentiable. Moreover any value generated is a continuous function of the other values involved; thus, u_i from (6.33) - (6.34) is a continuous function of (x_i, p_{i+1}), x_{i+1} from (6.4) is a continuous function of u_i and p_i from (6.13) is a continuous function of (x_i, u_i, p_{i+1}). Considering the algorithm as a mapping this mapping is then point-to-point and continuous and therefore closed. All iteration points are contained in a compact set.

If $u^k \neq u^*$ then the criterion will increase. If $u^k = u^*$ then a stationarity condition is fulfilled and therefore also the weak maximum principle is fulfilled, cf. Proposition 1.3.4. From Luenberger (1989) pp. 187 - 188 the first part then follows. Under the assumptions of the second part the optimal solution is unique and satisfies the maximum principle, cf. Proposition 1.3.5. From Luenberger (1989) pp. 187 - 188 the second part then follows. □

The strength of the above algorithms is the simplicity. In particular the possibility of exploiting any separability structure at stage i may be advantageous, cf. the discussion after Proposition 6.1.5, but also in many other cases it will be simpler to perform the projection (6.33) than to maximize the Hamiltonian, in particular if a simple structure of U_i may be exploited as in (6.35).

6.3 State Constraints and Feedback Strategies

A weakness of the algorithms above is that they cannot handle constraints which involve state variables. The algorithms rely on the calculation of partial derivatives of the criterion function with respect to the control u_i in order to analyze the consequences of a small change δu_i in u_i. Thus it is essential that u_i can move freely, while all other u_j, $j \neq i$, are kept constant. If state constraints are present, this need not be possible. In particular it is desirable to be able to treat an end point constraint. We discuss how this may be handled and show that we arrive naturally at the definition of (feedback) strategies.

Assume in relation to the formulation (6.3) - (6.7) that the constraint $u_{N-1} \in U_{N-1}$ and an additional end constraints $x_N \in V_N$ may be represented by

$$g_{N-1}(u_{N-1}) \leq 0 \tag{6.36}$$

$$h_{N-1}(x_{N-1}, u_{N-1}) = 0 \tag{6.37}$$

This is particular covers the case with a fixed end point, $x_N = \underline{x}_N$. This is seen by using the dynamic equation to yield

$$f_{N-1}(x_{N-1}, u_{N-1}) - \underline{x}_N = 0 \tag{6.38}$$

which is similar to (6.37).

Now for a given $x_{N-1} = x_{N-1}^k$ we may find u_{N-1} as

$$u_{N-1} = \arg\max_{u_{N-1}} [r_{N-1}(x_{N-1}, u_{N-1}) + r_N(f_{N-1}(x_{N-1}, u_{N-1}))] \tag{6.39}$$

subject to (6.36) - (6.37).

In addition to u_{N-1} from (6.39) it is required to find the KKT multipliers $\lambda \in R^k$ and $\mu \in R^\ell$ corresponding to the problem (6.36) - (6.37), (6.39).

Now the algorithms on pages 171, 173, and 174 of Section 6.1 are applied with the following two modifications.

In the forwards recursions, u_{N-1} is always calculated from (6.36) - (6.37), (6.39). The backwards recursions are always started from $i = N-1$ with

$$p_{N-1}^k = \nabla_x [r_{N-1}(x_{N-1}^k, u_{N-1}) + r_N(f_{N-1}(x_{N-1}^k, u_{N-1})) \\ -\lambda g_{N-1}(u_{N-1}) - \mu h_{N-1}(x_{N-1}^k, u_{N-1})] \tag{6.40}$$

Proposition 6.3.1 *Consider the problem (6.3) - (6.7), (6.38). Assume as in the first part of the previous Propositions 6.1.3, 6.1.4 and 6.1.5, and in addition that at stage $N-1$ the constraint functions g_{N-1} and h_{N-1} are continuously differentiable, and that the solution u_{N-1} and the multipliers λ, μ to (6.36) - (6.37), (6.39) are unique at all iterations. Then the first results also holds for the modified algorithms. If the assumptions of the last part of the previous Propositions 6.1.3, 6.1.4 and 6.1.5 hold and if in addition to the above g_{N-1} is convex and h_{N-1} linear, then the last result also holds for the modified algorithms.*

Proof. From Proposition 2.7.4 it follows that $ub_{N-1}^N(x_{N-1}, \underline{x}_N)$ as a function of x_{N-1} is continuously differentiable at all the points considered in the iterations with $\nabla_{x_{N-1}} ub_{N-1}^N(x_{N-1}, \underline{x}_N) = p_{N-1}$ where p_{N-1} is defined in (6.40). Therefore essentially the modified algorithms operate on a problem like (6.3) - (6.7) but with N one less and $r_{N-1}(x_{N-1}) = ub_{N-1}^N(x_{N-1}, \underline{x}_N)$. The results therefore follow from the previous Propositions 6.1.3, 6.1.4 and 6.1.5. □

A second way to handle the end constraints is to apply a reduced gradient idea. Thus, assume e.g. that $n = m = 1$ and that the conditions (6.36) - (6.37) may be written

$$u_{N-1} \leq \overline{u}_{N-1} \tag{6.41}$$

$$-u_{N-1} \leq -\underline{u}_{N-1} \tag{6.42}$$

$$x_{N-1} + u_{N-1} = \underline{x}_N \tag{6.43}$$

Then, if

$$\underline{u}_{N-1} < \underline{x}_N - x_{N-1} < \overline{u}_{N-1} \tag{6.44}$$

we may take

$$u_{N-1} = \underline{x}_N - x_{N-1} \tag{6.45}$$

and use this to obtain a problem with "free" end point and one stage less. The new terminal criterion function will then be

$$r_{N-1}(x_{N-1}, \underline{x}_N - x_{N-1}) + r_N(\underline{x}_N) \tag{6.46}$$

This will work fine, as long as (6.44) is fulfilled. However, for all other x_{N-1} there will be problems.

We observe that this way of defining u_{N-1} will in this particular case yield exactly the same unique u_{N-1} as (6.39) does. Moreover those values of x_{N-1} for which (6.44) is not fulfilled and for

which the reduced gradient idea therefore will have problems are exactly the same values as those for which the definition (6.39) will not yield a unique value for (λ, μ).

We are therefore basically in the same type of difficulties as in relation to the analysis in Section 2.5 and Section 2.7.

In Section 2.5 we pointed to stage aggregation as a way out of these difficulties. This implies that if (6.44) is not fulfilled we should consider stages N, $N-1$ and $N-2$ together. In line with (6.39) we would then determine $(u_{N-2}, x_{N-1}, u_{N-1})$ as

$$(u_{N-2}, x_{N-1}, u_{N-1}) = \arg \max_{u_{N-2}, x_{N-1}, u_{N-1}} [r_{N-2}(x_{N-2}, u_{N-2}) + r_{N-1}(x_{N-1}, u_{N-1}) + r_N(f_{N-1}(x_{N-1}, u_{N-1}))] \tag{6.47}$$

subject to the local constraints, end constraints and the dynamic equation. With this we also determine the KKT multipliers, such that p_{N-2} can be determined in a way similar to (6.40).

It is not clear in advance how many stages that will have to be aggregated. Therefore the determination of this must be part of the solution algorithm. See Example 6.3.1.

Example 6.3.1 *Consider the same problem as i Example 1.5.1 on page 38: Let* $n = 1$, $m = 1$, $N = 24$, $\underline{x}_0 = \underline{x}_N = 25$, $\underline{x}_i = -\infty$ *and* $\overline{x}_i = \infty$, $i = 1, \ldots, 23$; $\underline{u}_i = 0$, $\overline{u}_i = 10$, $i = 0, \ldots, 23$; $r_i(u_i) = -u_i^2 - \gamma_i u_i$, *where* $\gamma_i = 5$ *for* $0 \leq i \leq 23$; *and* $x_{i+1} = x_i + u_i - d_i$ *where* $d_i = 5 + 2\sin(2\pi i/24)$, $i = 0, \ldots, 23$.

For this problem there is no difficulty with the end constraint because $\underline{u}_{23} < u_{23}^* = 5 < \overline{u}_{23}$. *However, change now the problem, such that* $\gamma_i = 1$ *for* $0 \leq i \leq 17$ *and* $\gamma_i = 20$ *for* $18 \leq i \leq 23$. *Now the optimal solution has* $u_i^* = 0$ *for* $18 \leq i \leq 23$. *Therefore, if* $x_{N-1} = x_{N-1}^* = 29.48$ *the condition (6.44) is not fulfilled because* $\underline{u}_{N-1} = \underline{x}_{N-1} - x_{N-1}^* + d_{N-1}$. *Similarly difficulty will be found al all stages* $18 \leq i \leq 23$. *In order to attain a unique* p_i *we must take* $i \leq 17$ *and stages* $j = i$ *to* $j = 23$ *must be taken together, cf. (6.47). With e.g.* $i = 17$ *and* $x_{17} = x_{17}^* \approx 49.47$ *the solution of the problems similar to (6.47) from* $i = 17$ *will yield the unique* $p_{17} \approx 14.33$. □

Finally, for a given x_{N-1} it is not sure that (6.36) - (6.37) admits a feasible solution. This may also be the case if stages have been aggregated. In this situation it may be necessary to introduce artificial variables, cf. the discussion in Section 2.5.

It follows from the above that u_{N-1} will depend on x_{N-1}, cf. (6.39) or (6.45); this may be indicated as $u_{N-1}^*(x_{N-1})$. Thus, we have naturally arrived at the specification of $u_{N-1}^*(.)$ through a *(feedback) strategy.* i.e., a function $u_{N-1}^*(.) : R^n \to R^m$.

As Example 6.3.1 shows, the end constraint will not only necessitate the definition of the strategy $u_{N-1}^*(.)$ but in general strategies $u_i^*(.)$ for all i.

Therefore also the above extends immediately to state dependent constraints at any stage i.

Example 6.3.1 also shows that it is not immediate what variables u_i^j to associate with what state dependent constraint. Obviously a feedback strategy $u_i^*(.)$ will relate to a state dependent constraint at a later stage j, $i < j$, but j need not be equal to $i + 1$. In the terminology of linear programming we may say that a basic variable at stage i may relate to a constraint at any stage later than i, not only to state $i + 1$. Obviously, this is fatal for the stagewise approach. (Cf. also the discussion at the end of Section 9.3.)

6.4 The Generalized Maximum Principle

In this section we consider an OCP which has more general constraints than (6.5). Thus, the OCP may be stated as

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{6.48}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{6.49}$$

$$g_i(x_i, u_i) \leq 0 \tag{6.50}$$

$$h_i(x_i, u_i) = 0 \tag{6.51}$$

$$x_0 = \underline{x}_0 \tag{6.52}$$

Thus, we admit in this section state-dependent constraints at all stages.

End constraints are assumed included in (6.50) - (6.51), cf. the discussion around (6.36) - (6.38).

We introduce the generalized Hamiltonian

$$H_i(x_i, u_i, \pi_{i+1}) = r_i(x_i, u_i) + \pi_{i+1}(f_i(x_i, u_i)) \tag{6.53}$$

where $\pi_{i+1} : R^n \rightarrow R$ is not necessarily linear nor smooth.

In close analogy to the analysis in the previous section, we shall at each stage solve the following problem

$$\max_{u_i}[H_i(x_i, u_i, \pi_{i+1})] \tag{6.54}$$

$$g_i(x_i, u_i) \leq 0 \tag{6.55}$$

$$h_i(x_i, u_i) = 0 \tag{6.56}$$

The adjoint equations will take the following form if all π_i are smooth:

$$\nabla\pi_i(x_i) = \nabla_x H_i(x_i, u_i, \pi_{i+1}) - \lambda_i \nabla_x g_i(x_i, u_i) - \mu_i \nabla_x h_i(x_i, u_i). \tag{6.57}$$

Here it is assumed that π_i and π_{i+1} are continuously differentiable, and that λ_i and μ_i are unique KKT-multipliers, found at the solution of (6.54) - (6.56). Again, suitable conditions, to be specified below, are imposed on π_N. See also Section 3.4.

Thus, the algorithms construct a solution (x^*, u^*, π^*), which satisfies the generalized maximum principle, viz., the maximum part (6.54) - (6.56) and the adjoint equations part (6.57) along the trajectory, cf. e.g. Proposition 3.4.4.

Later in this section we shall introduce non-smooth π, and the necessary modifications in the adjoint equations.

Compared to the previous section we see that there we actually worked with a special case, viz., with a linear π_i, $\pi_i(x_i) = p_i x_i$.

The advantages of the generalization of the Hamiltonian are that we can treat state constraints at all stages. The cost of this is that the maximization of the Hamiltonian is more difficult. Particularly, if there is some decomposability with respect to the control, this will be lost by the introduction of the term $\pi_{i+1}(f_i(x_i, u_i))$ if π_{i+1} is nonlinear. Contrary, any decomposability in f_i is preserved in the linear case with $\pi_{i+1}(f_i(x_i, u_i)) = p_{i+1} f_i(x_i, u_i)$ as in (6.22) - (6.21) and (6.23) - (6.24). A second disadvantage is that the maximization is to take place both in the forwards and in the backwards directions.

Strategy and lower support

We now define a local strategy $u_i^\diamond(.)$ at stage i as a mapping $R^n \to R^m$ of state x_i into control u_i, see also the previous Section 6.3. Thus we have $u_i^\diamond = u_i(x_i)^\diamond$. The local strategy is defined for all those x_i, for which $\{u_i \in R^m \mid g_i(x_i, u_i) \leq 0, h_i(x_i, u_i) = 0\} \neq \emptyset$.

We define a terminating strategy from stage i as a mapping of x_i into $u_i^\diamond$, and, through the dynamic equation and the sequence of local strategies at stage $i+1, i+2, \ldots, N-1$, into $(x_{i+1}, u_{i+1}^\diamond, \ldots, x_{N-1}, u_{N-1}^\diamond, x_N)$. The terminating strategy from stage i is defined for all these x_i, for which it is possible to find a trajectory up to stage N.

We define a lower support at stage i at a point $\underline{x}_i$, relative to a terminating strategy from stage i, as a function $\pi_i : R^n \to R$ such that

$$\pi(x_i) - \pi_i(\underline{x}_i) \leq [\sum_{j=i}^{N-1} r_j(x_j, u_j^\diamond(x_j)) + r_N(x_N)] \tag{6.58}$$

$$-\{\sum_{j=i}^{N-1} r_j(x_j, u_j^\diamond(x_j)) + r_N(x_N)\} \tag{6.59}$$

Here the expression in [ ] is the criterion value found by using the terminating strategy starting at x_i, while the expression in { } is the value found by using the terminating strategy starting at $\underline{x}_i$. Observe that RUB_i may serve as lower supports.

In the sequel we define the terminating strategy as follows. At all stages the local strategy for a given π_{i+1} is defined as

$$u_i^\diamond(x_i) = \arg\max_{u_i}[H_i(x_i, u_i, \pi_{i+1})] \tag{6.60}$$

$$g_i(x_i, u_i) \leq 0 \tag{6.61}$$

$$h_i(x_i, u_i) = 0 \tag{6.62}$$

Quadratic Support

As in Section 6.1 we may work with quadratic π_i. Thus, for a given x_N^k we calculate

$$p_N = \nabla r_N(x_N^k) \tag{6.63}$$

and then

$$\pi_N(x_N) = p_N x_N - \gamma \sum_{j=1}^{n} (x_N^j - (x_N^j)^k)^2 \tag{6.64}$$

Here γ is a positive constant. At stage i we calculate u_i, λ_i and μ_i for a given x_i^k from (6.60) - (6.62). Then we define H_i as in (6.53) and

$$p_i = \nabla_x(H_i(x_i^k, u_i, \pi_{i+1}) - \lambda_i g_i(x_i^k, u_i) - \mu h_i(x_i^k, u_i)) \tag{6.65}$$

and finally let

$$\pi_i(x_i) = p_i x_i - \gamma \sum_{j=1}^{n} (x_i^j - (x_i^j)^k)^2 \tag{6.66}$$

Compare with the quadratic state penalty in (6.28).

The adjoint equations are

$$\begin{aligned} \nabla\pi_i(x_i) &= \nabla_x H_i(x_i, u_i, \pi_{i+1}) \\ -\lambda_i \nabla_x g_i(x_i, u_i) - \mu_i \nabla_x h_i(x_i, u_i) & \end{aligned} \tag{6.67}$$

Algorithm GMP - Quadratic Support

Step 0 Choose u_i^0 and calculate x_{j+1}^0 from the dynamic equation (6.4) recursively forwards, $i = 0, \ldots, N-1$. Choose $\gamma_* > 0$

Step 1 Calculate π_N from (6.64) and then recursively backwards, $i = N-1, \ldots, 0$, (u_i, λ_i, μ_i) from (6.60) - (6.62) and π_i from (6.66).

Step 2 Calculate recursively forwards, $i = 0, \ldots, N-1, u_i$ from (6.60) - (6.62) and x_{i+1} from the dynamic equation (6.4).

Step 3 If $u_i = u_i^k$ for all i then stop.

Step 4 If $\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) > \sum_{i=0}^{N-1} r_i(x_i^k, u_i^k) + r_N(x_N^k)$ then go to Step 5 else go to Step 6.

Step 5 Let $x_i^{k+1} = x_i$, $u_i^{k+1} = u_i$, $k = k+1$, $\gamma = \gamma_*$ and go to Step 1.

Step 6 Choose $\gamma = 2\gamma$ and go to Step 1.

Observe that maximization is performed in both the forwards (Step 2) and the backwards (Step 1) directions. Thus, the feedback strategy $u_i^\diamond(.)$ is not expressed explicitly but only indirectly as the solution to an optimization problem ((6.60) - (6.62)).

Proposition 6.4.1 *Consider the problem (6.48) - (6.52) and assume that for all i r_i, f_i, g_i, and h_i are continuously differentiable and that the criterion-to-go satisfies a Lipschitz bound on the gradients.*

Assume that (x^0, u^0) is feasible and that the set $\{u \in R^{Nm} \mid \sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) \geq \sum_{i=0}^{N-1} r_i(x_i^0, u_i^0) + r_N(x_N^0), g_i(x_i, u_i) \leq 0, h_i(x_i, u_i) = 0, x_{i+1} = f_i(x_i, u_i)\}$ is compact and nonempty. Assume that at all points a constraint qualification is fulfilled and that (6.60) - (6.62) yields a unique solution and unique multipliers (λ_i, μ_i).

Then any accumulation point for the algorithm satisfies the generalized maximum principle of Proposition 3.4.6.

If in addition for all i r_i is concave and strictly concave with respect to u_i and f_i is linear then the algorithm converges to an optimal solution of the problem where the maximum principle is satisfied.

Proof. We start with the following

Lemma *Let there be given a terminating strategy from stage i. Assume $\underline{x}_{i+1} = f_i(\underline{x}_i, u_i^\diamond(x_i))$. Let π_{i+1} be a lower support at $\underline{x}_{i+1}$ at stage $i+1$ to the terminating strategy. Let*

$$\begin{aligned} u_i^* &= \arg\max_{u_i} [H_i(\underline{x}_i, u_i, \pi_{i+1})] \\ g_i(\underline{x}_i, u_i) &\leq 0 \\ h_i(\underline{x}_i, u_i) &= 0 \end{aligned}$$

Then

$$r_i(\underline{x}_i, u_i^*) + [\sum_{j=i+1}^{N-1} r_j(x_j, u_j^\diamond(x_j)) + r_N(x_N)]$$
$$\geq r_i(\underline{x}_i, u_i^\diamond(\underline{x}_i)) + \{\sum_{j=i+1}^{N-1} r_j(x_j, u_j^\diamond(x_j)) + r_N(x_N)\}$$

If $u_i^\diamond(\underline{x}_i)$ is not maximizing, then the inequality is sharp.

Here the expression in [ ] was found using the terminating strategy from stage $i+1$ starting at $x_{i+1} = f_i(\underline{x}_i, u_i^*)$, while the expression in $\{\}$ was found using the terminating strategy from stage $i+1$ starting at $\underline{x}_{i+1} = f_i(\underline{x}_i, u_i^\diamond(\underline{x}_i))$.

Proof of Lemma. Optimality of u_i^* means

$$r_i(\underline{x}_i, u_i^*) + \pi_{i+1}(f_i(\underline{x}_i, u_i^*)) \geq r_i(\underline{x}_i, u_i) + \pi_{i+1}(f_i(\underline{x}_i, u_i^\diamond(\underline{x}_i)))$$

and the inequality is sharp if $u_i^\diamond(\underline{x}_i)$ is not optimal.

Now use that π_{i+1} was assumed a lower support at $\underline{x}_{i+1} = f_i(\underline{x}_i, u_i(\underline{x}_i)^\diamond))$. Considering this at the point $x_{i+1} = f_i(\underline{x}_i, u_i^*)$, and combining with the expression above yields

$$r_i(\underline{x}_i, u_i^*) + \pi_{i+1}(f_i(\underline{x}_i, u_i^*)) - \pi_{i+1}(x_{i+1}) + \pi_{i+1}(\underline{x}_{i+1})$$
$$\geq r_i(\underline{x}_i, u_i^\diamond(\underline{x}_i)) + \pi_{i+1}(f_i(\underline{x}_i, u_i^\diamond(\underline{x}_i))) - [\,] + \{\}.$$

Here [ ] and $\{\}$ have the same meaning as in the Lemma. After reduction and rearrangement we get the stated result. This concludes the proof of the Lemma.

For the proof of the Proposition we observe that if $u^k = u^*$ then by the construction (6.64) - (6.66) it follows that the generalized maximum principle is fulfilled. In fact, in the backwards direction the adjoint relation is fulfilled at stage i at x_i^k by the construction of π_i. If $u^k \neq u^*$ and the criterion does not increase then γ will be increased, cf. Steps 4 and 6. Due to the continuous differentiability, Lipschitz bounds and uniqueness there exists a lower support of the form (6.64) - (6.66), cf. the Lemma.

Therefore eventually γ will be so big that π_1 is a lower support. By the Lemma an increase will then be attained in the criterion function. At all steps of the algorithm the values generated are unique and depend continuously of the other values involved. Considering the algorithm as a mapping this mapping is then point-to-point and continuous and therefore closed. All iteration points are contained in a compact set. From Luenberger (1989) pp. 187 - 188 the result then follows. □

Non-Smooth π

Now consider the situation where at stage i the maximizing u_i^* and/or the associated (λ_i, μ_i) in (6.60) - (6.62) are not unique. In this case the upper boundary is not necessarily smooth, cf. Section 2.7, and the above procedure need not work. We may in this case proceed as follows.

Consider the definition of an upper boundary $B_i : R^n \to R$ as follows:

$$B_i(x_i^k) = \max_{u_i}[H_i(x_i^k, u_i, \pi_{i+1})] \tag{6.68}$$
$$g_i(x_i^k, u_i) \leq 0 \tag{6.69}$$
$$h_i(x_i^k, u_i) = 0 \tag{6.70}$$

Assume that all functions in the above problem (6.68) - (6.70) are Lipschitz continuous, that the optimal solution u_i^* is unique and that a constraint qualification is satisfied. We do not assume that (λ_i, μ_i) is unique, but rather is contained in the set $KKT(u_i^*)$. This set is polyhedral and compact (Gauvin (1977)). Under these assumptions B_i is Lipschitz with lower Dini directional derivative $D(B_i; x_i; s)$ (cf. e.g. Gauvin and Debeau (1982), Clarke (1983) p. 242) in the direction $s \in R^n$ given as

$$\begin{aligned} D(B_i; x_i^k; s) = \quad & \min_{\lambda_i, \mu_i}[(\nabla_x(H_i(x_i^k, u_i^*, \pi_{i+1}) \\ & -\lambda_i g_i(x_i^k, u_i^*) - \mu_i h_i(x_i^k, u_i^*)))s] \qquad (6.71) \\ & (\lambda_i, \mu_i) \in KKT(u_i^*) \qquad (6.72) \end{aligned}$$

Now construct a function $\tilde{\pi}_i : R^n \to R$ which has the same directional derivatives as B_i. It follows that $\tilde{\pi}_i$ may be specified as

$$\begin{aligned} \tilde{\pi}_i(x_i) = \quad & \max_{\alpha_i \in R}[\alpha_i] \qquad (6.73) \\ & \alpha_i \leq \min_{(\lambda_i, \mu_i) \in KKT(u_i^*)} \qquad (6.74) \\ & [(\nabla_x(H_i(x_i, u_i, \pi_{i+1}) - \lambda_i g_i(x_i, u_i^*) - \mu_i h_i(x_i, u_i^*)))(x_i - x_i^k)] \end{aligned}$$

Since $KKT(u_i^*)$ is polyhedral and compact, a solution for (λ_i, μ_i) in (6.71) - (6.72) and (6.73) - (6.74) is obtained at an extreme point. There is a finite number E_i of extreme points, and (6.74) may therefore be substituted by

$$\begin{aligned} \alpha_i \leq \quad & (\nabla_x(H_i(x_i^k, u_i^*, \pi_{i+1}) \\ & -\lambda_i^e g_i(x_i^k, u_i^*) - \mu_i^e h_i(x_i^k, u_i^*)))(x_i - x_i^k), \qquad (6.75) \\ & e = 1, \ldots, E_i \end{aligned}$$

Finally we may specify the lower support π_i as

$$\pi_i(x_i) = \tilde{x}_i(x_i) - o_i(x_i) \qquad (6.76)$$

Here, the function $o_i : R^n \to R$ satisfies $o_i(x_i^k) = 0$ and $o_i(x_i) \geq 0$. We shall assume that $o_i(x_i - x_i^k) / \| x_i - x_i^k \| \to 0$ as $\| x_i - x_i^k \| \to 0$.

Also if u_i^* is non-unique this specifies a lower support for an arbitrary optimal u_i^*. This is because the Dini directional derivative D of B_i is in this case specified as

$$\begin{aligned} D(B_i; x_i^k; s) = \quad & \max_{u_i^*}[\min_{\lambda_i, \mu_i}[(\nabla_x(H_i(x_i^k, u_i^*, \pi_{i+1}) \qquad (6.77) \\ & -\lambda_i g_i(x_i^k, u_i^*) - \mu_i h_i(x_i^k, u_i^*)))s]] \end{aligned}$$

subject to $u_i^* \in U_i^*$ where U_i^* is the set of optimal solutions in (6.68) - (6.70), and subject to (6.72) for each individual $u_i^* \in U_i^*$.

With this approach the problem (6.60) - (6.62) must be replaced by

$$\begin{aligned} & u_i^\diamond(x_i) = \qquad (6.78) \\ & \arg \max_{u_i, \alpha_{i+1}} [r_i(x_i^k, u_i) + \alpha_{i+1} - o_{i+1}(f_i(x_i^k, u_i)] \\ & g_i(x_i^k, u_i) \leq 0 \qquad (6.79) \end{aligned}$$

$$h_i(x_i^k, u_i) = 0 \tag{6.80}$$

$$\alpha_{i+1} \leq (\nabla_x(H_i(x_i^k, u_i^*, \pi_{i+1}) \\ -\lambda_i^e g_i(x_i^k, u_i^*) - \mu_i^e h_i(x_i^k, u_i^*)))(f_i(x_i^k, u_i) - x_{i+1}^k), \\ e = 1, \ldots, E_{i+1} \tag{6.81}$$

For such π_i, the adjoint relations at stage i are

$$0 \in \partial_x(H_i(x_i^*, u_i^*, \pi_{i+1}^*) - \pi_i^*(x_i^*) - \lambda_i^* g_i(x_i^*, u_i^*) - \mu_i^* h_i(x_i^*, u_i^*)), \\ i = 1, \ldots, N-1 \tag{6.82}$$

cf. Proposition 3.4.5.

If u_i^* is not unique, i.e., U_i^* contains more than one point, then by taking an arbitrary $u_i^* \in U_i^*$ (i.e., using (6.71) for this u_i^*, not (6.77)) the function π_i constructed in (6.73) - (6.76) will be a lower support, however, it will not necessarily fulfill the adjoint equation (6.82).

Assuming that o_i may be chosen quadratic the algorithm is specified similarly to the one above. In particular observe that maximization is performed in both the forwards (Step 2) and the backwards (Step 1) directions in order to implement the feedback strategy and to determine the adjoint relations, respectively.

Proposition 6.4.2 *Assume that all functions r_i, f_i, g_i and h_i are Lipschitz continuous. Assume that (x^0, u^0) is feasible and that the set $\{u \in R^{Nm} \mid \sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) \geq \sum_{i=0}^{N-1} r_i(x_i^0, u_i^0) + r_N(x_N^0), g_i(x_i, u_i) \leq 0, h_i(x_i, u_i) = 0, x_{i+1} = f_i(x_i, u_i)\}$ is compact and nonempty and that (6.78) - (6.82) yields a unique solution u_i. Assume that a constraint qualification holds at all points considered and that o_i may be chosen such that $o_i(x_i^k) = 0$, $o_i(x_i) \geq 0$ and $o_i(x_i - x_i)/ \parallel x_i - x_i^k \parallel \rightarrow 0$ as $\parallel x_i - x_i^k \parallel \rightarrow 0$. Then any accumulation point for the algorithm satisfies the generalized maximum principle of Proposition 3.4.6.*

Proof. In the backwards direction the adjoint relations hold at stage i at x_i^k by construction. The result then follows as in Proposition 6.4.1. □

6.5 Conclusions

It is possible to construct convergent algorithms that are in an intuitive way applications of the classical maximum principle. Such algorithms capture the essence of the maximum principle idea, linking the maximum part and the dynamic equation in a forwards run, and the adjoint equations in the backwards run.

Such algorithms are simple to implement, in particular if advantage may be taken of the structure at each stage i, permitting analytical or otherwise simple solution to the maximum part. A number of step size ideas for these algorithms have been presented in Section 6.1, based on interpolation or penalties. The obvious alternatives to these algorithms are therefore to use only the gradient (or at stage i, the partial derivative). Also this approach will permit the exploitation of the structure of the local problem, as discussed in Section 6.2.

The classical approach does not handle state dependent constraints, not even an end constraint. In order to treat problems with such constraints it is necessary to consider the interaction between controls at different stages. Through the states acting as intermediate variables the notion of a feedback strategy is then born. Such strategy expresses the control at stage i as a function of the

state at this stage, cf. Section 6.3. This notion is similar to the one used on control theory and dynamic programming.

In the generalized maximum principle of Section 6.4 we treat state dependent constraints. The algorithms here exploit the concept of strategy in order to overcome the dependence of the local constraints on the state.

As analyzed in Section 2.5 another inconvenience in relation to the dependence of the local constraints on the state is that the upper boundaries need not be smooth. In the context of the present algorithms this is implied if the solution or the KKT multipliers (λ_i, μ_i) are not unique. In this case it is necessary to introduce non-smooth supports π_i. Consequently, the adjoint equation has to be substituted by the more general adjoint relation. This has been done in relation to the generalized maximum principle.

All the algorithms apply in a direct way the partial derivatives of the criterion function, mixed with the local criterion function at the stage under consideration at a particular time in the algorithm. Also in relation to the application of strategies the algorithms may be interpreted this way. Therefore the rate of convergence is at most linear, and for this reason the algorithms may not be attractive.

On the other hand it is easy to indicate that some of the ideas in the present chapter, in particular those of the feedback strategy and the nonlinear support, bear a potential for construction of rapidly converging algorithms. Thus, if we in the generalized maximum principle choose $\pi_i = RUB_i$ then we solve the problem in one iteration. In fact, with this choice of π_i, the generalized maximum principle and dynamic programming turn out to be two names for the same thing. This indicates that fast local convergence might be possible, even if π_i and the feedback strategy are chosen as a compromise between accuracy and simplicity of computations.

A difficulty of using $\pi_i = RUB_i$ is that in general this function is complicated, and the feedback strategy will be similarly complicated. The main exception is the QLE problem, which is easily solved, cf. Chapter 4; here, RUB_i is quadratic and the feedback strategy is linear.

In the next chapter the approach will be to base the optimization on solving successive QLE approximations to the OCP. This approach handles the challenge encountered in the present chapter of representing sufficiently close approximations π_i. It also handles the feedback strategies elegantly, as they are linear; when inequality constraints are present as in the QLEI problem, this approach partly breaks down, see Section 7.1. As pointed out in the present chapter such methods need not be interpreted as dynamic programming algorithms, but may also be seen as generalized maximum principle algorithms.

Chapter 7

DDP and Newton Algorithms

From the previous chapter we have learnt that a strategy is necessary if local constraints involving state variables must be handled in a stagewise approach that maintained feasibility. Further, in order to attain fast local convergence, second order approximations must be made.

The approach taken is this chapter is to solve a sequence of quadratic-linear approximations to the problem. In Section 4.3 we showed how the QLE problem could be solved by dynamic programming, and in particular that a strategy for handling state dependent local equality constraints could be found.

Newton's method may be applied to the OCP and the subproblems to be solved are precisely QLE problems. However, this may be done in several ways. Thus, in Section 7.3 we derive the Newton method in two versions. First, we eliminate the state variables and express the problem in terms of the control variables only. Second, we treat state and control variables as optimization variables, and consider the dynamics as a constraint. This version of Newton's method may be seen as solution of the system of KKT equations, and is also known as Wilson's method. For both versions we show that the iterations may be performed by solution of a QLE problem.

In the optimal control literature another approach has been developed, differential dynamic programming, DDP. This is fairly close to the Newton method to the extent that it solves a sequence of QLE problems; both may be seen as versions of sequential quadratic programming. The details differ, though, in particular with respect to the sequence in which the various variables are updated, and the possibility of keeping a feasible solution throughout the iterations. It is believed that the method with QLE subproblems is capable of attaining a quadratic rate of convergence as the Newton method is.

The DP method is dependent on concavity properties of the criterion function when applied to the QLE problem. It is shown in Section 7.2 how the QLE problem may be modified to become concave in order that DP can be applied. Further, in order to handle also local inequality constraints the DP idea will be extended in Section 7.1 to this situation. Here it is also shown that the GMP, the generalized maximum principle, could be applied as well.

DDP applies a DP idea applying an approximation to RUB_i when solving the backwards recursion. Taking in particular the approximation as a quadratic function, and linearizing the dynamics and local constraints, the solution at stage i is relatively simple, and further the feedback form for $u_i^*()$ is linear, cf. Section 4.3 and Section 6.3. For the locally unconstrained problem, DDP algorithms were suggested in Bellman and Dreyfus (1962) (linear approximation), Mayne (1966), Dyer and McReynolds (1970) and Jacobsen and Mayne (1970).

Problems with local constraints were treated in Yakowitz (1986) where global convergence was shown under assumptions of strict concavity. This assumption was removed for locally unconstrained problems in Liao and Shoemaker (1991).

Rate of convergence analysis was undertaken in Murray and Yakowitz (1981), who demonstrated that for locally unconstrained problems the rate of convergence is quadratic. They did so by showing that the deviation from the Newton iterates were small; indeed, that if the dynamics is linear, then the iterates of the two methods coincide. A more direct rate of convergence proof was given in Liao and Shoemaker (1991).

A somewhat different DDP methods was suggested in Ohno (1978), who showed that the rate of convergence for locally constrained problems was quadratic if the local problems were appropriately solved.

Quasi-Newton DDP methods with superlinear rate of convergence were considered in Sen and Yakowitz (1987) and in Rakshit and Sen (1990).

Practical implementations and applications of DDP were described in Yakowitz and Rutherford (1984), Murray and Yakowitz (1979), Liao and Shoemaker (1991). A review is given in Yakowitz (1988,1989).

A main motivation for the development of DDP algorithms was that the attractive quadratic rate of convergence may be attained at a computational cost proportional to Nm^3 (for the locally unconstrained case). In contrast, a direct application of Newton's method requires computational costs proportional to $(Nm)^3$. However, Jonson (1983), Pantoja (1988) and Dunn and Bertsekas (1989) showed independently that the exact Newton iteration may be implemented in a way similar to one DDP iteration and thus requires computational costs only proportional to Nm^3.

In this chapter, we treat aspects of global convergence in Section 7.1 and Section 7.2. Section 7.1 discusses how the QLEI subproblems may be solved by a stagewise approach, applying an active set strategy with DP, or applying the GMP algorithm. Section 7.2 discusses global convergence of the original, non-QLEI problem, applying the absolute value penalty function. The rate of convergence is treated in Section 7.3 with application of Newton iterations and in Section 7.4 the quadratic rate of convergence of the DDP algorithm is derived.

The problem considered is the following one:

$$\max[\sum_{i=0}^{N-1} r_i(x_i,u_i) + r_N(x_N)] \tag{7.1}$$

$$x_{i+1} = f_i(x_i,u_i) \tag{7.2}$$

$$g_i(x_i,u_i) \leq 0 \tag{7.3}$$

$$h_i(x_i,u_i) = 0 \tag{7.4}$$

$$x_0 = \underline{x}_0 \tag{7.5}$$

End constraints are assumed included in (7.3) - (7.4), cf. the discussion around (6.36) - (6.38).

7.1 DP and GMP on the QLEI Problem

In this section we describe an iterative method based on DP for solution of the QLEI problem, i.e., the problem with quadratic criterion function, linear dynamics and linear local equality and inequality constraints. In Section 4.3 we solved by DP the similar problem QLE without the inequality constraints. We extend that method to the QLEI problem using an active set idea, thus solving a sequence of QLE problems. As will be seen, special care must be taken due to

the stagewise approach. We also show that GMP, the generalized maximum principle algorithm, Section 6.4, may be applied. This is not based on an active set idea, but rather on application of non-smooth functions π_i.

Application of DP

The local problem to be solved at stage i in backwards dynamic programming was considered in Section 4.3. Here we give the following adaption:

$$\max_{u_i}[x_i^{k\prime}A_i^k x_i^k + x_i^{k\prime}B_i^k u_i + u_i' C_i^k u_i + D_i^k u_i + E_i^k x_i^k] \tag{7.6}$$

$$(G_i^x x_i^k + G_i^u u_i - \underline{g}_i)^j \leq 0,\ j \in J_i^\epsilon(x_i^k, u_i^k) \tag{7.7}$$

$$(G_i^x x_i^k + G_i^u u_i - \underline{g}_i)^j = 0,\ j \in J_i^B \tag{7.8}$$

$$H_i^x x_i^k + H_i^u u_i - \underline{h}_i = 0 \tag{7.9}$$

In this, (x_i^k, u_i^k) is the current iteration point, with k being the iteration counter. ϵ is a given nonnegative number. J_i^ϵ is the set of indexes j on inequality constraints which are at most ϵ from being active at the point (x_i^k, u_i^k), i.e.

$$J_i^\epsilon(x_i^k, u_i^k) = \{j \mid (G_i^x x_i^k + G_i^u u_i^k - \underline{g}_i)^j \geq -\epsilon\} \tag{7.10}$$

and we define J^ϵ as the set of index pairs (i,j) for which $j \in J_i^\epsilon$ for some (i,j). In particular we define J_i^0, J^0, J_i^∞ and J^∞ corresponding to $\epsilon = 0$ and $\epsilon = \infty$, respectively.

Given the solution u_i^o to (7.6) - (7.9) we can identify the set of inequality constraints which shall be treated as equality constraints in the definition of Q_i and P_i, cf. Section 4.3. This set is denoted J_i^A and is identified as containing those indexes (i,j) for which

$$(G_i^x x_i^k + G_i^u u_i^k - \underline{g}_i)^j = 0 \text{ and } (G_i^x x_i^k + G_i^u u_i^o - \underline{g}_i)^j = 0 \tag{7.11}$$

J^A is the set of index pairs (i,j) for which $j \in J_i^A$ for some (i,j).

We also define the index sets J_i^B which are the sets of indexes j for which we force the inequality to hold as equality, i.e.,

$$j \in J_i^B \Rightarrow (G_i^x x_i^k + G_i^u u_i - \underline{g}_i)^j = 0 \tag{7.12}$$

cf. (7.8), and J^B is the set of index pairs (i,j) for which $j \in J_i^B$ for some (i,j). We shall only let J_i^B contain indexes (i,j) for which $(G_i^x x_i^k + G_i^u u_i^k - \underline{g}_i)^j = 0$, therefore $j \in J_i^B \Rightarrow j \in J_i^A$.

The set J_i^B is necessary because the apparently natural definition of J_i^A will miss some indexes j. The reason is the following. Without the set J_i^B there may be a solution u_i^o such that $(G_i^x x_i^k + G_i^u u_i^o - \underline{g}_i)^j < 0$ and yet it is not possible to take a positive steplength because for any positive steplength, $(G_i^x x_i + G_i^u u_i - \underline{g}_i)^j > 0$. This in turn is due to the simultaneous changes in all controls, and the implied change in x_i. Thus, J^B captures elements of the interplay between the stages that can not be identified stagewise.

The set of equality constraints used at stage i in the backwards QLE DP procedure to define Q_i, P_i, L_i and K_i are all the inequalities with $j \in J_i^A$ and the equality constraints (7.9).

This also identifies the KKT multipliers μ_i relative to the equality constraints (7.9) and λ_i^j relative to (7.7) - (7.8) for $j \in J_i^A$. These can be found as explained in Section 4.3, where it was shown how to find the KKT multipliers μ_i^j relative to the equality constraints. Therefore the

optimal solution to the QLE problem consisting of (7.7) - (7.8) for $(i,j) \in J^A$ in addition to (7.9) can be found as explained in Section 4.3.

We adopt this as follows. The solution to (7.6) - (7.9) with the corresponding KKT multipliers may be written as:

$$\begin{pmatrix} u_i^o \\ \hat{\lambda}_i^o \\ \mu_i^o \end{pmatrix} = \begin{pmatrix} -2C_i^k & \hat{G}_i^{u\prime} & H_i^{u\prime} \\ \hat{G}_i^u & 0 & 0 \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} D_i \\ \underline{\hat{g}}_i \\ \underline{h}_i \end{pmatrix} + \begin{pmatrix} -2C_i^k & \hat{G}_i^{u\prime} & H_i^{u\prime} \\ \hat{G}_i^u & 0 & 0 \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} B_i' \\ -\hat{G}_i^x \\ -H_i^x \end{pmatrix} x_i \tag{7.13}$$

The matrices $\hat{G}_i^u$, $\hat{G}_i^x$ and $\underline{\hat{g}}_i$ contain those rows of G_i^u, G_i^x and $\underline{g}_i$, respectively, which correspond to active inequality constraints, i.e. those with $j \in J_i^A$. The remaining λ_i^j are zero. The equation (7.13) identifies $\underline{K}_i$ and $\underline{L}_i$ cf. Section 4.3, and the upper m rows of these are K_i and L_i, respectively, such that

$$\begin{pmatrix} u_i^o \\ \hat{\lambda}_i^o \\ \mu_i^o \end{pmatrix} = \underline{K}_i + \underline{L}_i x_i \tag{7.14}$$

and in particular

$$u_i^o = K_i + L_i x_i \tag{7.15}$$

However, some of the inequality constraints not included in J^A may be violated at the new optimal strategy and trajectory given by the forwards DP recursion using (7.15) and the dynamic equation. We will therefore have to use a restricted step length. If the step length is α, $0 \leq \alpha \leq 1$, then the new point $(u_i^{k+1}, \hat{\lambda}_i^{k+1}, \mu_i^{k+1}, x_{i+1}^{k+1})$ is constructed in the forwards DP recursion as

$$\begin{pmatrix} u_i^{k+1} \\ \hat{\lambda}_i^{k+1} \\ \mu_i^{k+1} \end{pmatrix} = \alpha \begin{pmatrix} u_i^o \\ \hat{\lambda}_i^o \\ \mu_i^o \end{pmatrix} + (1-\alpha) \begin{pmatrix} u_i^k \\ \hat{\lambda}_i^k \\ \mu_i^k \end{pmatrix} + \underline{L}_i (x_i^{k+1} - x_i^k) \tag{7.16}$$

$$x_{i+1}^{k+1} = F_i^x x_i^{k+1} + F_i^u u_i^{k+1} + \underline{f}_i \tag{7.17}$$

starting with $x_0^{k+1} = \underline{x}_0$. For $(i,j) \notin J^A$ we have $(\lambda_i^j)^{k+1} = 0$.

We can now define an algorithm as follows:

DP Algorithm for the QLEI problem

Step 0 Let a feasible point (x^0, u^0) be given. Let an $\epsilon \geq 0$ be given. Let $\lambda_i^0 = 0$ and $\mu_i^0 = 0$ for all i. Let $J^B = \emptyset$. Let $k = 0$.

Step 1 Backwards DP solution, $i = N-1, \ldots, 0$:
Formulate and solve (7.6) - (7.9) to find

- $(u_i^o, \hat{\lambda}_i^o, \mu_i^o)$
- J_i^A

- $\underline{K}_i$ and $\underline{L}_i$, cf. (7.14) and Q_i, P_i, cf. Section 4.3.

Step 2 Optimality test:

- If $u^o = u^k$: If $\lambda \geq 0$ then stop, else go to Step 3.
- If $u_i^o \neq u_i^k$ for some i: go to Step 4.

Step 3 Revision of J^B:
Find one or more index pairs (i^*, j^*) for which $(\lambda_i^j)^k < 0$. Exclude (i^*, j^*) from J^B, let $(\lambda_{i*}^{j*})^k = 0$ and go to Step 1.

Step 4 Trial step:
Construct the solution $(\tilde{x}, \tilde{u})$ by a forwards DP recursion, $i = 0, \ldots, N-1$, with $\alpha = 1$ (use (7.16) - (7.17) but write $(\tilde{x}_i, \tilde{u}_i)$ in stead of (x_i^{k+1}, u_i^{k+1})).

Step 5 Find steplength:
Find $\overline{\alpha}$ such that

$$\overline{\alpha} = \min_{i,j}[\frac{-g_i^j(x_i^k, u_i^k)}{g_i^j(\tilde{x}_i, \tilde{u}_i) - g_i^j(x_i^k, u_i^k)}]$$

over all (i,j) for which $g_i^j(\tilde{x}_i, \tilde{u}_i) > 0$.

- If $0 < \overline{\alpha}$ then go to Step 6.
- If $\overline{\alpha} = 0$ then go to Step 7.

Step 6 Restricted step:
Construct $(x^{k+1}, u^{k+1}, \hat{\lambda}^{k+1}, \mu^{k+1})$ from (7.16) - (7.17) with $\alpha = \min\{\overline{\alpha}, 1\}$. Let $k = k+1$. Go to Step 1.

Step 7 Revision of J^B:
Include (i,j) in J^B, where (i,j) is one indexpair giving $\overline{\alpha} = 0$ in Step 5. Go to Step 1.

Proposition 7.1.1 *Assume an initial feasible point (x^0, u^0) given. Assume V_i compact. Assume $C_i^k < 0$ on the subspace defined by (7.8) - (7.9) for all i and all k. Assume all constraints in J_i^A linearly independent with respect to u_i. Assume all $\lambda_i^j \neq 0$, $(i,j) \in J^A$. Then the algorithm iterates through feasible points and finds in a finite number of steps the unique optimal solution.*

Proof. The algorithm iterates through feasible points due to the assumption of an initial feasible point and the restricted step length used in Step 6. By the assumption of $C_i^k < 0$ on the subspace (7.8) - (7.9) and the assumption of linear independence the DP solution of the QLE is well defined and unique, cf. Proposition 4.3.1.

If $\overline{\alpha} > 0$, an increase is attained in the criterion, and after a finite number of iterations (Step 6) the optimal solution for a given J^B will be found in Step 2. If $\lambda \geq 0$ then the solution is optimal and the algorithm stops. If not, J^B is revised (Step 3); this will permit an increase of the criterion value due to dropping one or more constraints in J^B with $\lambda_i^j < 0$.

By the assumption of linear independence and $\lambda_i^j \neq 0$ for $(i,j) \in J^A$ this part of the algorithm will not cycle, and the active set part of the algorithm will therefore in a finite number of arithmetic operations either produce a feasible point with an increased criterion value or stop with $J^B = \emptyset$.

As seen, the algorithm either stops or increases the criterion. We observe that the algorithm will construct only a finite number of points, because there are only a finite number of combinations of active constraints in J^A and only a finite number of indexpair combinations (i,j) in Step 5. In

each point a finite number of arithmetic operations are made. The feasible set is compact, and therefore the algorithm terminates in a finite number of steps.

If all inequality constraints g_i^j for which $\lambda_i^j = 0$ at the final point are dropped, the solution found is optimal with respect to the remaining problem, considered as a QLE, cf. Proposition 4.3.1. Therefore the KKT necessary conditions may be formulated for the solution found. Since $C_i < 0$ the problem has a concave criterion functions on the feasible subspace, cf. Proposition 4.3.2, and the KKT conditions are sufficient for optimality. □

We now add a number of additional comments. An initial feasible solution may formally be attained by modifying the problem by introducing additional variables as discussed in Section 2.5.

If the set of active constraints at stage i are linearly dependent with respect to u_i then this difficulty may be surmounted by the stage aggregation technique discussed in Section 2.5.

We need not calculate μ_i corresponding to the equality constraints (7.9). Further we need only calculate λ_i^j corresponding to blocking inequality constraints (i.e, those with $(i,j) \in J^B$) when they are needed in the active set part of the algorithm (Step 3). The KKT multipliers may be calculated if this is expedient for the solution of (7.6) - (7.9). This depends on the solution technique chosen; if this for instance is duality based, then $(\lambda^{k-1}, \mu^{k-1})$ may serve as good initial values.

If $i^\diamond$ is the largest index i such that $\overline{\alpha}$ is attained at stage $i^\diamond$ the solution to (7.6) - (7.9) remains the same for all i with $i^\diamond < i$. Therefore the backwards solution procedure (Step 1) can in this case be initiated from stage $i^\diamond$. Also when an inequality constraint with index $(i^\diamond, j^\diamond)$ is included in J^B in Step 7 the solution to (7.6) - (7.9) remains the same for all i with $i^\diamond < i$ and the backwards solution procedure (Step 1) can also in this case be initiated from stage $i^\diamond$.

If we take $\epsilon = \infty$ then all inequality constraints are considered in (7.6) - (7.9). The idea of using a finite ϵ is that the solution of (7.6) - (7.9) may be easier if there are fewer inequality constraints to be considered. The idea of not necessarily using $\epsilon = 0$ is that the algorithm may choose better solution directions u_i^o in (7.6) - (7.9) if more constraints are considered. For further discussion of the role of ϵ, see Bertsekas (1982).

The role of the set J^B may be clarified by the following example:

Example 7.1.1 *Let $n = m = 1$, $N = 4$, $x_0 = \underline{x}_0$ and let there be one inequality and one equality constraint at stage 2 in the following problem:*

$$\max[\sum_{i=0}^{3} -(u_i)^2]$$

$$x_{i+1} = x_i + u_i$$

$$x_0 = 0$$

$$g_2^1(x_2, u_2) = x_2 - u_2 + 1 \leq 0$$

$$h_3^1(x_3, u_3) = x_3 + u_3 - 1 = 0$$

Let the initial values be $u^0 = (0, -1, 0, 2)'$, $x^0 = (0, 0, -1, -1, 1)'$. We verify that this satisfies the dynamic equation and also the local constraints: $g_2^1(x_2^0, u_2^0) = 0 \leq 0$ and $h_3^1(x_3^0, u_3^0) = 0$. We see that $J_2^\epsilon(x_2^0, u_2^0) = \{1\}$ for all $\epsilon \geq 0$. We let $J^B = \emptyset$.

From Example 4.3.1 page 136 we have $u_3(x_3) = 1 - x_3$, $Q_3 = (-1)$, $P_3 = (2)$, $A_2 = (-1)$, $B_2 = (-2)$, $C_2 = (-2)$, $D_2 = (2)$ and $E_2 = (2)$, such that the optimization problem corresponding to (7.6) - (7.9) at stage 2 is

$$\max_{u_2}[(-1)'(-1)(-1) + (-1)'(-2)u_2 + u_2'(-2)u_2 + (2)u_2 + (2)(-1)]$$

$$g_2^1(-1,u_2) = -1 - u_2 + 1 \leq 0$$

The unique optimal solution to this is $u_2 = 1$. *We find* $g_2^1(-1,1) = -1 < 0$. *We therefore find that* $1 \notin J_2^A$.

Therefore $K_2 = (\frac{1}{2})$, $L_2 = (-\frac{1}{2})$, $Q_2 = (-\frac{1}{2})$, $P_2 = (1)$, $A_1 = (-\frac{1}{2})$, $B_1 = (-1)$, $C_1 = (-3/2)$, $D_1 = (1)$ *and* $E_1 = (1)$, *as found in Example 4.2.1 page 134. Also at stages 1 and 0 the solution is seen to be as described in that example.*

The trial step with $\alpha = 1$ *gives the solution* $x_0^* = 0$, $u_0^* = 1/4$, $x_1^* = 1/4$, $u_1^* = 1/4$, $x_2^* = \frac{1}{2}$, $u_2^* = 1/4$, $x_3^* = 3/4$ *and* $u_3^* = 1/4$, *cf. again Example 4.2.1. We find that* $g_2^1(x_2^*, u_2^*) = 5/4 > 0$. *We see that the constraint* g_2^1 *is blocking for taking any positive step. Therefore from Step 5 of the algorithm* $\overline{\alpha} = 0$ *and in Step 7 we let* $J^B = \{(2,1)\}$, *i.e., require* $g_2^1(x_2^0, u_2) = 0$.

The solution now was given in Example 4.3.1 as $x_0^* = 0$, $u_0^* = -1/11$, $x_1^* = -1/11$, $u_1^* = -1/11$, $x_2 = -2/11$, $u_2^* = 9/11$, $x_3^* = 2/11$, $u_3^* = 4/11$ *and* $x_4^* = 1$. *We verify that* $g_2^1(x_2^*, u_2^*) = 0$.

Now the problem corresponding to (7.6) - (7.9) at stage 2 with $J^B = \emptyset$ *is*

$$\max_{u_2}[(-2/11)'(-1)(-2/11) + (-2/11)'(-2)u_2 + u_2'(-2)u_2 + (2)u_2 + (2)(-2/11)]$$

$$g_2^1(-2/11,u_2) = 9/11 - u_2 \leq 0$$

The solution now is $u_2^* = 9/11$.

It is observed that when we identified g_2^1 *as blocking it would have been insufficient to just include* $(i,j) = (2,1)$ *in* J^B *and then again try with the forwards recursion found in the first place; it is necessary to reoptimize at stages 2, 1 and 0. Thus, the correct direction* δu_0 *with* g_2^1 *active is a decrease, i.e.* $\delta u_0 < 0$, *while the direction* δu_0 *found with* g_2^1 *as an inequality constraint specifies* $\delta u_0 > 0$. □

A weakness of the active set idea is that changes in the set J^B take place relatively slowly over the iterations, and therefore many QLE problems have to be solved.

If the inequality constraints (or a substantial part of them) are simple lower and upper bounds on the control variable, $\underline{u}_i \leq u_i \leq \overline{u}_i$, then projection methods may be advantageous. In these, an improving direction is first found, then it is modified by projection onto the feasible set, cf. e.g. (6.33) page 177. This permits a more flexible handling of inequality constraints. For a concave criterion function the direction of the projected gradient is an improving direction, but this need not be so for the direction of the projected Newton direction, cf. e.g. Bertsekas (1982a), Bertsekas (1982).

For the optimal control problem, projection methods were described in Bertsekas (1982a), Jonsen (1983) and Gawande and Dunn (1988). These methods fit well into the general structure of the above DP based solution method, also when this is used as subproblem in solution of general, smooth non-QLEI problems to be treated in subsequent sections.

Application of GMP

As discussed in Section 6.4, application of dynamic programming and the GMP may be seen to be very similar. We now extend on this by showing how GMP may be applied to the QLEI problem.

At stage i rather than the problem (7.6) - (7.9) we have

$$\max_{u_i,\alpha_{i+1}} [\tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^x x_i + R_i^u u_i \qquad (7.18)$$
$$+\alpha_{i+1} + (F_i^x x_i + F_i^u u_i + \overline{f}_i)'\tilde{Q}_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i)]$$

$$G_i^x x_i^k + G_i^u u_i - \underline{g}_i \leq 0 \tag{7.19}$$

$$H_i^x x_i^k + H_i^u u_i - \underline{h}_i = 0 \tag{7.20}$$

$$\alpha_{i+1} \leq (\nabla_x (H_i(x_i^k, u_i^*, \pi_{i+1}) - \lambda_i^e g_i(x_i^k, u_i^*) - \mu_i^e h_i(x_i^k, u_i^*))) \\ ((F_i^x x_i + F_i^u u_i + \overline{f}_i) - x_{i+1}^k),\ e = 1, \ldots, E_{i+1} \tag{7.21}$$

Here the matrix $\tilde{Q}_{i+1}$ is chosen such that the function π_{i+1} implicitly defined (see also (6.73) - (6.82) in Section 6.4) is a lower support, irrespective of what constraint(s) are active in (7.21). The formulation assumes a unique solution at stage $i+1$ (compare (6.71) and (6.77)), this is fulfilled if the local criterion function is strictly concave, i.e., under the same assumptions as taken for the DP approach.

It is essential to observe that if $E_{i+1} > 1$ in (7.21), or if there are inequality constraints (7.19) present, then the feedback strategy $u_i(.)$ can not necessarily be given in the linear form (7.15): it will depend on which constraints are active in (7.21) and (7.19) and this in turn depends on x_i in nonlinear ways.

Given the solution to the QLEI problem (7.18) - (7.21) we may proceed recursively backwards to stage $i = 0$. In the forwards direction we have to maximize (7.18) - (7.21) again, relative to the new x_i found, $i = 0, \ldots, N-1$ for the reason given above.

We see that it is possible to apply the GMP towards solution of the QLEI problem. As in the application of DP with the active set idea, the solution procedure will be iterative. However, where the DP approach has an inner loop with update (also iterative) of J^B to determine the feedback strategies, the GMP approach finds the feedback strategies directly at the cost of solving optimization problems also in the forwards direction.

7.2 The Linearization Method, DDP and GMP

We shall in this section discuss how to solve a smooth OC problem by solving a sequence of QLEI problems. These QLEI problems in turn may be solved by any suitable method in particular DP or GMP as described in Section 7.1. The aim of this is the establishment of global convergence. The exact specification of the QLEI problem to be solved is not unique. In particular, we may or may not choose to eliminate the state variables. The particular iterations will depend on this choice. Further, application of the DP or GMP solution of Section 7.1 will permit a certain freedom in the choice of updating. The choice of version will in particular influence the possibility of keeping a feasible solution throughout the iterations.

Our point of departure is the linearization method with the absolute value penalty function. Therefore we introduce this first. Consider the following mathematical programming problem:

$$\max[r(z)] \tag{7.22}$$

$$g(z) \leq 0 \tag{7.23}$$

$$h(z) = 0 \tag{7.24}$$

Given a nominal z^k we linearize this to get

$$\max[\nabla r(z^k)\delta_z + \tfrac{1}{2}\delta_z' S \delta_z] \tag{7.25}$$

$$g(z^k) + \nabla g(z^k)\delta_z \leq 0 \tag{7.26}$$

$$h(z^k) + \nabla h(z^k)\delta_z = 0 \tag{7.27}$$

Here S is a matrix of suitable dimensions, and $S < 0$.

Let δ_z^* be the optimal solution to (7.25) - (7.27). Then the new nominal point z^{k+1} is found as

$$z^{k+1} = z^k + \alpha \delta z^* \tag{7.28}$$

where α is a scalar such that the absolute value merit function

$$r(z^{k+1}) - \gamma |g(z^{k+1})|_+ - \gamma |h(z^{k+1})| \tag{7.29}$$

is suitably increased relative to the similar expression evaluated at z^k. Here γ is a positive parameter and

$$|h(z)| = \sum_{j=1}^{\ell} |h_j(z)| \tag{7.30}$$

$$|g_j(z)|_+ = \max\{g_j(z), 0\} \tag{7.31}$$

$$|g(z)|_+ = \sum_{j=1}^{k} |g_j(z)|_+ \tag{7.32}$$

An α giving a suitable increase is obtained by maximization of the expression (7.29) with respect to α or by using e.g. the Armijo rule. If a feasible z^k does not satisfy the KKT conditions then the direction δ_z^* is an improving direction for the absolute value merit functions (7.29), provided the scalar γ is greater than or equal to the greatest of the absolute values of the KKT multipliers corresponding to the solution of (7.25) - (7.27). The direct application of this linearization method to the optimal control problem is straightforward. However, there are at least three ways, in which this method can be adapted to the specific structure of the optimal control problem. We consider these in turn under the headings Control-space Interpolation, State-and-control-space Interpolation and Differential Dynamic Programming Interpolation.

Control-space Interpolation

The OCP with fixed initial state $\underline{x}_0$ may be reformulated by elimination of all x_i for $1 \leq i \leq N$ and then interpreting $z = u = (u_0', \ldots, u_{N-1}')'$. Let $r : R^{Nm} \to R$ denote this criterion function:

$$\begin{aligned} r(u) = & \\ & r_0(\underline{x}_0, u_0) + r_1(f_0(\underline{x}_0, u_0), u_1) + \ldots \\ & \ldots + r_N(f_{N-1}(\ldots(f_0(\underline{x}_0, u_0), u_1)\ldots), u_{N-1}) \end{aligned} \tag{7.33}$$

Let a nominal u^k be given. Assume x^k calculated recursively forwards from the dynamic equation

$$x_{i+1}^k = f_i(x_i^k, u_i^k) \tag{7.34}$$

starting from $x_0^k = \underline{x}_0$.

Now formulate the linearized problem corresponding to (7.25) - (7.27) in the variables $(\delta x, \delta u) = (x - x^k, u - u^k)$ as

$$\max_{\delta x, \delta u} [\sum_{i=0}^{N-1} (\nabla_x r_i(x_i^k, u_i^k)\delta x_i + \nabla_u r_i(x_i^k, u_i^k)\delta u_i \tag{7.35}$$

$$+\tfrac{1}{2}\delta x_i' S_i^{xx}\delta x_i + \delta x_i' S_i^{xu}\delta u_i + \tfrac{1}{2}\delta u_i' S_i^{uu}\delta u_i)$$
$$+\nabla r_N(x_N^k)\delta x_N + \tfrac{1}{2}\delta x_N' S_N^{xx}\delta x_N]$$
$$\delta x_{i+1} = \nabla_x f_i(x_i^k, u_i^k)\delta x_i + \nabla_u f_i(x_i^k, u_i^k)\delta u_i \tag{7.36}$$
$$g_i(x_i^k, u_i^k) + \nabla_x g_i(x_i^k, u_i^k)\delta x_i + \nabla_u g_i(x_i^k, u_i^k)\delta u_i \leq 0, \tag{7.37}$$
$$j \in J_i^\epsilon(x_i^k, u_i^k)$$
$$h_i(x_i^k, u_i^k) + \nabla_x h_i(x_i^k, u_i^k)\delta x_i + \nabla_u h_i(x_i^k, u_i^k)\delta u_i = 0 \tag{7.38}$$
$$\delta x_0 = 0 \tag{7.39}$$

We observe that with $(\delta x, \delta u)$ satisfying (7.36) and (7.39) and (x^k, u^k) satisfying (7.34) we have that (7.33) and the criterion function in [ ] in (7.35) are related as follows:

$$\nabla r(u^k)\delta u = \tag{7.40}$$
$$\sum_{i=0}^{N} \nabla_x r_i(x_i^k, u_i^k)\delta x_i + \sum_{i=0}^{N-1} \nabla_u r_i(x_i^k, u_i^k)\delta u_i$$

such that to the first order approximation the two versions of the criterion function coincide.

In (7.35) S_i^{xx}, S_i^{xu} and S_i^{uu} are $n \times n$, $n \times m$ and $m \times m$ matrices, respectively, with S_i^{xx} and S_i^u symmetric, which could be chosen such that $S_N^{xx} < 0$ and

$$\begin{pmatrix} S_i^{xx} & S_i^{xu} \\ S_i^{xu\prime} & S_i^{uu} \end{pmatrix} < 0 \tag{7.41}$$

Weaker conditions may be applicable. What is important is that for the direction $(\delta x^*, \delta u^*) \neq (0,0)$ solving (7.35) - (7.39) we have

$$\tfrac{1}{2}\sum_{i=0}^{N} \delta x_i^{*\prime} S_i^{xx}\delta x_i^* + \sum_{i=0}^{N-1} \delta x_i^{*\prime} S_i^{xu}\delta u_i^* + \tfrac{1}{2}\sum_{i=0}^{N-1} \delta u_i^{*\prime} S_i^{uu}\delta u_i^* < 0 \tag{7.42}$$

If we solve (7.35) - (7.39) by DP this is essentially the same as $C_i < 0$ on the relevant subspace, cf. Proposition 4.3.2. We note that we need not in (7.37) consider all inequalities g_i^j. As in (7.7) we can use a subset $J_i^\epsilon(x_i^k, u_i^k)$ of all indexes (i,j). In contrast to the QLEI problem we shall use a strictly positive ϵ.

Once the solution $(\delta x^*, \delta u^*)$ to (7.35) - (7.39) is found, the new point (x^{k+1}, u^{k+1}) is calculated recursively forwards, given the scalar α:

$$u_i^{k+1} = u_i^k + \alpha\delta u_i^* \tag{7.43}$$
$$x_{i+1}^{k+1} = f_i(x_i^{k+1}, u_i^{k+1}) \tag{7.44}$$

starting from $x_0^{k+1} = \underline{x}_0$.

The interesting feature here is that the interpolation with α is only in u, see (7.43), and due to (7.44) the dynamical equation is fulfilled at the new iteration point (x^{k+1}, u^{k+1}). The absolute value merit function corresponding to (7.29) is

$$MF_\gamma(\alpha) = MF_\gamma(x^{k+1}, u^{k+1}) = \tag{7.45}$$
$$\sum_{i=0}^{N-1} r_i(x_i^{k+1}, u_i^{k+1}) + r_N(x_N^{k+1}) - \gamma\sum_{i=0}^{N-1}\sum_{j=1}^{k} |g_i^j(x_i^{k+1}, u_i^{k+1})|_+$$

$$-\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} |h_i^j(x_i^{k+1}, u_i^{k+1})|$$

Here γ is chosen such that it satisfies

$$\gamma \geq \lambda_i^j \tag{7.46}$$

$$\gamma \geq |\mu_i^j| \tag{7.47}$$

over all (i,j), where λ and μ are the KKT multipliers corresponding to the solution of (7.35) - (7.39). Observe that no terms $|x_{i+1} - f_i(x_i, u_i)|$ are used here because the constraints corresponding to (7.54) are eliminated due to (7.44) and therefore automatically fulfilled.

Proposition 7.2.1 *Assume that for all i r_i, f_i, g_i, and h_i are continuously differentiable. Let (x^k, u^k) satisfy (7.34). Let $\epsilon > 0$. Assume (7.42) holds. Assume (7.35) - (7.39) solved and let the unique nonzero solution be $(\delta x^*, \delta u^*)$. Let λ and μ be KKT multipliers corresponding to (7.37) - (7.38). Assume γ chosen such that (7.46) - (7.47) hold. Let $MF_\gamma(x^{k+1}, u^{k+1})$ be given from (7.45) with (x^{k+1}, u^{k+1}) calculated from (7.43) - (7.44) for a given α with $0 \leq \alpha \leq 1$. Then there is an α that provides an increase of MF_γ.*

Proof. Let $\Delta u_i = u_i^{k+1} - u_i^k$, $\Delta x_i = x_i^{k+1} - x_i^k$. From (7.43) is follows that $\Delta u_i = \alpha \delta u_i^*$. We will first show that $\Delta x_i = \alpha \delta x_i^* + o_i^x(\alpha)$, where $o_i^x(\alpha)/\alpha \to 0$ as $\alpha \to 0$. Obviously this holds for $i = 0$ due to the initialization of (7.43) - (7.44). Now assume it holds for a given i. We then have

$$\begin{aligned}
&\Delta x_{i+1} = x_i^{k+1} - x_i^k = f_i(x_i^{k+1}, u_i^{k+1}) - f_i(x_i^k, u_i^k) \\
&= \nabla_x f_i(x_i^k, u_i^k)\Delta x_i + \nabla_u f_i(x_i^k, u_i^k)\Delta u_i + o_i(\Delta x_i, \Delta u_i) \\
&= \nabla_x f_i(x_i^k, u_i^k)(\alpha \delta x_i^* + o_i^x(\alpha)) + \nabla_u f_i(x_i^k, u_i^k)(\alpha \delta u_i^*) + o_i(\alpha \delta x_i^* + o_i^x(\alpha), \alpha \delta u_i^*) \\
&= \alpha(\nabla_x f_i(x_i^k, u_i^k)\delta x_i^* + \nabla_u f_i(x_i^k, u_i^k)\delta u_i^*) + o_{i+1}^x(\alpha) \\
&= \alpha \delta x_{i+1}^* + o_{i+1}^x(\alpha)
\end{aligned}$$

where $o_i(\Delta x_i, \Delta u_i)/(\| \Delta x_i \| + \| \Delta u_i \|) \to 0$ as $(\| \Delta x_i \| + \| \Delta u_i \|) \to 0$ and $o_{i+1}^x(\alpha)/\alpha \to 0$ as $\alpha \to 0$. It follows that the expression holds for all i.

Now consider the following formulae where all function values and all gradients are evaluated at the point (x^k, u^k) (r_N and ∇r_N evaluated at x_N^k). All g_i^j with $(i,j) \notin J^\epsilon$ are temporarily disregarded. We find

$$MF_\gamma(\alpha) \equiv MF_\gamma(x^{k+1}, u^{k+1}) \tag{7.48}$$

$$\begin{aligned}
&= \sum_{i=0}^{N} r_i + \alpha \sum_{i=0}^{N-1} \nabla_x r_i(\delta x_i^* + o_i^x(\alpha)) + \alpha \sum_{i=0}^{N-1} \nabla_u r_i \delta u_i^* + \alpha \nabla r_N(\delta x_N^* + o_N^x(\alpha)) \\
&\quad -\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{k} |g_i^j + \alpha \nabla_x g_i^j(\delta x_i^* + o_i^x(\alpha)) + \alpha \nabla_u g_i^j \delta u_i^*|_+ \\
&\quad -\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} |h_i^j + \alpha \nabla_x h_i^j(\delta x_i^* + o_i^x(\alpha)) + \alpha \nabla_u h_i^j \delta u_i^*| + \tilde{o}(\alpha) \\
&= \sum_{i=0}^{N} r_i + \alpha \sum_{i=0}^{N-1} \nabla_x r_i \delta x_i^* + \alpha \sum_{i=0}^{N-1} \nabla_u r_i \delta u_i^* + \alpha \nabla r_N \delta x_N^*
\end{aligned}$$

$$
\begin{aligned}
&-\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{k} |g_i^j|_+ - \alpha\gamma \sum_{i=0}^{N-1} \sum_{j \in M_i} (\nabla_x g_i^j \delta x_i^* + \nabla_u g_i^j \delta u_i^*) \\
&-\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} |h_i^j| - \alpha\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} (\nabla_x h_i^j \delta x_i^* + \nabla_u h_i^j \delta u_i^*) + o(\alpha) \\
&= MF_\gamma(0) + \alpha\Big(\sum_{i=0}^{N-1} \nabla_x r_i \delta x_i^* + \nabla r_N \delta x_N^* + \sum_{i=0}^{N-1} \nabla_u r_i \delta u_i^*\Big) \\
&\qquad -\alpha\gamma \sum_{i=0}^{N-1} \sum_{j \in M_i} (\nabla_x g_i^j \delta x_i^* + \nabla_u g_i^j \delta u_i^*) \\
&\qquad -\alpha\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} (\nabla_x h_i^j \delta x_i^* + \nabla_u h_i^j \delta u_i^*) + o(\alpha)
\end{aligned}
$$

In the last expressions the index set M_i contains those j for which $g_i^j(x_i^k, u_i^k) > 0$. It is valid to limit the summations to those (i, j) because if $g_i^j(x_i^k, u_i^k) \leq 0$ then (7.37) is fulfilled for $(\delta x_i^*, \delta u_i^*)$ and also for $(\alpha\delta x_i^*, \alpha\delta u_i^*)$ if $0 \leq \alpha \leq 1$. Now we observe that for all such α and with $\gamma \geq 0$ we have

$$
\alpha\gamma \sum_{i=0}^{N-1} \sum_{j \in M_i} (\nabla_x g_i^j \delta x_i + \nabla_u g_i^j \delta u_i) \leq -\alpha\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{k} |g_i^j|_+ \tag{7.49}
$$

$$
\alpha\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} (\nabla_x h_i^j \delta x_i + \nabla_u h_i^j \delta u_i) \leq -\alpha\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{\ell} |h_i^j| \tag{7.50}
$$

By assumption, p, λ and μ are the KKT multipliers corresponding to the solution to (7.35) - (7.39). Therefore the following stationarity part of the KKT conditions holds,

$$
\begin{aligned}
\nabla_u r_i + \delta x_i^{*\prime} S_i^{xu} + \delta u_i^{*\prime} S_i^{uu} + p_{i+1} \nabla_u f_i - \lambda_i \nabla_u g_i - \mu_i \nabla_u h_i = 0 \\
\nabla_x r_i + \delta x_i^{*\prime} S_i^{xx} + \delta u_i^{*\prime} S_i^{xu} + p_{i+1} \nabla_x f_i - p_i - \lambda_i \nabla_x g_i - \mu_i \nabla_x h_i = 0 \\
\nabla r_N - p_N = 0
\end{aligned}
$$

By multiplication with δu_i^*, δx_i^* and δx_N^*, respectively, summation, and further manipulation we get from this

$$
\begin{aligned}
&\sum_{i=0}^{N} \nabla_x r_i \delta x_i^* + \sum_{i=0}^{N-1} \nabla_u r_i \delta u_i^* \qquad (7.51) \\
&= -\sum_{i=0}^{N} \delta x_i^{*\prime} S_i^{xx} \delta x_i^* - 2 \sum_{i=0}^{N-1} \delta x_i^{*\prime} S_i^{xu} \delta u_i^* - \sum_{i=0}^{N-1} \delta u_i^{*\prime} S_i^{uu} \delta u_i^* \\
&\quad - \sum_{i=0}^{N-1} p_{i+1} \nabla_x f_i \delta x_i^* + \sum_{i=0}^{N} p_i \delta x_i - \sum_{i=0}^{N-1} p_{i+1} \nabla_u f_i \delta u_i^* \\
&\quad + \sum_{i=0}^{N-1} \lambda_i \nabla_x g_i \delta x_i^* + \sum_{i=0}^{N-1} \lambda_i \nabla_u g_i \delta u_i^*
\end{aligned}
$$

$$+\sum_{i=0}^{N-1}\mu_i\nabla_x h_i\delta x_i^*+\sum_{i=0}^{N-1}\mu_i\nabla_u h_i\delta u_i^*$$
$$\geq -\sum_{i=0}^{N}\delta x_i^{*\prime}S_i^{xx}\delta x_i^*-2\sum_{i=0}^{N-1}\delta x_i^{*\prime}S_i^{xu}\delta u_i^*-\sum_{i=0}^{N-1}\delta u_i^{*\prime}S_i^{uu}\delta u_i^*$$
$$-\gamma\sum_{i=0}^{N-1}\sum_{j=1}^{k}|g_i^j|_+-\gamma\sum_{i=0}^{N-1}\sum_{j=1}^{\ell}|h_i^j|$$

In the last transformation we could eliminate the terms $-\sum_{i=0}^{N-1}p_{i+1}^k\nabla_x f_i\delta x_i^*-\sum_{i=0}^{N-1}p_{i+1}^k\nabla_u f_i\delta u_i^*$ $+\sum_{i=0}^{N}p_i^k\delta x_i^*$ because (7.36) and (7.39) imply that these sum to zero. We used the complementary slackness condition corresponding to the optimal solution of (7.35) - (7.39) which assures that $\lambda_i^j g_i^j(x_i^k+\delta x_i^*,u_i^k+\delta u_i^*)=0$. We used (7.49) - (7.50), and we used (7.46) - (7.47).

Now substituting (7.51), (7.49) and (7.50) into (7.48) we find

$$MF_\gamma(\alpha)\geq MF_\gamma(0)$$
$$-\alpha\left(\sum_{i=0}^{N}\delta x_i^{*\prime}S_i^{xx}\delta x_i^*+2\sum_{i=0}^{N-1}\delta x_i^{*\prime}S_i^{xu}\delta u_i^*\sum_{i=0}^{N-1}\delta u_i^{*\prime}S_i^{uu}\delta u_i^*\right)+o(\alpha)$$

As (7.42) holds we see that there is an $\alpha>0$ such that $MF_\gamma(\alpha)>MF_\gamma(0)$.

Now consider the g_i^j with $(i,j)\notin J^\epsilon$ which were temporarily disregarded. By definition $g_i^j(x_i^k,u_i^k)<-\epsilon$ for such (i,j). Therefore for some $\tilde{\alpha}>0$ we see that for all $\alpha\in(0,\tilde{\alpha})$ we have

$$g_i^j(x_i^k,u_i^k)+\alpha(\nabla_x g_i(x_i^k,u_i^k)\delta x_i+\nabla_u g_i(x_i^k,u_i^k)\delta u_i)\leq 0$$

and these constraints will not contribute to MF_γ for such α. We have $MF_\gamma(\alpha)>MF_\gamma(0)$ for such α, and the proof is complete. □

The step length α is chosen such that at the new point the merit function is suitably improved. Thus with the maximization step length rule, α is chosen such that (7.45) is maximized with respect to α. With the Armijo rule we select parameters $\beta\in(0,1)$, $\sigma\in(0,\frac{1}{2})$ and γ satisfying (7.46) - (7.47). Then m is chosen as the smallest non-negative integer for which

$$MF_\gamma(\alpha)=MF_\gamma(\beta^m)=MF_\gamma(x^{k+1},u^{k+1})\geq MF_\gamma(0) \tag{7.52}$$
$$+\sigma\beta^m(\sum_{i=0}^{N-1}\nabla_x r_i\delta x_i^*+\sum_{i=0}^{N-1}\nabla_u r_i\delta u_i^*+\nabla r_N\delta x_N^*$$
$$-\gamma\sum_{i=0}^{N-1}\sum_{j\in M_i}(\nabla_x g_i^j\delta x_i^*+\nabla_u g_i^j\delta u_i^*)$$
$$-\gamma\sum_{i=0}^{N-1}\sum_{j=1}^{\ell}(\nabla_x h_i^j\delta x_i^*+\nabla_u h_i^j\delta u_i^*))$$

In this formula all function values and all gradients are evaluated at the point (x^k,u^k) (r_N and ∇r_N evaluated at x_N^k) and the index set M_i contains those j for which $g_i^j(x_i^k,u_i^k)>0$.

We can then state an algorithm using the Armijo steplength rule as follows:

Algorithm

Step 0 Choose u^0 and calculate x^0 from (7.34). Choose $\beta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$ and $\epsilon > 0$. Let $k = 0$.

Step 1 Define (7.41) and solve (7.35) - (7.39) to obtain $(\delta x^*, \delta u^*)$, λ and μ.

Step 2 Select γ satisfying (7.46) - (7.47). Let $b = 0$ and repeat incrementing b by one while (7.52) is not fulfilled. In this, the forwards solution $(\tilde{x}, \tilde{u})$ is constructed from (7.43) - (7.44) using $\alpha = \beta^m$ (write $(\tilde{x}, \tilde{u})$ in stead of (x^{k+1}, u^{k+1})).

Step 3 Let $\alpha = \beta^b$ and calculate (x^{k+1}, u^{k+1}) from (7.43) - (7.44). Let $k = k + 1$. Go to Step 1.

Proposition 7.2.2 *Assume that for all i r_i, f_i, g_i, and h_i are continuously differentiable. Apply the above Algorithm. Let S_i^{xx}, S_i^{xu} and S_i^{uu} be chosen such that (7.41) holds. Assume that (7.35) -(7.38) for all k has a feasible (and hence also unique optimal) solution. Assume that there is a compact set such that (x^k, u^k) are contained in this set for all k. Then every limit point satisfies the KKT conditions.*

Proof. The solution to the problem (7.35) - (7.38) and the direction $(\delta x^*, \delta u^*)$ are unique and depend continuously on (x, u) (i.e., on u). Also MF_γ is continuous and it is an ascent function for all u not satisfying the KKT conditions (cf. Luenberger (1989) p. 440). The result then follows by application of the global convergence theorem, Luenberger (1989) p. 187. □

If a feasible solution to the local problem does not exist then the problem may be expanded by introduction of artificial variables as discussed in Section 2.5.

The algorithm can be used in connection with any solution procedure which in a finite number of iterations solves the QLEI (7.35) - (7.39). In particularly DP can be used as explained in Section 7.1. If DP is used, there is available a second way of implementation, viz., DDP.

State-and-control-space Interpolation

A second possibility of forming the QLEI problem is to interpret z in (7.22) - (7.24) as $z = (x_0', u_0', x_1', \ldots, x_{N-1}', u_{N-1}', x_N')'$. In this scheme there are two kinds of equality constraints besides the inequalities:

$$g_i(x_i, u_i) \leq 0 \quad i = 0, \ldots, N-1 \tag{7.53}$$

$$x_{i+1} - f_i(x_i, u_i) = 0 \quad i = 0, \ldots, N-1 \tag{7.54}$$

$$h_i(x_i, u_i) = 0 \quad i = 0, \ldots, N-1 \tag{7.55}$$

The state variables are not attempted eliminated, and therefore if f_i is nonlinear then in general $x_{i+1}^{k+1} \neq f_i(x_i^{k+1}, u_i^{k+1})$ due to the interpolation (7.28), even if $x_{i+1}^k = f_i(x_i^k, u_i^k)$. Therefore (7.36) should be replaced by

$$\delta x_{i+1} = \nabla_x f_i(x_i^k, u_i^k)\delta x_i + \nabla_u f_i(x_i^k, u_i^k)\delta u_i + f_i(x_i^k, u_i^k) - x_{i+1}^k \tag{7.56}$$

As in the Control-space Interpolation described above, the problem (7.35) - (7.39), (7.41) is formulated and the solution $(\delta x_i^*, \delta u_i^*)$ is found. For a given stepsize α the updating is

$$u_i^{k+1} = u_i^k + \alpha \delta u_i^* \tag{7.57}$$

$$x_i^{k+1} = x_i^k + \alpha \delta x_i^* \tag{7.58}$$

The merit function consists of the same terms as in (7.45) plus

$$-\gamma \sum_{i=0}^{N-1} \sum_{j=1}^{n} |(x_{i+1}^j)^{k+1} - f_i^j(x_i^{k+1}, u_i^{k+1})| \tag{7.59}$$

and γ is chosen to satisfy (7.46) - (7.47) as well as

$$\gamma \geq |p_i^j|, \quad i = 1, \ldots, N, \quad j = 1, \ldots, n \tag{7.60}$$

where p_{i+1} is the multiplier to (7.36). Convergence results for this algorithm are the same as in Proposition 7.2.1.

Differential Dynamic Programming - DDP

Consider now the third possibility for application of the linearization method, based on application of DP for solution of the QLEI problem. This will permit modifications both in definition of the QLEI problem (backwards recursion) and in the interpolation to construct the next iteration point (forwards recursion).

DDP: Backwards

In the above the QLEI problem was defined by (7.35) - (7.39) i.e., it is defined before the QLEI algorithm is applied at iteration k. It is also possible to define the QLEI problem recursively backwards during the QLEI algorithm's backwards run. With the backwards construction (7.64) of the QLEI problem and the forwards construction (7.84) - (7.85) we call the method the linearization method with DDP. We proceed as follows. At stage i we have the nominal (x^k, u^k) in addition to Q_{i+1} and P_{i+1}, cf. (4.23) - (4.24) and after (4.43). Then the criterion (7.6) is obtained by performing first a Taylor's expansion to the first order of

$$r_i(x_i, u_i) + f_i(x_i, u_i)' Q_{i+1} f_i(x_i, u_i) + P_{i+1} f_i(x_i, u_i) \tag{7.61}$$

around the point (x^k, u^k). To this are then added the terms involving S_i^{xx}, S_i^{xu} and S_i^{uu}, cf. (7.35). We may now solve the problem (7.6) - (7.9) at stage i to obtain $\underline{K}_i$, $\underline{L}_i$, Q_i and P_i and then proceed to stage $(i-1)$.

The formulation and solution of (7.35) - (7.39) are in terms of changes $(\delta x, \delta u)$ relative to the iteration point (x^k, u^k). In this case (7.61) should more appropriately be written

$$\begin{aligned} r_i(x_i, u_i) + (f_i(x_i, u_i) - f_i(x_i^k, u_i^k))' Q_{i+1}^\delta (f_i(x_i, u_i) - f_i(x_i^k, u_i^k)) \\ + P_{i+1}^\delta (f_i(x_i, u_i) - f_i(x_i^k, u_i^k)) \end{aligned} \tag{7.62}$$

It is seen that $Q_{i+1} = Q_{i+1}^\delta$ and

$$P_{i+1}^\delta = 2 f_i(x_i^k, u_i^k)' Q_{i+1} + P_{i+1} \tag{7.63}$$

With the formulation in $(\delta x, \delta u)$ the local criterion is (the argument (x^k, u^k) and constant are omitted):

$$\begin{aligned} &(\nabla_x r_i + P_{i+1}^\delta \nabla_x f_i) \delta x_i \\ &+ (\nabla_u r_i + P_{i+1}^\delta \nabla_u f_i) \delta u_i + \tfrac{1}{2} \delta x_i' S_i^{xx} \delta x_i + \delta x_i' S_i^{xu} \delta u_i + \tfrac{1}{2} \delta u_i' S_i^{uu} \delta u_i \end{aligned} \tag{7.64}$$

Observe that if p_{i+1} is interpreted as the slope of the approximation π_{i+1} to RUB_{i+1}, i.e., $p_{i+1} = \nabla\pi_{i+1} = P^{\delta}_{i+1}$, then with the usual definition of the Hamiltonian, $H_i = r_i + p_{i+1}f_i$ we see that the linear terms in (7.64) are the familiar ones

$$\nabla_x H_i \delta x_i + \nabla_u H_i \delta u_i \tag{7.65}$$

The left hand sides of (7.7) - (7.9) are obtained by performing a Taylor's expansion to the first order of $g_i(x_i, u_i)$ and $h_i(x_i, u_i)$ around the point (x^k, u^k), such that (7.37) - (7.38) is attained.

We now state for completeness the recursive formulae for the DP solution relative to the formulation using $(\delta x, \delta u)$. We use matrices A^{δ}_i, B^{δ}_i, C^{δ}_i, D^{δ}_i, E^{δ}_i, K^{δ}_i, L^{δ}_i, Q^{δ}_i and P^{δ}_i, cf. (4.16) - (4.24) page 132, and, in case of local constraints, matrices $\underline{K}^{\delta}_i$ and $\underline{L}^{\delta}_i$, cf. (4.41) - (4.43) page 135. We define

$$Q^{\delta}_N = \tfrac{1}{2}\nabla^2 r_N(x^k_N) \tag{7.66}$$
$$P^{\delta}_N = \nabla r_N(x^k_N) \tag{7.67}$$

and then recursively backwards, $i = N-1, \ldots, 0$ (the argument (x^k_i, u^k_i) is omitted):

$$A^{\delta}_i = \tfrac{1}{2}(S^{xx}_i + 2\nabla_x f_i' Q^{\delta}_{i+1} \nabla_x f_i) \tag{7.68}$$
$$B^{\delta}_i = S^{xu}_i + 2\nabla_x f_i' Q^{\delta}_{i+1} \nabla_u f_i \tag{7.69}$$
$$C^{\delta}_i = \tfrac{1}{2}(S^{uu}_i r_i + 2\nabla_u f_i' Q^{\delta}_{i+1} \nabla_u f_i) \tag{7.70}$$
$$D^{\delta}_i = \nabla_u r_i + P^{\delta}_{i+1} \nabla_u f_i \tag{7.71}$$
$$E^{\delta}_i = \nabla_x r_i + P^{\delta}_{i+1} \nabla_x f_i \tag{7.72}$$

such that the local criterion function (7.64) is written

$$\delta x_i' A^{\delta}_i \delta x_i + \delta x_i' B^{\delta}_i \delta u_i + \delta u_i' C^{\delta}_i \delta u_i + D^{\delta}_i \delta u_i + E^{\delta}_i \delta x_i \tag{7.73}$$

Local constraints are written as

$$\nabla_x g_i(x^k_i, u^k_i)\delta x_i + \nabla_u g_i(x^k_i, u^k_i)\delta u_i + g_i(x^k_i, u^k_i) \leq 0 \tag{7.74}$$
$$\nabla_x h_i(x^k_i, u^k_i)\delta x_i + \nabla_u h_i(x^k_i, u^k_i)\delta u_i + h_i(x^k_i, u^k_i) = 0 \tag{7.75}$$

and we then have (the argument (x^k_i, u^k_i) is omitted):

$$\underline{K}^{\delta}_i = \begin{pmatrix} 2C^{\delta}_i & \nabla_u g_i' & \nabla_u h_i' \\ \nabla_u g_i & 0 & 0 \\ \nabla_u h_i & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} D^{\delta\prime}_i \\ -g_i \\ -h_i \end{pmatrix} \tag{7.76}$$

$$\underline{L}^{\delta}_i = \begin{pmatrix} -2C^{\delta}_i & \nabla_u g_i' & \nabla_u h_i' \\ \nabla_u g_i & 0 & 0 \\ \nabla_u h_i & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} B^{\delta\prime}_i \\ -\nabla_x g_i \\ -\nabla_x h_i \end{pmatrix} x_i \tag{7.77}$$

and K^{δ}_i and L^{δ}_i are now defined as the upper m rows of $\underline{K}^{\delta}_i$ and $\underline{L}^{\delta}_i$, respectively. Then finally we define

$$Q^{\delta}_i = A^{\delta}_i + \tfrac{1}{2}(B^{\delta}_i L^{\delta}_i + L^{\delta\prime}_i B^{\delta\prime}_i) + L^{\delta\prime}_i C^{\delta}_i L^{\delta}_i \tag{7.78}$$
$$P^{\delta}_i = K^{\delta\prime}_i B^{\delta\prime}_i + 2K^{\delta\prime}_i C^{\delta}_i L^{\delta}_i + D^{\delta}_i L^{\delta}_i + E^{\delta}_i \tag{7.79}$$

If there are no local constraints the calculation of K_i^δ, L_i^δ, Q_i^δ and P_i^δ simplifies to

$$K_i^\delta = -\tfrac{1}{2}(C_i^\delta)^{-1}D_i^{\delta\prime} \tag{7.80}$$
$$L_i^\delta = -\tfrac{1}{2}(C_i^\delta)^{-1}B_i^{\delta\prime} \tag{7.81}$$
$$Q_i^\delta = A_i^\delta - \tfrac{1}{4}B_i^\delta(C_i^\delta)^{-1}B_i^{\delta\prime} \tag{7.82}$$
$$P_i^\delta = -\tfrac{1}{2}D_i^\delta C_i^{-1}B_i^{\delta\prime} + E_i^\delta \tag{7.83}$$

If all S_i^{xx}, S_i^{xu} and S_i^{uu}, are predefined, the backwards recursion here is of course the same to the one described in relation to problem (7.35) However, as is known from Propositions 4.3.1 and 4.3.2 the essential requirement for the solvability by DP is that $C_i < 0$ on the feasible subspace for all i. Therefore S_i^{xx}, S_i^{xu} and S_i^{uu} may in the DDP approach be defined when solving at stage i with the only requirement that $C_i < 0$ on the feasible subspace, hence this approach is more flexible in application although equivalent in terms of the requirement (7.42).

DDP: Forwards

In the forwards recursion we have the matrices $\underline{K}_i$ and $\underline{L}_i$ from the backwards recursion. We may then in the forwards recursion with restricted step length modify the strategy from (7.16) - (7.17) in Section 7.1 as follows:

$$\begin{pmatrix} u_i^{k+1} \\ \lambda_i^{k+1} \\ \mu_i^{k+1} \end{pmatrix} = \alpha \begin{pmatrix} u_i^o \\ \lambda_i^o \\ \mu_i^o \end{pmatrix} + (1-\alpha)\begin{pmatrix} u_i^k \\ \lambda_i^k \\ \mu_i^k \end{pmatrix} + \underline{L}_i(x_i^{k+1} - x_i^k) \tag{7.84}$$
$$x_{i+1}^{k+1} = f_i(x_i^{k+1}, u_i^{k+1}) \tag{7.85}$$

starting from $x_0^{k+1} = \underline{x}_0$. If all f_i are linear then this is the same as (7.43) - (7.44). Otherwise, u_i^{k+1} from (7.84) is seen to depend on x_{i-1}^{k+1}, while u_i^{k+1} from (7.43) may be calculated independently of x_{i-1}^{k+1}.

In terms of the δ-form, cf. (7.66) - (7.83), (7.84) may be written

$$\begin{pmatrix} u_i^{k+1} \\ \lambda_i^{k+1} \\ \mu_i^{k+1} \end{pmatrix} = \begin{pmatrix} u_i^k \\ \lambda_i^k \\ \mu_i^k \end{pmatrix} + \alpha\underline{K}_i^\delta + \underline{L}_i^\delta(x_i^{k+1} - x_i^k) \tag{7.86}$$

again starting from $x_0^{k+1} = \underline{x}_0$ and updating x according to (7.85).

With the updating (7.84) - (7.85) or (7.85) - (7.86) we use the merit function (7.45). Improvement results for this algorithm are the same as expressed in relation to the previous algorithm, Proposition 7.2.3.

Proposition 7.2.3 *Assume that for all i r_i, f_i, g_i, and h_i are continuously differentiable. Let (x^k, u^k) satisfy (7.34). Assume that (7.35) - (7.39) is solved by DP as explained in Section 7.1. Assume that (7.42) and that the assumptions of the backwards recursion are fulfilled. Assume that in the forwards recursion a positive steplength $\overline{\alpha}$ is found in Step 5 of the Algorithm in Section 7.1. Assume γ chosen such that (7.46) - (7.47) hold. Then there is an α that provides an increase of MF_γ in (7.45).*

Proof. Let $\Delta u_i = u_i^{k+1} - u_i^k$, $\Delta x_i = x_i^{k+1} - x_i^k$. We will first show that $\Delta x_i = \alpha\delta x_i^* + o_i^x(\alpha)$, where $o_i^x(\alpha)/\alpha \to 0$ as $\alpha \to 0$, and where δx_i^* is the solution to (7.35) - (7.39); in (7.35) - (7.39)

the same matrices S_i^{xx}, S_i^{xu} and S_i^{uu} are used as applied here in the DDP backwards recursion, and therefore the two solutions will be identical since the same QLE problem is solved. Obviously the desired expression holds for $i = 0$ due to the initialization of (7.84) - (7.85). Now assume the expression holds for a given i. We then have

$$\begin{aligned}
&\Delta x_{i+1} = x_i^{k+1} - x_i^k = f_i(x_i^{k+1}, u_i^{k+1}) - f_i(x_i^k, u_i^k) \\
&= \nabla_x f_i(x_i^k, u_i^k)\Delta x_i + \nabla_u f_i(x_i^k, u_i^k)(\alpha(u_i^o - u_i^k) + L_i\Delta x_i) + o_i(\Delta x_i, \Delta u_i) \\
&= \nabla_x f_i(x_i^k, u_i^k)(\alpha\delta x_i^* + o_i^x(\alpha)) + \nabla_u f_i(x_i^k, u_i^k)(\alpha(u_i^o - u_i^k) + L_i(\delta x_i^* + o_i^x(\alpha))) \\
&+ o_i(\alpha\delta x_i^* + o_i^x(\alpha), \alpha\delta u_i^*) \\
&= \alpha(\nabla_x f_i(x_i^k, u_i^k)\delta x_i^* + \nabla_u f_i(x_i^k, u_i^k)\delta u_i^*) + o_{i+1}^x(\alpha) \\
&= \alpha\delta x_{i+1}^* + o_{i+1}^x(\alpha)
\end{aligned}$$

where $o_i(\Delta x_i, \Delta u_i)/(\| \Delta x_i \| + \| \Delta u_i \|) \to 0$ as $(\| \Delta x_i \| + \| \Delta u_i \|) \to 0$ and $o_{i+1}^x(\alpha)/\alpha \to 0$ as $\alpha \to 0$. It follows that the expression holds for all i.

With this, the conclusion can be reached by repeating the argumentation of the proof of Proposition 7.2.1. □

Now assume that g_i and h_i are additively separable with respect to x_i and u_i, and linear with respect to u_i. These constraints can then be written as

$$g_i^x(x_i) + G_i^u u_i \leq 0 \tag{7.87}$$

$$h_i^x(x_i) + H_i^u u_i = 0 \tag{7.88}$$

Then we observe that the local constraints (7.87) for which $(i,j) \in J^A$ and the constraints (7.88) may automatically be fulfilled at the new point (x^{k+1}, u^{k+1}), provided we modify the formulae corresponding to (7.84) - (7.85) as follows:

$$\begin{aligned}
\begin{pmatrix} u_i^{k+1} \\ \lambda_i^{k+1} \\ \mu_i^{k+1} \end{pmatrix} &= \alpha \begin{pmatrix} -2C_i^k & \hat{G}_i^{w\prime} & H_i^{w\prime} \\ \hat{G}_i^u & 0 & 0 \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} D_i + B_i' x_i^k \\ -\hat{g}_i^x(x_i^k) \\ h_i^x(x_i^k) \end{pmatrix} \\
&+ (1-\alpha) \begin{pmatrix} u_i^k \\ \lambda_i^k \\ \mu_i^k \end{pmatrix} \\
&+ \begin{pmatrix} -2C_i^k & \hat{G}_i^{w\prime} & H_i^{w\prime} \\ \hat{G}_i^u & 0 & 0 \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} B_i'(x_i^{k+1} - x_i^k) \\ \hat{g}_i^x(x_i^{k+1}) - \hat{g}_i^x(x_i^k) \\ h_i^x(x_i^{k+1}) - h_i^x(x_i^k) \end{pmatrix}
\end{aligned} \tag{7.89}$$

$$x_{i+1}^{k+1} = f_i(x_i^{k+1}, u_i^{k+1}) \tag{7.90}$$

where $\hat{g}_i^x$ are the rows of g_i^x which correspond to active inequality constraints.

Therefore in this case the terms $\gamma|g|_+$ and $\gamma|h|$ in (7.45) are not necessary, and we use (7.33): $r(u) = \sum_{i=0}^N r_i$ as merit function. If g_i and h_i are linear with respect to both x_i and u_i then (7.89) is identical to (7.84).

Proposition 7.2.4 *Assume that for all i r_i, f_i, g_i, and h_i are continuously differentiable. Let (x^k, u^k) be calculated from (7.34) and assume that $g_i(x_i^k, u_i^k) \leq 0$ and $h_i(x_i^k, u_i^k) = 0$ for $i = 0, \ldots, N-1$. Assume $(\delta x^*, \delta u^*)$ is the unique nonzero solution to (7.35) - (7.39) or that an*

improving direction is found as described in Proposition 7.2.3. Assume (7.42) holds. If the local constraints are of the form (7.87) - (7.88) and (x^{k+1}, u^{k+1}) *is calculated from (7.89) - (7.90) then there is an* α *that provides an increase of* $\sum_{i=0}^{N} r_i$.

Proof. Due to the continuous differentiability of g_i and h_i we will again attain relations as in the proof of Proposition 7.2.3, and therefore the result follows as in that proof. □

The aim of keeping the local constraints always fulfilled may also be reached if the local constraints are nonlinear and nonseparible between x_i and u_i, provided only that the local problem may be solved in this case.

An initial solution (x^0, u^0) satisfying the dynamical equation can in the case (7.87) - (7.88) be conveniently constructed recursively forwards as follows. Let $x_0^0 = \underline{x}_0$ and choose u_0^0 such that (7.87) - (7.88) are satisfied for $i = 0$. Such u_0^0 can easily be found (provided one exists), because the constraint are linear with respect to u_0. With this u_0^0 calculate $x_1^0 = f_0(x_0^0, u_0^0)$. Then find u_1^0 satisfying (7.87) - (7.88), etc. If it is not possible to find an u_i^0 such that the local constraints (7.87) - (7.88) are fulfilled, then artificial variables may be introduced as discussed in Section 2.5.

Now we can specify an algorithm as follows:

Algorithm

Step 0 Choose u^0 and calculate x^0 from (7.34). Choose $\beta \in (0,1)$, $\sigma \in (0, \frac{1}{2})$ and $\epsilon > 0$. Let $k = 0$.

Step 1 Find an improving direction as described in Proposition 7.2.3 or Proposition 7.2.4.

Step 2 Select γ satisfying (7.46) - (7.47) (in case that (x_i^k, u_i^k) is feasible and (7.87) - (7.90) apply we may take $\gamma = 0$). Let $b = 0$ and repeat incrementing b by one while (7.52) is not fulfilled. In this the forwards solution $(\tilde{x}, \tilde{u})$ is constructed from (7.84) - (7.85) (write $(\tilde{x}, \tilde{u})$ in stead of (x^{k+1}, u^{k+1})) (in case that (7.87) - (7.88) apply from (7.89) - (7.90)).

Step 3 Let $\alpha = \beta^b$ and calculate (x^{k+1}, u^{k+1}) from (7.84) - (7.85). Let $k = k + 1$. Go to Step 1.

Convergence results are the same as in Proposition 7.2.2.

Conclusions

Three different approaches towards attaining global convergence have been presented, and algorithms based thereon have been formulated. All three algorithms have global convergence under the same assumptions. In all three algorithms the same type of QLEI problem (7.35) - (7.39) has to be solved in the backwards recursion. The problems need not be exactly the same in all three algorithms. However, there is no reason to believe that they will be very dissimilar. The difference is therefore in the updating procedures. The updating in the control-and-state-space interpolation seems to be the simplest one. However, the saving is marginal, since f_i must be evaluated in order to find (7.59). In the control-and-state-space interpolation the dynamics will not in general be fulfilled, if f_i are nonlinear. In the control-space and the DDP forwards interpolations the dynamics will be fulfilled, once a feasible trajectory is found. The local constraints will in general not be fulfilled in any of the methods. If the local constraints are additively separable in x_i and u_i and

linear in u_i then the local constraints may be fulfilled during all iterations, once a feasible solution has been found, if the DDP updating is used in the forwards recursions.

7.3 Newton's Method

In this section we develop a Newton method for solution of the optimal control problem. We also show how to combine it with the linearization method of Section 7.2 to obtain global convergence. As in Section 7.2 we consider different applications to the OCP. Thus, in this section we consider control-space and state-and-control-space interpolations, while the DDP interpolation will be treated in the next section.

Control-space Interpolation

The Locally Unconstrained Problem

As in Section 7.2 page 197 we use first the version with elimination of the state variables. We first state the idea for the locally unconstrained problem:

$$\begin{aligned} & r(u) = \\ & r_0(\underline{x}_0, u_0) + r_1(f_0(\underline{x}_0, u_0), u_1) + \dots \\ & \dots + r_N(f_{N-1}(\dots(f_0(\underline{x}_0, u_0), u_1)\dots), u_{N-1}) \end{aligned} \tag{7.91}$$

Let a nominal point (x^o, u^o) be given satisfying the dynamic equation and $x_0^o = \underline{x}_0$. Then define $p_N = \nabla r_N(x_N)$ and recursively backwards

$$p_i = \nabla_x(r_i(x_i^o, u_i^o) + p_{i+1} f_i(x_i^o, u_i^o)) \tag{7.92}$$

Further define the matrices

$$S_N^{xx} = \nabla_{xx}^2 r_N(x_N^o) \tag{7.93}$$
$$S_i^{xx} = \nabla_{xx}^2(r_i(x_i^o, u_i^o) + p_{i+1} f_i(x_i^o, u_i^o)) \tag{7.94}$$
$$S_i^{xu} = \nabla_{xu}^2(r_i(x_i^o, u_i^o) + p_{i+1} f_i(x_i^o, u_i^o)) \tag{7.95}$$
$$S_i^{uu} = \nabla_{uu}^2(r_i(x_i^o, u_i^o) + p_{i+1} f_i(x_i^o, u_i^o)) \tag{7.96}$$

Using the Hamiltonian $H_i = r_i + p_{i+1} f_i$, $H_N = r_N$, the expressions (7.92) - (7.96) may be written

$$p_i = \nabla_x H_i(x_i^o, u_i^o, p_{i+1}^o) \tag{7.97}$$
$$S_N^{xx} = \nabla^2 r_N(x_N^o) \tag{7.98}$$
$$S_i^{xx} = \nabla_{xx}^2 H_i(x_i^o, u_i^o, p_{i+1}^o) \tag{7.99}$$
$$S_i^{xu} = \nabla_{xu}^2 H_i(x_i^o, u_i^o, p_{i+1}^o) \tag{7.100}$$
$$S_i^{uu} = \nabla_{uu}^2 H_i(x_i^o, u_i^o, p_{i+1}^o) \tag{7.101}$$

Finally set up and solve the problem

$$\max_{\delta x, \delta u}[\sum_{i=0}^{N-1}(\nabla_x r_i(x_i^o, u_i^o)\delta x_i + \nabla_u r_i(x_i^o, u_i^o)\delta u_i \tag{7.102}$$

$$+\tfrac{1}{2}\delta x_i' S_i^{xx}\delta x_i + \delta x_i' S_i^{xu}\delta u_i + \tfrac{1}{2}\delta u_i' S_i^{uu}\delta u_i)$$
$$+\nabla r_N(x_N^o)\delta x_N + \tfrac{1}{2}\delta x_N' S_N^{xx}\delta x_N]$$
$$\delta x_{i+1} = \nabla_x f_i(x_i^o, u_i^o)\delta x_i + \nabla_u f_i(x_i^o, u_i^o)\delta u_i \tag{7.103}$$
$$\delta x_0 = 0 \tag{7.104}$$

Let the solution be $(\delta x^*, \delta u^*)$. Then the new point u^D in the control space is found as

$$u^D = u^o + \delta u^* \tag{7.105}$$

The new states x^D corresponding to this may then be found by using the dynamic equation recursively forwards from $\underline{x}_0$ using u^D (use the original dynamic equation, not (7.103)).

This can be compared to the Newton step applied directly on the criterion (7.91) where the new control u^N is found as

$$u^N = u^o - (\nabla^2 r(u^o))^{-1}\nabla r(u^o)' \tag{7.106}$$

and the corresponding new states are then found from the original dynamic equation using u^N.

If $u^N = u^D$ then clearly also the corresponding states will be identical.

Proposition 7.3.1 *Assume that r_i and f_i are twice continuously differentiable and that $\nabla^2 r(u^o) < 0$. Then with definitions (7.105) and (7.106) there holds $u^N = u^D$.*

Proof. In the following all the function values and derivatives are calculated at (x^o, u^o). We have

$$\frac{\partial r}{\partial u_k} = \nabla_u r_k + \nabla_x r_{k+1}\nabla_u f_k + \nabla_x r_{k+2}\nabla_x f_{k+1}\nabla_u f_k \tag{7.107}$$
$$+\nabla_x r_{k+3}\nabla_x f_{k+2}\nabla_x f_{k+1}\nabla_u f_k$$
$$+\ldots+\nabla_x r_N \nabla_x f_{N-1}\ldots\nabla_x f_{k+1}\nabla_u f_k$$

We observe that by (7.92) we have

$$p_{i+1} = \nabla_x r_{i+1} + \nabla_x r_{i+2}\nabla_x f_{i+1} + \nabla_x r_{i+3}\nabla_x f_{i+2}\nabla_x f_{i+1} + \tag{7.108}$$
$$\ldots+\nabla_x r_N \nabla_x f_{N-1}\ldots\nabla_x f_{i+1}$$

Therefore also

$$\frac{\partial r}{\partial u_k} = \nabla_u H_k$$

as already observed in Proposition 6.1.2.

Now we calculate $\partial^2 r/\partial u_j \partial u_k = \partial^2 r/\partial u_k \partial u_j$. First we let $j = k$ and find

$$\frac{\partial^2 r}{\partial u_k \partial u_k} = \frac{\partial(\text{expression (7.107)})}{\partial u_k}$$
$$= \nabla_{uu}^2 r_k$$
$$+ \nabla_u f_k' \nabla_{xx}^2 r_{k+1}\nabla_u f_k + \nabla_x r_{k+1}\nabla_{uu}^2 f_k$$
$$+ \nabla_u f_k' \nabla_x f_{k+1}' \nabla_{xx}^2 r_{k+2}\nabla_x f_{k+1}\nabla_u f_k$$
$$+\nabla_u f_k' \nabla_x r_{k+2}\nabla_{xx}^2 f_{k+1}\nabla_u f_k + \nabla_x r_{k+2}\nabla_x f_{k+1}\nabla_{uu}^2 f_k$$

$$
\begin{aligned}
+ \quad & \nabla_u f_k' \nabla_x f_{k+1}' \nabla_x f_{k+2}' \nabla_{xx}^2 r_{k+3} \nabla_x f_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& + \nabla_u f_k' \nabla_x f_{k+1}' \nabla_x r_{k+3} \nabla_{xx}^2 f_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\
& + \nabla_u f_k' \nabla_x r_{k+3} \nabla_x f_{k+2} \nabla_{xx}^2 f_{k+1} \nabla_u f_k \\
& + \nabla_x r_{k+3} \nabla_x f_{k+2} \nabla_x f_{k+1} \nabla_{uu}^2 f_k \\
+ \quad & \\
& \vdots \\
+ \quad & \nabla_u f_k' \nabla_x f_{k+1}' \cdots \nabla_x f_{N-1}' \nabla_{xx}^2 r_N \nabla_x f_{N-1} \cdots \nabla_x f_{k+1} \nabla_u f_k \\
& + \\
& \vdots \\
& + \nabla_x r_N \nabla_x f_{N-1} \cdots \nabla_x f_{k+1} \nabla_{uu}^2 f_k
\end{aligned}
$$

By $\nabla_{xx}^2 f_k$ we mean the set of n matrices $\nabla_{xx} f_k^j$, $j = 1, \ldots, n$, arranged in a column. Thus, with p a row vector with n components we have $p \nabla_{xx} f_k = \sum_{j=1}^n p^j \nabla_{xx} f_k^j = \nabla_{xx}(p f_k)$. Similarly, $p \nabla_{uu} f_k = \sum_{j=1}^n p \nabla_{uu} f_k^j = \nabla_{uu}(p f_k)$.

Using (7.108) we can rearrange the above expressions as follows:

$$
\begin{aligned}
\frac{\partial^2 r}{\partial u_k \partial u_k} = \quad & \nabla_{uu}^2 (r_k + p_{k+1} f_k) \\
+ \quad & \nabla_u f_k' (\nabla_{xx}^2 (r_{k+1} + p_{k+2} \nabla_{xx}^2 f_{k+1})) \nabla_u f_k \\
+ \quad & \nabla_u f_k' \nabla_x f_{k+1}' (\nabla_{xx}^2 (r_{k+2} + p_{k+3} \nabla_{xx}^2 f_{k+2})) \nabla_x f_{k+1} \nabla_u f_k \\
+ \quad & \\
& \vdots \\
+ \quad & \nabla_u f_k \cdots \nabla_x f_{N-2}' (\nabla_{xx}^2 (r_{N-1} + p_N f_{N-1})) \nabla_x f_{N-2} \cdots \nabla_u f_k \\
+ \quad & \nabla_u f_k' \cdots \nabla_x f_{N-1}' \nabla^2 r_N \nabla_x f_{N-1} \cdots \nabla_u f_k
\end{aligned}
$$

Now introduce $\delta \tilde{u}_k$ and pre- and post-multiply with $\delta \tilde{u}_k'$ and $\delta \tilde{u}_k$, respectively. We then get the following expression where we have indicated a $\delta \tilde{x}_j$ and the S_i's from (7.93) - (7.96):

$$
\begin{aligned}
& \delta \tilde{u}_k' \frac{\partial^2 r}{\partial u_k \partial u_k} \delta \tilde{u}_k = \delta \tilde{u}_k' \overbrace{\nabla_{uu}^2 (r_k + p_{k+1} f_k)}^{S_k^{uu}} \delta \tilde{u}_k \\
+ \quad & \overbrace{\delta \tilde{u}_k' \nabla_u f_k'}^{\delta \tilde{x}_{k+1}'} \overbrace{\nabla_{xx}^2 (r_{k+1} + p_{k+2} f_{k+1})}^{S_{k+1}^{xx}} \overbrace{\nabla_u f_k \delta \tilde{u}_k}^{\delta \tilde{x}_{k+1}} \\
+ \quad & \overbrace{\delta \tilde{u}_k' \nabla_u f_k' \nabla_x f_{k+1}'}^{\delta \tilde{x}_{k+2}'} \overbrace{\nabla_{xx}^2 (r_{k+2} + p_{k+3} f_{k+2})}^{S_{k+2}^{xx}} \overbrace{\nabla_x f_{k+1} \nabla_u f_k \delta \tilde{u}_k}^{\delta \tilde{x}_{k+2}} \\
+ \quad & \\
& \vdots \\
+ \quad & \overbrace{\delta \tilde{u}_k' \nabla_u f_k \cdots \nabla_x f_{N-2}'}^{\delta \tilde{x}_{N-1}'} \overbrace{\nabla_{xx}^2 (r_{N-1} + p_N f_{N-1})}^{S_{N-1}^{xx}} \overbrace{\nabla_x f_{N-2} \cdots \nabla_u f_k \delta \tilde{u}_k}^{\delta \tilde{x}_{N-1}} \\
+ \quad & \overbrace{\delta \tilde{u}_k' \nabla_u f_k' \cdots \nabla_x f_{N-1}'}^{\delta \tilde{x}_N'} \overbrace{\nabla^2 r_N}^{S_N^{xx}} \overbrace{\nabla_x f_{N-1} \cdots \nabla_u f_k \delta \tilde{u}_k}^{\delta \tilde{x}_N}
\end{aligned}
$$

From this we see that if $\delta\tilde{u}_j = 0$ for $j \neq k$, $\delta\tilde{x}_j$ satisfies (7.103) - (7.104) and S_i satisfies (7.93) - (7.96) then we have

$$\begin{aligned}&\delta\tilde{u}_k' \frac{\partial^2 r}{\partial u_k \partial u_k} \delta\tilde{u}_k \\ &= \delta\tilde{u}_k' S_k^{uu} \delta\tilde{u}_k + \sum_{i=0}^{N-1} \delta\tilde{x}_i' S_i^{xx} \delta\tilde{x}_i + \delta\tilde{x}_N' S_N^{xx} \delta\tilde{x}_N\end{aligned}$$

Second, we calculate $\partial^2 r / \partial u_j \partial u_k$ for $j < k$:

$$\begin{aligned}\frac{\partial^2 r}{\partial u_k \partial u_j} = & \quad \frac{\partial(\text{expression (7.107)})}{\partial u_j} \\ = & \quad \nabla_u f_j' \nabla_x f_{j+1}' \nabla_x f_{j+2}' \cdots \nabla_x f_{k-2}' \nabla_x f_{k-1}' \nabla_{xu}^2 r_k \\ + & \quad \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_k' \nabla_{xx}^2 r_{k+1} \nabla_u f_k \\ & \qquad + \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_{k-1}' \nabla_{xu}^2 f_k \nabla_x r_{k+1} \\ + & \quad \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_{k+1}' \nabla_{xx}^2 r_{k+2} \nabla_x f_{k+1} \nabla_u f_k \\ & \qquad + \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_k' \nabla_{xx}^2 f_{k+1} \nabla_x r_{k+2} \nabla_u f_k \\ & \qquad + \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_{k-1}' \nabla_{xu}^2 f_k \nabla_x f_{k+1} \nabla_x r_{k+2} \\ + & \\ & \vdots \\ + & \quad \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_{xx}^2 r_N \nabla_x f_{N-1} \cdots \\ & \qquad + \\ & \qquad \vdots \\ & \qquad + \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x f_{N-2}' \nabla_{xu}^2 f_{N-1} \nabla_x r_N\end{aligned}$$

As above, $\nabla_{xu} f_k$ is interpreted such that $p \nabla_{xu} f_k = \sum_{j=1}^n p^j \nabla_{xu} f_k^j = \nabla_{xu}(p f_k)$. This expression can be rewritten similarly to the case with $j = k$. We pre- and post-multiply with $\delta\hat{u}_j'$ and $\delta\tilde{u}_k$, respectively and get:

$$\begin{aligned}&\delta\hat{u}_j' \frac{\partial^2 r}{\partial u_j \partial u_k} \delta\tilde{u}_k = \\ &\overbrace{\delta\hat{u}_j' \nabla_u f_j' \nabla_x \hat{x} f_{j+1}' \cdots \nabla_x f_{k-1}'}^{\delta\hat{x}_k'} \overbrace{\nabla_{xu}^2 (r_k + p_{k+1} f_k)}^{S_k^{xu}} \delta\tilde{u}_k \\ + & \overbrace{\delta\hat{u}_j' \nabla_u f_j' \nabla_x f_{j+1}' \cdots \nabla_x \hat{x} f_k'}^{\delta\hat{x}_{k+1}'} \overbrace{\nabla_{xx}^2 (r_{k+1} + p_{k+2} f_{k+1})}^{S_{k+1}^{xx}} \overbrace{\nabla_u f_k \delta\tilde{u}_k}^{\delta\tilde{x}_{k+1}} \\ + & \overbrace{\delta\hat{u}_j' \nabla_u f_j' \cdots \nabla_x f_{k+1}'}^{\delta\hat{x}_{k+2}'} \overbrace{\nabla_{xx}^2 (r_{k+2} + p_{k+3} f_{k+2})}^{S_{k+2}^{xx}} \overbrace{\nabla_x f_{k+1} \nabla_u f_k \delta\tilde{u}_k}^{\delta\tilde{x}_{k+2}} \\ + & \end{aligned}$$

$$\vdots$$

$$+ \quad \overbrace{\delta\hat{u}_j'\nabla_u f_j \dots \nabla_x f_{N-2}'}^{\delta\hat{x}_{N-1}'} \overbrace{\nabla_{xx}^2(r_{N-1}+p_N f_{N-1})}^{S_{N-1}^{xx}} \overbrace{\nabla_x f_{N-2} \dots \nabla_u f_k \delta\tilde{u}_k}^{\delta\tilde{x}_{N-1}}$$

$$+ \quad \overbrace{\delta\hat{u}_k'\nabla_u f_k' \dots \nabla_x f_{N-1}'}^{\delta\hat{x}_N'} \overbrace{\nabla^2 r_N}^{S_N^{xx}} \overbrace{\nabla_x f_{N-1} \dots \nabla_u f_k \delta\tilde{u}_k}^{\delta\tilde{x}_N}$$

As seen we have defined $\delta\hat{x}_j$ and $\delta\tilde{x}_j$, corresponding to $\delta\hat{u}_j$ and $\delta\tilde{u}_j$, respectively. Now combining the expressions obtained we get

$$\begin{aligned}
&\delta\hat{u}_j'\frac{\partial^2 r}{\partial u_j \partial u_j}\delta\hat{u}_j + \delta\tilde{u}_k'\frac{\partial^2 r}{\partial u_k \partial u_k}\delta\tilde{u}_k \\
&+\delta\hat{u}_j'\frac{\partial^2 r}{\partial u_j \partial u_k}\delta\tilde{u}_k + \delta\tilde{u}_k'\frac{\partial^2 r}{\partial u_k \partial u_j}\delta\hat{u}_j = \\
&\delta\hat{u}_j' S_j^{uu}\delta\hat{u}_j + \sum_{i=j}^{k-1}\delta\hat{x}_i' S_i^{xx}\delta\hat{x}_i + \delta\hat{u}_k' S_k^{uu}\delta\hat{u}_j + \delta\hat{x}_k' S_k^{xu}\delta\tilde{u}_k \\
&+ \sum_{i=k+1}^{N}\delta\hat{x}_i' S_i^{xx}\delta\hat{x}_i + \sum_{i=k+1}^{N}\delta\tilde{x}_i' S_i^{xx}\delta\tilde{x}_i + 2\sum_{i=k+1}^{N}\delta\tilde{x}_i' S_i^{xx}\delta\hat{x}_i = \\
&\delta\hat{u}_j' S_j^{uu}\delta\hat{u}_j + \sum_{i=j}^{k-1}\delta\hat{x}_i' S_i^{xx}\delta\hat{x}_i + \delta\hat{u}_k' S_k^{uu}\delta\hat{u}_j + \delta\hat{x}_k' S_k^{xu}\delta\tilde{u}_k \\
&+ \sum_{i=k+1}^{N}(\delta\tilde{x}_i + \delta\hat{x}_i)' S_i^{xx}(\delta\hat{x}_i + \delta\tilde{x}_i)
\end{aligned}$$

We see in conclusion that for arbitrary j, k, δu_j and δu_k we have

$$\begin{aligned}
&\delta u_j'\frac{\partial^2 r}{\partial u_j \partial u_j}\delta u_j + \delta u_k'\frac{\partial^2 r}{\partial u_k \partial u_k}\delta u_k \\
&+\delta u_j'\frac{\partial^2 r}{\partial u_j \partial u_k}\delta u_k + \delta u_k'\frac{\partial^2 r}{\partial u_k \partial u_j}\delta u_j = \\
&\sum_{i=0}^{N-1}(\delta x_i' S_i^{xx}\delta x_i + 2\delta x_i' S_i^{xu}\delta u_i + \delta u_i' S_i^{uu}\delta u_i) + \delta x_N' S_N^{xx}\delta x_N
\end{aligned}$$

provided $\delta u_i = 0$ for $i \neq j$ and $i \neq k$, δx satisfies (7.103) - (7.104) and S satisfies (7.92) - (7.96).

Moreover we have from (7.107) that with the same conditions on δx and δu

$$\frac{\partial r}{\partial u_j}\delta u_j = \sum_{i=0}^{N-1}(\nabla_x r_i \delta x_i + \nabla_u r_i \delta u_i) + \nabla r_N \delta x_N$$

Since j and k are arbitrary we can for any δu get under the same assumptions that the second order expansion $(\nabla r \delta u + \frac{1}{2}\delta u' \nabla^2 r \delta u)$ of (7.91) for any δu attains exactly the same value as the expression in [ ] in (7.102) provided δx and δu satisfy (7.103) - (7.104). As the Newton step (7.106) is well defined because $\nabla^2 r < 0$ and represents the optimal solution to the second order expansion

of r and the step (7.105) similarly is well defined, cf. Proposition 4.3.2, and represents the optimal solution to an equivalent expression, the two steps are identical. □

Local Constraints

Now introduce the usual local constraints at stages $i = 0, \ldots, N-1$. Newton's method can be adapted to this in the following way which is sometimes called Wilson's method. Define the Lagrangian to the problem (7.22) - (7.24) in the control variables, cf. (7.91), as

$$L(u; \lambda, \mu) = r(u) - \lambda g(u) - \mu h(u) \tag{7.109}$$

where λ and μ correspond to (7.23) - (7.24). The first order conditions for optimality of δz^* in the problem (7.25) - (7.27) with $S = \nabla_{uu} L$ may be expressed as

$$\nabla_u L + \delta z^* \nabla_{uu} L = 0 \tag{7.110}$$

in addition to feasibility in (7.23) - (7.24). Let a (u^k, λ^k, μ^k) be given. Now the Newton iteration corresponding to (7.106) is

$$\begin{pmatrix} u^N \\ \lambda^N \\ \mu^N \end{pmatrix} = \begin{pmatrix} u^k \\ \lambda^k \\ \mu^k \end{pmatrix} - \begin{pmatrix} -\nabla^2_{uu} L & \nabla g' & \nabla h' \\ \nabla g & 0 & 0 \\ \nabla h & 0 & 0 \end{pmatrix}^{-1} \begin{pmatrix} \nabla_u L \\ g \\ h \end{pmatrix} \tag{7.111}$$

It is here assumed that all inequality constraints that are not supposed to be active are disregarded.

We shall now develop the iteration corresponding to (7.92) - (7.105) for the constrained problem. Therefore define

$$p_N = \nabla r_N(x_N^k) \tag{7.112}$$

and recursively backwards

$$p_i = \nabla_x (r_i(x_i^k, u_i^k) + p_{i+1} f_i(x_i^k, u_i^k) - \lambda_i^k g_i(x_i^k, u_i^k) - \mu_i^k h_i(x_i^k, u_i^k)) \tag{7.113}$$

(These p_i are in fact, together with λ_i^k and μ_i^k, the KKT multipliers corresponding to the optimal solutions of (7.118) - (7.122) at the previous iteration. If the method of solution of (7.118) - (7.122) uses p then (7.112) - (7.113) are not necessary.)

Then define the matrices

$$S_N^{xx} = \nabla^2_{xx} r_N(x_N^k) \tag{7.114}$$

$$\begin{aligned} S_i^{xx} = {} & \nabla^2_{xx} (r_i(x_i^k, u_i^k) + p_{i+1} f_i(x_i^k, u_i^k) \\ & - \lambda_i g_i(x_i^k, u_i^k) - \mu_i g_i(x_i^k, u_i^k)) \end{aligned} \tag{7.115}$$

$$\begin{aligned} S_i^{xu} = {} & \nabla^2_{xu} (r_i(x_i^k, u_i^k) + p_{i+1} f_i(x_i^k, u_i^k) \\ & - \lambda_i g_i(x_i^k, u_i^k) - \mu_i g_i(x_i^k, u_i^k)) \end{aligned} \tag{7.116}$$

$$\begin{aligned} S_i^{uu} = {} & \nabla^2_{uu} (r_i(x_i^k, u_i^k) + p_{i+1} f_i(x_i^k, u_i^k) \\ & - \lambda_i g_i(x_i^k, u_i^k) - \mu_i g_i(x_i^k, u_i^k)) \end{aligned} \tag{7.117}$$

Finally set up and solve the problem

$$\max_{\delta x,\delta u}[\sum_{i=0}^{N-1}(\nabla_x r_i(x_i^k,u_i^k)\delta x_i+\nabla_u r_i(x_i^k,u_i^k)\delta u_i \tag{7.118}$$
$$+\tfrac{1}{2}\delta x_i' S_i^{xx}\delta x_i+\delta x_i' S_i^{xu}\delta u_i+\tfrac{1}{2}\delta u_i' S_i^{uu}\delta u_i)$$
$$+\nabla r_N(x_N^k)\delta x_N+\tfrac{1}{2}\delta x_N' S_N^{xx}\delta x_N]$$
$$\delta x_{i+1}=\nabla_x f_i(x_i^k,u_i^k)\delta x_i+\nabla_u f_i(x_i^k,u_i^k)\delta u_i \tag{7.119}$$
$$g_i(x_i^k,u_i^k)+\nabla_x g_i(x_i^k,u_i^k)\delta x_i+\nabla_u g_i(x_i^k,u_i^k)\delta u_i=0 \tag{7.120}$$
$$h_i(x_i^k,u_i^k)+\nabla_x h_i(x_i^k,u_i^k)\delta x_i+\nabla_u h_i(x_i^k,u_i^k)\delta u_i=0 \tag{7.121}$$
$$\delta x_0=0 \tag{7.122}$$

Let the solution be $(\delta x^*,\delta u^*)$. Then the new point u^D in the control space is found as in (7.105), the new (λ^D,μ^D) are the Lagrange multipliers corresponding to the solution to (7.118) - (7.122) and the new states are found by the original dynamic equation using u^D.

Proposition 7.3.2 *Assume that r_i and f_i are twice continuously differentiable, that the rows of $(\nabla g',\nabla h')'$ in (7.111) are linearly independent, and that $\nabla^2_{uu}L$ in (7.111) is negative definit on the subspace defined by the active constraints. Assume that the same inequalities g_i^j appear in (7.111) and (7.120). Then $u^N=u^D$, $\lambda^N=\lambda^D$ and $\mu^N=\mu^D$.*

Proof. The proof may be performed essentially as the proof of Proposition 7.3.1. □

Observe that the requirement of $(\nabla g',\nabla h')'$ being linearly independent is not sufficient to guarantee that DP may be used for solution of (7.118) - (7.122). DP requires $(\nabla_u g_i',\nabla_u h_i')'$ linearly independent for all i, cf. Proposition 4.3.2.

We can now modify the linearization algorithm of Section 7.2 to obtain quadratic rate of local convergence.

Algorithm

Step 0 Choose u^0 and calculate x^0 from (7.34). Let $\lambda^0=0$, $\mu^0=0$. Choose $\beta\in(0,1)$, $\sigma\in(0,0.5)$ and $\epsilon>0$. Let $k=0$.

Step 1 Define (7.118) - (7.122) using (7.112) - (7.117). Solve (7.118) - (7.122) to obtain $(\delta x^*,\delta u^*,\delta\lambda^*,\delta\mu^*)$, modifying, if necessary, S_i^{uu} such that the criterion function (7.118) becomes negative definite on the feasible subspace (7.119) - (7.122) (if DP is used, modify, if necessary, S_i^{uu} such that C_i becomes negative definite on the subspace).

Step 2 Select γ satisfying (7.46) - (7.47). Let $m=0$ and repeat incrementing m by one while (7.52) is not fulfilled. In this the forwards solution $(\tilde{x},\tilde{u})$ is constructed from (7.43) - (7.44) using $\alpha=\beta^m$ (write $(\tilde{x},\tilde{u})$ in stead of (x^{k+1},u^{k+1})).

Step 3 Let $\alpha=\beta^m$, calculate (x^{k+1},u^{k+1}) from (7.43) - (7.44), let $\lambda^{k+1}=\lambda^k+\alpha\delta\lambda^*$, $\mu^{k+1}=\mu^k+\alpha\delta\mu^*$. Let $k=k+1$. Go to Step 1.

Proposition 7.3.3 *Under the assumptions of Proposition 7.2.2 the algorithm converges as stated there.*

Now in addition assume that r_i, f_i, g_i and h_i are twice continuously differentiable. Assume that the algorithm converges to a unique local maximum where the following holds for all $k > \underline{k}$, where $\underline{k}$ is an iteration number:

- *For all active g_i^j the corresponding λ_i^j are strictly positive (strict complementarity)*
- *The rows of $(\nabla g', \nabla h')'$ are linearly independent (if DP is used, the rows of $(\nabla_u g_i', \nabla_u h_i')'$ are linearly independent) (only active g_i^j are considered)*
- *The modification of S_i^{uu} in Step 1 is not performed.*
- *The steplength $\alpha = 1$ is accepted in Step 2.*

Then there is an itertion number $\overline{k}$ such that for $k > \overline{k}$ the algorithm converges at a quadratic rate.

Proof. Due to the assumption of strict complementarity the same set of g_i^j will be holding as equalities from a certain iteration number. If S_i^{uu} are not modified, then the criterion function (7.118) is negative definite on the feasible subspace (7.119) - (7.122) (if DP is used, C_i is negative definite on the subspace). Therefore the step taken in the algorithm with $\alpha = 1$ is the same as the Newton step (7.111), cf. Proposition 7.3.2. The Newton iteration converges at a quadratic rate under the assumptions stated. □

The assumption of the stepsize $\alpha = 1$ in Step 2 is necessary due to the application of the absolute value merit function. By extension of the algorithm with the watchdog technique, this socalled Maratos effect may be avoided.

Simple control constraints may be handled by projection ideas, see page 177, cf. e.g. Bertsekas (1982a), Jonson (1983) and Gawande and Dunn (1988).

State-and-control-space Interpolation

Now consider the state-and-control-interpolation, cf. page 202 in Section 7.2. In this, the dynamic equation is treated as a constraint in line with the local constraints. This means that it has an associated multiplier, and that it can not be expected to be fulfilled exactly during the iterations. This approach may be extended to Wilson's application of Newton's method in the variables (x, u, λ, μ, p) along ideas similar to those above and with the same conclusions regarding convergence.

The third approach in Section 7.2, DDP on page 203, is treated in the next section.

7.4 DDP: Local Convergence

We now describe local convergence properties for the DDP algorithm at the end of Section 7.2.

The problem considered is

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{7.123}$$

$$x_{i+1} = f_i(x_i, u_i),\ i = 0, \ldots, N-1 \tag{7.124}$$

$$g_i(x_i, u_i) \leq 0,\ i = 0, \ldots, N-1 \tag{7.125}$$

$$h_i(x_i, u_i) = 0,\ i = 0, \ldots, N-1 \tag{7.126}$$

It is assumed that (7.126) specifies a unique initial point $\underline{x}_0$.

There are three essential phases in the iterations. In the first phase the major objective is to identify that part of the inequality constraints (7.125) that will be binding at the optimal point. This may be accomplished by a combination of the DP algorithm of Section 7.1 for the QLEI problem and the linearization idea of Section 7.2. The selection of positive definite matrices S_i, cf. (7.41), assures that $C_i < 0$ such that DP may be applied at stage i, and also that the direction of change is an improving direction for the merit function. In the second phases the set J^A of active inequalities has been identified (assuming strict complementarity holds). Again DP is combined with the linearization method to assure global convergence, however the iterations are simpler because a QLE - not a QLEI - problem is dealt with. In the third phase modifications of S_i in order to get $C_i < 0$ are not necessary, assuming that second order sufficiency conditions hold, cf. Proposition 2.8.1 and Proposition 4.3.2. For (x^k, u^k) sufficiently close to (x^*, u^*) the stepsize $\alpha = 1$ may under this assumption be applied and the quadratic rate of convergence is achieved.

We here describe the third phase. In the backwards run at iteration k we have at stage i the previous point (x_i^k, u_i^k), the previous multipliers (λ_i^k, μ_i^k) and from stage $(i+1)$ we have Q_{i+1}^{k+1} and P_{i+1}^{k+1}. We now define the criterion (7.6) in the QLE problem at stage i as the sum of two parts. One part is the approximation up to second order of

$$\begin{aligned} & r_i(x_i, u_i) \\ + \quad & (f_i(x_i, u_i) - f_i(x_i^k, u_i^k))' Q_{i+1}^{k+1} (f_i(x_i, u_i) - f_i(x_i^k, u_i^k)) \\ + \quad & P_{i+1}^{k+1} (f_i(x_i, u_i) - f_i(x_i^k, u_i^k)) \end{aligned} \tag{7.127}$$

and the other part consists of the second (but not the first) order terms in the second order approximation to

$$-\lambda_i^k g_i(x_i, u_i) - \mu_i^k h_i(x_i, u_i) \tag{7.128}$$

cf. (7.115) - (7.118). All derivatives are evaluated at the current iteration point (x_i^k, u_i^k). If we formulate the problem in terms of changes $(\delta x, \delta u)$ in relation to (x_i^k, u_i^k) we obtain the problem (top index δ, cf. (7.62) - (7.83), and iteration counter $k+1$ omitted in the sequel)

$$\max_{u_i} [\delta x_i' A_i \delta x_i + \delta x_i' B_i \delta u_i + \delta u_i' C_i \delta u_i + D_i \delta u_i + E_i \delta x_i] \tag{7.129}$$

The local constraints are obtained by linearization of (7.125) - (7.126) around (x_i^k, u_i^k):

$$(\nabla_x g_i(x_i^k, u_i^k) \delta x_i + \nabla_u g_i(x_i^k, u_i^k) \delta u_i + g_i(x_i^k, u_i^k))^j = 0, \quad j \in J_i^A \tag{7.130}$$

$$\nabla_x h_i(x_i^k, u_i^k) \delta x_i + \nabla_u h_i(x_i^k, u_i^k) \delta u_i + h_i(x_i^k, u_i^k) = 0 \tag{7.131}$$

Here the set of active inequality constraints J_i^A is as discussed around (7.7) - (7.9).

The matrices in (7.129) are seen to be (the argument (x_i^k, u_i^k) is omitted):

$$\begin{aligned} A_i &= \tfrac{1}{2}(\nabla_{xx}^2 r_i + 2\nabla_x f_i' Q_{i+1} \nabla_x f_i + \nabla_{xx}^2 (P_{i+1} f_i) \\ &\quad - \nabla_{xx}^2 (\lambda_i^k g_i) - \nabla_{xx}^2 (\mu_i^k h_i)) \end{aligned} \tag{7.132}$$

$$\begin{aligned} B_i &= \nabla_{xu}^2 r_i + 2\nabla_x f_i' Q_{i+1} \nabla_u f_i + \nabla_{xu}^2 (P_{i+1} f_i) \\ &\quad - \nabla_{xu}^2 (\lambda_i^k g_i) - \nabla_{xu}^2 (\mu_i^k h_i) \end{aligned} \tag{7.133}$$

$$C_i = \tfrac{1}{2}(\nabla_{uu}^2 r_i + 2\nabla_u f_i' Q_{i+1} \nabla_u f_i + \nabla_{uu}^2 (P_{i+1} f_i) \tag{7.134}$$

$$
\begin{aligned}
& -\nabla^2_{uu}(\lambda_i^k g_i) - \nabla^2_{uu}(\mu_i^k h_i)) \\
D_i &= \nabla_u r_i + P_{i+1}\nabla_u f_i && (7.135)\\
E_i &= \nabla_x r_i + P_{i+1}\nabla_x f_i && (7.136)
\end{aligned}
$$

Assuming that the gradients with respect to u_i of (7.130) - (7.131) are linearly independent, and assuming that $C_i < 0$ on the subspace defined by (7.130) - (7.131), it is possible to solve the problem (7.129) - (7.131). If not $C_i < 0$ on the subspace, C_i should be modified in order to be so, cf. the discussion on page 205 in Section 7.2. Then K_i, L_i, Q_i and P_i are found as in (7.76) - (7.79). Thus, K_i and L_i are the upper m rows of

$$
\underline{K}_i = \left(\begin{array}{ccc} -2C_i & \nabla_u g_i' & \nabla_u h_i' \\ \nabla_u g_i & 0 & 0 \\ \nabla_u h_i & 0 & 0 \end{array} \right)^{-1} \left(\begin{array}{c} D_i' \\ -g_i \\ -h_i \end{array} \right) \qquad (7.137)
$$

$$
\underline{L}_i = \left(\begin{array}{ccc} -2C_i & \nabla_u g_i' & \nabla_u h_i' \\ \nabla_u g_i & 0 & 0 \\ \nabla_u h_i & 0 & 0 \end{array} \right)^{-1} \left(\begin{array}{c} B_i' \\ -\nabla_x g_i \\ -\nabla_x h_i \end{array} \right) \qquad (7.138)
$$

and

$$
\begin{aligned}
Q_i &= A_i + \tfrac{1}{2}(B_i L_i + L_i' B_i') + L_i' C_i L_i && (7.139)\\
P_i &= K_i' B_i' + 2K_i' C_i L_i + D_i L_i + E_i && (7.140)
\end{aligned}
$$

If there are no local constraints the calculation of K_i, L_i, Q_i and P_i simplifies to

$$
\begin{aligned}
K_i &= -\tfrac{1}{2} C_i^{-1} D_i' && (7.141)\\
L_i &= -\tfrac{1}{2} C_i^{-1} B_i' && (7.142)\\
Q_i &= A_i - \tfrac{1}{4} B_i C_i^{-1} B_i' && (7.143)\\
P_i &= -\tfrac{1}{2} D_i C_i^{-1} B_i' + E_i && (7.144)
\end{aligned}
$$

The following Proposition expresses the second order rate of local convergence.

Proposition 7.4.1 *Assume that all functions are three times continuously differentiable, that at the optimal solution the second order sufficiency conditions hold, that at each stage the gradients* $(\nabla_u g_i, \nabla_u h_i)$ *of the active constraints are linearly independent and that* $C_i < 0$ *for all* i *without modification. Then there is an iteration number* $\overline{k}$ *such that if the steplength* $\alpha = 1$ *is accepted for all* $k > \overline{k}$ *the DDP algorithm at the end of Section 7.2 converges and such that the rate of convergence is quadratic.*

Proof. To establish the proof, we will study the optimization problem at stage i. To this end, let the three times continuously differentiable functions $\rho : R^{n+m} \to R$ and $h : R^{n+m} \to R^\ell$, be given. Define the functions $\xi : R^{n+n+m+m+\ell} \to R$ and $\eta : R^{n+n+m+m} \to R^\ell$ as

$$
\begin{aligned}
\xi(x, \underline{x}, u, \underline{u}, \underline{\mu}) &= \tfrac{1}{2}(x - \underline{x})'(\nabla^2_{\underline{x}\underline{x}}(\rho(\underline{x}, \underline{u}) - \underline{\mu} h(\underline{x}, \underline{u}))(x - \underline{x}) && (7.145)\\
&+ (x - \underline{x})'(\nabla^2_{\underline{x}\underline{u}}(\rho(\underline{x}, \underline{u}) - \underline{\mu} h(\underline{x}, \underline{u}))(u - \underline{u}) \\
&+ \tfrac{1}{2}(u - \underline{u})'(\nabla^2_{\underline{u}\underline{u}}(\rho(\underline{x}, \underline{u}) - \underline{\mu} h(\underline{x}, \underline{u}))(u - \underline{u}) \\
&+ \nabla_{\underline{x}}\rho(\underline{x}, \underline{u})(x - \underline{x}) + \nabla_{\underline{u}}\rho(\underline{x}, \underline{u})(u - \underline{u}) \\
\eta(x, \underline{x}, u, \underline{u}) &= h(\underline{x}, \underline{u}) && (7.146)\\
&+ \nabla_{\underline{x}} h(\underline{x}, \underline{u})(x - \underline{x}) + \nabla_{\underline{u}} h(\underline{x}, \underline{u})(u - \underline{u})
\end{aligned}
$$

where $x \in R^n$, $\underline{x} \in R^n$, $u \in R^m$, $\underline{u} \in R^m$, and $\underline{\mu}' \in R^\ell$ are column vectors. It is seen that ξ and η are three times continuously differentiable. Consider the optimization problem

$$\max_u [\xi(x, \underline{x}, u, \underline{u}, \underline{\mu})] \tag{7.147}$$

$$\eta(x, \underline{x}, u, \underline{u}) = 0 \tag{7.148}$$

where $(x, \underline{x}, \underline{u}, \underline{\mu})$ are considered as parameters, and the optimization variable is $u \in R^m$. We assume that initially the parameter values are given as $(x, \underline{x}, \underline{u}, \underline{\mu}) = (x^o, x^o, u^o, \mu^o)$. For these values we assume that the second order sufficiency conditions hold (cf. (2.69) - (2.73)) and that the gradients $\nabla_u \eta$ in (7.148) are linearly independent. We assume that the optimal solution and corresponding multiplier take the value (u^o, μ^o) at (x^o, x^o, u^o, μ^o). We want to find the first order approximation to the solution (u^*, μ^*) to (7.147) - (7.148) as a function of $(x, \underline{x}, \underline{u}, \underline{\mu})$ near (x^o, x^o, u^o, μ^o).

According to Proposition 2.8.1 the solution (u^*, μ^*) is under the assumptions taken twice continuously differentiable as a function of $(x, \underline{x}, \underline{u}, \underline{\mu})$ near (x^o, x^o, u^o, μ^o) and the first order approximation to the solution is given as

$$\begin{aligned}
&\begin{pmatrix} u^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} u^o \\ \mu^o \end{pmatrix} + \begin{pmatrix} -\nabla^2_{uu}(\xi - \mu^o\eta) & \nabla_u \eta' \\ \nabla_u \eta & 0 \end{pmatrix}^{-1} \\
&\left[\begin{pmatrix} \nabla^2_{xu}(\xi - \mu^o\eta)' \\ -\nabla_x \eta \end{pmatrix} (x - x^o) + \begin{pmatrix} \nabla^2_{\underline{x}u}(\xi - \mu^o\eta)' \\ -\nabla_{\underline{x}} \eta \end{pmatrix} (\underline{x} - x^o) \right. \\
&\left. + \begin{pmatrix} \nabla^2_{\underline{u}u}(\xi - \mu^o\eta)' \\ -\nabla_{\underline{u}} \eta \end{pmatrix} (\underline{u} - u^o) + \begin{pmatrix} \nabla^2_{\underline{\mu}u}(\xi - \mu^o\eta)' \\ -\nabla_{\underline{\mu}} \eta \end{pmatrix} (\underline{\mu} - \mu^o) \right]
\end{aligned} \tag{7.149}$$

Here all derivatives are evaluated at $(x, \underline{x}, u, \underline{u}, \underline{\mu}) = (x^o, x^o, u^o, u^o, \mu^o)$. We find

$$\begin{aligned}
&\nabla_u \eta = \nabla_{\underline{u}} h \\
&\nabla^2_{uu} \eta = 0 \\
&\nabla_x \eta = \nabla_{\underline{x}} h \\
&\nabla^2_{xu} \eta = 0 \\
&\nabla_{\underline{x}} \eta = \nabla_{\underline{x}} h - \nabla_{\underline{x}} h + \nabla^2_{\underline{x}\underline{x}} h(x - \underline{x}) + \nabla^2_{\underline{x}\underline{u}} h(u - \underline{u}) = 0 \\
&\nabla^2_{\underline{x}u} \eta = \nabla^2_{\underline{x}\underline{u}} h \\
&\nabla_{\underline{\mu}} \eta = 0 \\
&\nabla^2_{\underline{\mu}u} \eta = 0 \\
&\nabla_{\underline{u}} \eta = \nabla_{\underline{u}} h + \nabla^2_{\underline{x}\underline{u}} h(x - \underline{x}) + \nabla^2_{\underline{u}\underline{u}} h(u - \underline{u}) - \nabla_{\underline{u}} h = 0 \\
&\nabla^2_{\underline{u}u} \eta = \nabla^2_{\underline{u}\underline{u}} h \\
&\nabla^2_{uu} \xi = \nabla^2_{\underline{u}\underline{u}} \rho - \nabla^2_{\underline{u}\underline{u}} (\mu^o h) \\
&\nabla^2_{xu} \xi = \nabla^2_{\underline{x}\underline{u}} \rho - \nabla^2_{\underline{x}\underline{u}} (\mu^o h) \\
&\nabla^2_{\underline{x}u} \xi = -\nabla^2_{\underline{x}\underline{u}} \rho + \nabla^2_{\underline{x}\underline{u}} (\mu^o h) + \nabla^2_{\underline{x}\underline{u}} \rho = \nabla^2_{\underline{x}\underline{u}} (\mu^o h) \\
&\nabla^2_{\underline{u}u} \xi = -\nabla^2_{\underline{u}\underline{u}} \rho + \nabla^2_{\underline{u}\underline{u}} (\mu^o h) + \nabla^2_{\underline{u}\underline{u}} \rho = \nabla^2_{\underline{u}\underline{u}} (\mu^o h) \\
&\nabla^2_{\underline{\mu}u} \xi = 0
\end{aligned}$$

With this we find that (7.149) reduces to

$$\begin{pmatrix} u^* \\ \mu^* \end{pmatrix} = \begin{pmatrix} u^o \\ \mu^o \end{pmatrix} - \begin{pmatrix} -\nabla^2_{uu}(\xi - \mu^o\eta) & \nabla_u\eta' \\ \nabla_u\eta & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} \nabla^2_{xu}(\xi - \mu^o\eta)' \\ -\nabla_x\eta \end{pmatrix} (x - x^o) \right] \tag{7.150}$$

As seen, to the first order the solution is independent of $(\underline{x}, \underline{u}, \underline{\mu})$, and by comparing the feedback solutions in (7.137) - (7.138) and (7.150) it is seen that they are identical. We summarize the conclusion as the following partial result :

Assume that $\rho : R^{n+m} \to R$ and $h : R^{n+m} \to R^\ell$ are three times continuously differentiable and that at the parameter value $(x, \underline{x}, \underline{u}, \underline{\mu}) = (x^o, x^o, u^o, \mu^o)$ the second order sufficiency conditions are fulfilled at the optimal solution to the problem (7.147) - (7.148), and that the gradients $\nabla_u\eta$ are linearly independent here; assume that the optimal solution and corresponding multiplier, (u^*, μ^*) to (7.147) - (7.148) for the parameter value (x^o, x^o, u^o, μ^o) takes the value (u^o, μ^o); then (u^*, μ^*) is a twice continuously differentiable function of $(x, \underline{x}, \underline{u}, \underline{\mu})$ near (x^o, x^o, u^o, μ^o) and the first order approximation is as given in (7.150).

It is obvious that we may reach the same conclusion if we substitute (7.148) by the constraint $h(x, u) = 0$.

Now we interpret the result in relation to the OCP. The problem (7.147) - (7.148) is of the same form as (7.129) - (7.136), and $(x, \underline{x}, \underline{u}, \underline{\mu})$, u, ξ and η correspond to $(x_i, x_i^k, u_i^k, \mu_i^k)$, to u_i, to the criterion in (7.129) and to the constraint functions in (7.130) - (7.131), respectively. The functions ρ and h in (7.145) - (7.146) correspond to (7.127) + (7.128) and (7.125) - (7.126), respectively (only active g_i^j are considered).

Using $h(x, u) = 0$ rather than (7.148) corresponds to keeping the local constraints fulfilled, cf. the discussion in relation to (7.87) - (7.90).

Consider the backwards DDP recursion. At $i = N - 1$ we see from the above partial result that to the first order the solution will be independent of (x_i^k, u_i^k, μ_i^k). Therefore also the matrices Q_i and P_i will be independent of (x_i^k, u_i^k, μ_i^k), cf. (7.139) - (7.140). By induction it follows that at all stages i the feedback form matrices (7.137) - (7.138) and the matrices Q_i and P_i will be independent of (x_i^k, u_i^k, μ_i^k). In the forwards recursion we start at $x_0^{k+1} = x_0^k = \underline{x}_0$ and we will in the forwards recursion therefore find $(u_i^{k+1}, \mu_i^{k+1}) = (u_i^k, \mu_i^k)$ to the first order approximation for all i. Consequently, also $x_i^{k+1} = x_i^k$ to the first order approximation.

Considering the iteration as a mapping $(u^k, \mu^k) \to (u^{k+1}, \mu^{k+1})$ we see that this mapping is twice continuously differentiable in a neighborhood of (u^*, μ^*), that at (u^*, μ^*) its gradient is vanishing, and that (u^*, μ^*) is a fixed point. Similarly holds for the mapping $(x^k, u^k) \to (x^{k+1}, u^{k+1})$. Using Ortega and Rheinboldt (1970) p. 304 it is seen that this implies convergence at a quadratic rate. □.

The convergence and rate of convergence result is the same as for the Newton iteration, Proposition 7.3.3, and the same comments as there applies to the stepsize condition $\alpha = 1$.

This concludes the analysis of the DDP algorithm.

7.5 Conclusions

The present chapter has dealt with algorithms that are based on recursive formulation and solution of QLEI problems. Three aspects are treated, related to handling of inequality constraints

in the QLEI problems, global convergence of the original non-QLEI problem, and rate of local convergence.

The solution of the QLEI was treated in Section 7.1 where two approaches were presented. One was to solve the QLEI as a sequence of QLE's using DP and an active set approach; thus, the results of Section 4.3 on the solution of the QLE may be applied. Further is was indicated how the generalized maximum principle may be applied, where the upper boundary approximation is implicitly given through a quadratic function and a set of linear inequality constraints, cf. Section 6.4.

Global convergence was treated in Section 7.2. There is was shown that the absolute value penalty function may be applied to this end.

Another important aspect dealt with in this section is that the OCP may be treated in at least three ways. One way is to eliminate all the state variables, and express the problem exclusively in terms of control variables. This way the dynamic equation is always fulfilled exactly. The solution of the QLE is not necessarily to be done by DP.

As an extension of this the DDP approach always keeps the dynamic equation fulfilled, however, the state variables are kept as essential elements since DP is used for solution of the QLE problems. This is important, since the sequences of updating of the various components at stage i (u_i, μ_i and p_{i+1}) depend on this. Another important aspect of the DDP approach is that local constraints may often be exactly fulfilled, such that a feasible solution is always available during the iterations.

In a third variant the specific structure of the OCP is neglected, and the control and state variables are treated equally. Therefore the dynamic equation need not be fulfilled during iterations in this approach.

The rate of local convergence is treated in Section 7.3 and in Section 7.4. In Section 7.3 it is shown that the first and third of the approaches identified in Section 7.2 may yield quadratic rate of convergence. In fact, for the locally unconstrained OCP, the first approach yields results that are identical to the Newton iterations applied to the problems where the state variables have been eliminated. Both approaches may also be interpreted as applications of Wilson's method, i.e., Newton iterations relative to the set of KKT conditions.

In Section 7.4 it is shown that also the DDP approach may yield the quadratic rate of convergence, and also if local nonlinear constraints are present.

Chapter 8

Price Decomposition

Dual algorithms play an important role in mathematical programming. Theoretically, they may be seen as the counterpart of primal decomposition algorithms, cf. Chapter 5. In most applications they may be interpreted as Lagrangian relaxation algorithms where the iterations take place in order to find the optimal values of the dual variables. For many problems this provides efficient solution procedures, in particular when the problem has an additively separable structure. It also provides for economic and organizational interpretations, as illustrated in Section 1.6; this is essentially due to the interpretation of the Lagrange multipliers as prices. In the case of nonconvex problems Lagrangian relaxation fails as direct solution procedure, however it may be applied further according to e.g. branch-and-bound strategies, or in heuristics.

Lagrangian relaxation has been applied to the dynamic equation, Tamura (1975). This is (as always) particularly convenient if the resulting relaxed problem is strictly concave, implying a unique solution. Moreover, in this case the dual function is smooth, and convergent algorithm for finding the optimal values of the dual variables p_i may be relatively simple.

On the other hand, if the relaxed problem is concave, but not strictly so, the solution to the relaxed problems need not be unique. In this case subgradient algorithms may be applied. However, they suffer from a slow rate of convergence. An alternative is to make the solution unique by addition of strictly concave terms; this in turn implies the need for an updating of these terms. In general this approach destroys decomposability. The linear problems are special in this respect, cf. Proposition 3.4.8.

Also in case of non-concave relaxed problems the relaxed criterion may be made strictly concave by addition of strictly concave terms. The application of augmented Lagrangians in mathematical programming is the obvious examplification of this, Section 1.4. It has the disadvantage that it destroys any separability which may be present (and which obviously is present for optimal control problems). Consequently, separability may be reintroduced by linearization as in Stephanopoulos and Westerberg (1975), also undertaken in Tatjewski (1985), or by transformation as in Watanabe, Nishimura and Matsubara (1978). Bertsekas (1979) preserves decomposability by adding additively separable terms, resulting in a three level procedure, simplified to two level procedures in Tanikawa and Mukai (1985) and Feng, Mukai and Brown (1990). All of these approaches will for the nonconcave case provide solutions that are not necessarily globally optimal.

In contrast to this, the relaxation with nonlinear π_i will provide sufficient conditions for optimality, Proposition 3.5.2.

The optimality conditions of Lagrangian relaxation are necessary conditions only for convex

problems and under assumptions of constraint qualifications fulfilled. As shown in Proposition 3.5.2 it is always possible to apply non-linear price functions π_i and also in this case economic and organizational interpretations apply.

For linear as well as nonlinear price functions the search for those values that provide for an optimal solution to the relaxed problems which is at the same time optimal in the OCP may be guided by duality theory, see Section 3.6.

In this chapter we describe algorithms based on duality theory for both linear and nonlinear price functions. Thus, in Section 8.1 we show how for special choices of nonlinear π_i it is possible to obtain that the dual function is a convex function of parameter values. This in turn makes it possible to derive convergent algorithms.

Section 8.2 deals with Lagrangian relaxation. The consequences of choosing various relaxations are discussed, and for each relaxation convergence results are derived. Under suitable assumptions, in particular that the relaxed solutions are unique and that the functions involved are sufficiently smooth, Newton iterations are derived. Section 8.3 treats the case of linearity in the state variables, again with Lagrangian relaxation. This situation is difficult, as now the relaxed solutions need not be unique. Again, various relaxations are discussed, and convergent algorithms are described.

8.1 Iterations With Nonlinear Supports

In this section we consider iterative procedures using non-linear supports. We give general algorithms for discrete problems, based on subgradient or descent ideas. This may be seen as an application of the results on relaxation with nonlinear price functions, Section 3.5, and duality, Section 3.6.

The problem considered is the following OCP:

$$\max[\sum_{i=0}^{N-1} r_i(x_i,u_i) + r_N(x_N)] \tag{8.1}$$

$$x_{i+1} = f_i(x_i,u_i), \tag{8.2}$$

$$(x_i,u_i) \in V_i \tag{8.3}$$

$$x_N \in V_N \tag{8.4}$$

We relax this problem as in Section 3.5 and get the relaxed problem:

$$\max_{(x_0,u_0)} [r_0(x_0,u_0) + \pi_1(f_0(x_0,u_0))] \tag{8.5}$$

$$(x_0,u_0) \in V_0 \tag{8.6}$$

$$\max_{(x_i,u_i)} [r_i(x_i,u_i) - \pi_i(x_i) + \pi_{i+1}(f_i(x_i,u_i))] \tag{8.7}$$

$$(x_i,u_i) \in V_i, i = 1,\ldots,N-1 \tag{8.8}$$

$$\max_{x_N}[r_N(x_N) - \pi_N(x_N)] \tag{8.9}$$

$$x_N \in V_N \tag{8.10}$$

We showed in Section 3.5 that for any problem there exist supports π_i such that we can solve the OCP by solving the sequence of smaller problems (8.5) - (8.10). Moreover, these supports may be simpler than the dynamic programming upper boundaries, as witnessed by Proposition 3.5.4. The supports must satisfy (3.34) - (3.37), and this leaves some freedom of choice.

For a problem where Proposition 3.5.6 applies it is well known that the linear supports may be successfully applied and that their slopes can be found by applying e.g. gradient or subgradient methods for solving the dual problem. See e.g. Shor (1985) and Kiwiel (1985) for recent monographs on nonsmooth optimization.

Subgradient Algorithms

The first question in relation to application of nonlinear support is how to select the nonlinear functions. In the light of Proposition 3.5.4., it seems reasonable to assume that in most cases it will be expedient to use the structure of the problem in order to suggest the forms of the π to be used. In other words, π should be problem specific. This will be exemplified later in the section. We start out with more general notions.

Therefore let us define π_i as the weighted sum of some other given functions π_i^j , i.e.

$$\pi_i(x_i) = \sum_{j=1}^{k} w_i^j \pi_i^j(x_i) \tag{8.11}$$

Here the scalar w_i^j defines the weight of the function π_i^j. Let $w_i = (w_i^1, w_i^2, \ldots, w_i^k)'$ and $w = (w_0', w_1', \ldots, w_N')'$. We may want to restrict w to a set $W \in R^{(N+1)k}$. We can express the dual function $D(\pi)$ as $D(w)$. We have then as follows:

Proposition 8.1.1 *Let π_i be fined as above for $i = 0, \ldots, N$, and assume that W is convex. Assume that the supremum in (8.5) - (8.10) is attained and finite (i.e. the finite maximum exists) for all $w \in W$. Then D is convex on W, and a subgradient s, $s = (s_0, s_1, \ldots, s_N)'$, $s_i = (s_i^{1\prime}, s_i^{2\prime}, \ldots, s_i^{k\prime})$ has components*

$$s_i^j = (\pi_i^j(f_{i-1}(x_{i-1}^\diamond, u_{i-1}^\diamond)) - \pi_i(x_i^\diamond))$$

where $x_{i-1}^\diamond$, $u_{i-1}^\diamond$ and $x_i^\diamond$ are maximizing in (8.5) - (8.10).

Proof. Let $w \in W$ and $w^\diamond \in W$ be arbitrary, and let (x, u) and $(x^\diamond, u^\diamond)$ be maximizing in (8.5) - (8.10) for w and $w^\diamond$, respectively. We then have

$$\begin{aligned}
D(w) &= \sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N) \\
&+ \sum_{j=1}^{k} \Big(\sum_{i=0}^{N-1} -w_i^j \pi_i^j(x_i) + w_{i+1}^j \pi_{i+1}^j(f_i(x_i, u_i)) - w_N^j \pi_N^j(x_N) + w_0^j \pi_0^j(x_0) \Big) \\
&\geq \sum_{i=0}^{N-1} r_i(x_i^\diamond, u_i^\diamond) + r_N(x_N^\diamond) \\
&+ \sum_{j=1}^{k} \Big(\sum_{i=0}^{N-1} -w_i^j \pi_i^j(x_i^\diamond) + w_{i+1}^j \pi_{i+1}^j(f_i(x_i^\diamond, u_i^\diamond)) - w_N^j \pi_N^j(x_N^\diamond) + w_0^j \pi_0^j(x_0^\diamond) \Big) \\
&= \sum_{i=0}^{N-1} r_i(x_i^\diamond, u_i^\diamond) + r_N(x_N^\diamond)
\end{aligned}$$

$$
\begin{aligned}
&+\sum_{j=1}^{k}(\sum_{i=0}^{N-1} -w_i^j\pi_i^j(x_i^\diamond) + w_{i+1}^j\pi_{i+1}^j(f_i(x_i^\diamond,u_i^\diamond)) - w_N^j\pi_N^j(x_N^\diamond) + w_0^j\pi_0^j(x_0^\diamond))\\
&+\sum_{j=1}^{k}(\sum_{i=0}^{N-1} -w_i^{j\diamond}\pi_i^j(x_i^\diamond) + w_{i+1}^{j\diamond}\pi_{i+1}^j(f_i(x_i^\diamond,u_i^\diamond)) - w_N^{j\diamond}\pi_N^j(x_N^\diamond) + w_0^{j\diamond}\pi_0^j(x_0^\diamond))\\
&-\sum_{j=1}^{k}(\sum_{i=0}^{N-1} -w_i^{j\diamond}\pi_i^j(x_i^\diamond) + w_{i+1}^\diamond\pi_{i+1}^j(f_i(x_i^\diamond,u_i^\diamond)) - w_N^{j\diamond}\pi_N^j(x_N^\diamond) + w_0^{j\diamond}\pi_0^j(x_0^\diamond))\\
&= D(w^\diamond)\\
&+\sum_{j=1}^{k}(\sum_{i=1}^{N}(w_i^j - w_i^{j\diamond})(\pi_i^j(f_{i-1}(x_{i-1}^\diamond,u_{i-1}^\diamond)) - \pi_i(x_i^\diamond)))
\end{aligned}
$$

From this it is seen that at the arbitrary point $w^\diamond$ D has a subgradient, implying that D is convex. It is also seen that the subgradient has components as stated. □

For the most general selection of π_i^j we assume that V_i contains only finitely many elements. Then we can assign a particular value of π_i^j to each point where π_i need be defined. Clearly by such specification it will be possible to select π_i such that the dynamics will be fulfilled; for instance, π_i may be selected equal to UB_i or RUB_i, cf. Proposition 3.5.3.

Formally we may proceed as follows. Let the elements for which we need define π_i be numbered 1 through t_i , *i*.e., the elements are $x_i^1, x_i^2, \ldots, x_i^{t_i}$. We then define π_i as the linear combination

$$\pi_i(x_i) = \sum_{j=1}^{t_i} w_i^j \delta_i^j(x_i) \tag{8.12}$$

where δ_i^j is defined as

$$\delta_i^j(x_i) = \begin{cases} 1 & \text{if } x_i = x_i^j \\ 0 & \text{otherwise} \end{cases} \tag{8.13}$$

and w_i^j is a scalar which indicates the "weight" of δ_i^j. All w_i^j are unrestricted, except for those indexes (N,j) for which $x_N^j \in X_N$ (for these indexes we require $w_N^j = 0$); and for the indexes w_0^j, which are all zero. If we then define $w_i = (w_i^1, \ldots, w_i^{t_i})'$ and $w = (w_0', \ldots, w_N')'$ we can conceive π_i as parameterized by w_i as before.

It follows from the above Proposition that D is convex as a function of w, and that the (i,j)-th component of a subgradient at the point w^o is given as

$$\delta_i^j(f_{i-1}(x_{i-1}^o, u_{i-1}^o)) - \delta_i^j(x_i^o) \tag{8.14}$$

where (x^o, u^o) are any maximizing values in (8.5) - (8.10) for $w = w^o$. Any component of the subgradient is seen to take one of the values -1, 0 or 1. In this context the following scheme is intuitively clear for selection of π_i^{k+1} at iteration k, given π_i^k and a steplength α:

$$
\begin{aligned}
&\text{If } f_{i-1}(x_{i-1}^o, u_{i-1}^o) = x_i^o \qquad \text{then let } \pi_i^{k+1} = \pi_i^k \text{ for all } x_i \\
&\text{If } f_{i-1}(x_{i-1}^o, u_{i-1}^o) \neq x_i^o \qquad \text{then decrease } \pi_i^{k+1}(f_{i-1}(x_{i-1}^o, u_{i-1}^o)) \text{ by } \alpha \\
&\qquad\qquad\qquad\qquad\quad \text{and increase } \pi_i^{k+1}(x_i^o) \text{ by } \alpha \\
&\qquad\qquad\qquad\qquad\quad \text{and let } \pi_i^{k+1} = \pi_i^k \text{ for all other } x_i
\end{aligned}
\tag{8.15}
$$

We see that with this we can apply any method of subgradient optimization. We can then by minimization of D find the lowest upper bound on the optimal criterion value in (8.1) - (8.4). If we find the minimizing w, we also have the solution to the problem (although not necessarily uniquely specified).

Example 8.1.1 *The following example was given by Pedersen and Ravn in 1987 in the context of optimal scheduling of combined heat and power systems.*

We have an OCP starting at $\underline{x}_1$ and ending at $\underline{x}_3$. The control space for u_1 consists of three points: $U_1 = \{u_1^{10}, u_1^{01}, u_1^{11}\}$. The criterion and dynamic equation at stage 1 are

$$r_1(\underline{x}_1, u_1) = \begin{cases} -6000 & \text{if } u_1 = u_1^{10} \\ 0 & \text{if } u_1 = u_1^{01} \\ -2600 & \text{if } u_1 = u_1^{11} \end{cases}$$

$$f_1(\underline{x}_1, u_1) = \begin{cases} x_2^{10} & \text{if } u_1 = u_1^{10} \\ x_2^{01} & \text{if } u_1 = u_1^{01} \\ x_2^{11} & \text{if } u_1 = u_1^{11} \end{cases}$$

The space for states x_2 contain three points: $X_2 = \{x_2^{01}, x_2^{11}, x_2^{10}\}$. The space for controls u_2 contain three points: $U_1 = \{u_1^{01}, u_1^{11}, u_1^{10}\}$ and $V_2 = \{(x_2^{01}, u_2^{01}), (x_2^{11}, u_2^{11}), (x_2^{11}, u_2^{11})\}$. The criterion and dynamics at stage 2 are

$$r_2(x_2, u_2) = \begin{cases} 0 & \text{if } (x_2, u_1) = (x_2^{10}, u_1^{10}) \\ -2400 & \text{if } (x_2, u_1) = (x_2^{01}, u_1^{01}) \\ -7000 & \text{if } (x_2, u_1) = (x_2^{11}, u_1^{11}) \end{cases}$$

$$f_2(\underline{x}_1, u_1) = \underline{x}_3 \ \forall \ (x_2, u_2) \in V_2$$

As seen the only difficulty in this example is to get the dynamic equation fulfilled at stage 1.

Since x_1 and x_3 are given we choose $\pi_1(x_1) \equiv 0$ and $\pi_3(x_3) \equiv 0$ and there is no need to change them. Let us initially select $\pi_2(x_2^{10}) = 6000$, $\pi_2(x_2^{01}) = -6000$ and $\pi_2(x_2^{11}) = 2000$.

We find the values of $r_i - \pi_i + \pi_{i+1}$, $i = 1, 2$, as in the following table:

Stage 1			Stage 2		
$(\underline{x}_1, u_1^{10})$	$(\underline{x}_1, u_1^{01})$	$(\underline{x}_1, u_1^{11})$	(x_2^{10}, u_2^{10})	(x_2^{01}, u_2^{01})	(x_2^{11}, u_2^{11})
0*	−6000	−600	−6000	−1000*	4400

Maximizing with respect to (x_1, u_1) at stage 1 we see that the values are 0, -6000 and -600, so that the maximal value is 0, attained by $u_1 = u_1^{10}$. At stage 2 the values are -6000, -1000 and -4400, such that the maximal value is -1000, attained at $(x_2^*, u_2^*) = (x_2^{01}, u_2^{01})$. Since $f_1(x_1^*, u_1^*) = x_2^{10} \neq x_2^* = x_2^{01}$ the dynamic equation is not fulfilled, and π_2 must be changed. According to (8.15) we increase $\pi_2(x_2^{01})$ and decrease $\pi_2(x_2^{10})$. The following table indicates the values of π_2 used throughout the iteration; iteration 0 corresponds to the initial value used above.

Iteration no.	$\pi_2(x_2^{10})$	$\pi_2(x_2^{01})$	$\pi_2(x_2^{11})$
0	6000	−6000	2000
1	5000	−5000	2000
2	5000	−4000	1000
3	4000	−3000	1000

As seen, π_2 has been changed by the arbitrary steplength 1000. Continuing this way we find the subsequent iterations with the values of $r_i - \pi_i + \pi_{i+1}$, $i = 1, 2$, as in the following table:

	Stage 1			Stage 2		
Iteration no.	$(\underline{x}_1, u_1^{10})$	$(\underline{x}_1, u_1^{01})$	$(\underline{x}_1, u_1^{11})$	(x_2^{10}, u_2^{10})	(x_2^{01}, u_2^{01})	(x_2^{11}, u_2^{11})
0	0*	−6000	−600	−6000	−1000*	−4400
1	−1000	−5000	−600*	−5000	−2000*	−4400
2	−1000*	−4000	−1600	−5000	−3000*	−3400
3	−2000	−3000	−1600*	−4000	−4000	−3400*

It is seen that the sum of the optimal values of $r_i - \pi_i + \pi_{i+1}$ is decreasing through the iterations (the values are -1000, -2600, -4000 and -5000). It is further seen that the direction of change of π_2 at iteration k has been chosen according to (8.15).

We see that at the last iteration we get the dynamic equation fulfilled, as $f_1(x_1^*, u_1^*) = x_2^{11} = x_2^*$ and the problem is solved by $(x_1^*, u_1^*) = (\underline{x}_1, u_1^{11})$, $(x_2^*, u_2^*) = (x_2^{11}, u_2^{11})$.

The function π_2 may in the context of combined heat and power scheduling be interpreted as follows. There are two production units. They produce the amounts u_1 and u_2. The units must between them produce a certain amount of heat. They do so, if they produce such that the dynamic equation is fulfilled (i.e., the dynamic equation represents a sum constraint). Each of the units has production costs given by $-r_i$. The units 'see' a 'price' π_2 such that unit 1 has total costs $-r_1(u_1) - \pi_2(u_1)$ and unit 2 has total costs $-r_2(u_1) + \pi_2(u_2)$. Each unit produces such as to minimize its own costs, including the 'income' relative to the 'price'. A 'coordinator' tries to select a price function π_2 such that the units between them produce the desired amount, i.e., such that the dynamic equation is fulfilled. The coordinator may be seen to have a certain amount of money which is allocated between the units, according to how they produce. However, the coordinator is indifferent as to which productions the units choose, if only they fulfill the dynamic equation, because the total amount paid to unit 1 and to unit 2 is zero.

We may also interpret the price function π_2 as follows. Assume that production unit 1 is responsible for the fulfillment of the dynamic equation, i.e. that total production balances demand. Unit 1 may produce all itself, or pay unit 2 for some of the production; payment is given according to the price function π_2. If a payment can be found such that the dynamic equation is fulfilled by the optimal solutions of the two individual units, then total payment cancels. Therefore, as both units attempt to minimize costs, including payments π_2, such decentral optimal solution will also be optimal to the whole problem. Other applications of nonlinear π are given in Hansen (1987) and Koppelhus (1991). □

Finally we shall interpret in the terminology established here the usual Lagrangian relaxation method based on linear support. Assume that V_i contains only finitely many elements. To each component x_i^j of x_i we define a scalar $w_i^j(x_i^j)$. Single out one element x_i; for simplicity, let it be the zero element, $(0) = (0, \ldots, 0)'$. Now define W by the following restrictions on w_i and w_i^j :

$$w_i(x_i) = \sum_{j=1}^{n} w_i^j(x_i^j) \tag{8.16}$$

$$w_i^j(x_i^j) = x_i^j w_i^j(0) \tag{8.17}$$

We have thus restricted w_i to be linear in x_i, and in fact at each stage i there is only freedom to choose the n-dimensional vector $w_i(0)$. Since the restrictions are linear and $D(w)$ is convex, the minimization of $D(w)$ subject to the constraints is a convex optimization problem.

We thus see that we obtain the usual Lagrangian relaxation with $\pi_i = w_i(0)$.

Nonlinear Supports: A Descent Algorithm

A weakness of subgradient algorithm is that during the iterations the value of the dual function is not necessarily decreasing, even for arbitrarily small step lengths. In other words, the direction specified by the subgradient is not necessarily a descent direction for the dual function.

If more care is taken in the selection of the direction by evaluating the subdifferential rather than only finding one element in it, then improvement may be attained at each iteration. We shall now specify a descent algorithm as follows. We disregard π_0, which is identically zero.

Algorithm

Step 0 Let iteration counter $k = 0$. Select $\pi_i^k(x_i)$, $i = 1, \ldots, N-1$ arbitrarily. Let $\pi_N^{k+1}(x_N) = r_N(x_N)$, $x_N \in X_N$, and $\pi_N^{k+1}(x_N) = \pi_N^k(x_N) - a$ otherwise, where a is a sufficiently large positive constant. Let $\pi_0(x_0) \equiv 0$.

Step 1 Find all maximizing (x_0, u_0) in (8.5) - (8.6) and call this set $\{x_0^k, u_0^k\}$.

Step 2 For $i = 1, \ldots, N-1$ do:

- Find all maximizing (x_i, u_i) in (8.7) - (8.8) and call this set $\{x_i^k, u_i^k\}^\diamond$.
- Let $\{x_i^k, u_i^k\}$ be the subset of $\{x_i^k, u_i^k\}^\diamond$ for which $x_i^k = f_{i-1}(x_{i-1}, u_{i-1})$ for some $(x_{i-1}, u_{i-1}) \in \{x_{i-1}^k, u_{i-1}^k\}$.

Step 3 If there is a $(x_{N-1}, u_{N-1}) \in \{x_{N-1}^k, u_{N-1}^k\}$ such that $f_{N-1}(x_{N-1}, u_{N-1}) \in X_N$, then the sequence $\{x_i^k, u_i^k\}$, $i = 0, \ldots, N-1$ is optimal: Stop.

Step 4 Calculate an upper bound according to Proposition 3.5.1 if desired.

Step 5 For $i = N-1, \ldots, 0$ do:

Step 5A If $\{x_i^k, u_i^k\}$ from Step 2 is not empty then let $\pi_i^{k+1}(x_i) = \pi_i^k(x_i)$; take next i.

Step 5B If $\{x_i^k, u_i^k\}$ from Step 2 is empty then let $\pi_i^{k+1}(x_i) = \pi_i^k(x_i) + d1(x_i)$ for all those x_i for which there is a u_i such that $(x_i, u_i) \in \{x_i^k, u_i^k\}^\diamond$ ($d1(x_i)$ is defined in (8.18) below), and let $\pi_i^{k+1}(x_i) = \pi_i^k(x_i)$ for all other x_i. Skip all remaining i, let $k = k+1$ and go to Step 1.

The number $d1(x_i)$ used in Step 5B is defined as follows:

$$d1(x_i) = r_{i-1}(x_{i-1}, u_{i-1}) - \pi_{i-1}^k(x_{i-1}) + \pi_{i-1}^k(f_{i-1}(x_{i-1}, u_{i-1})) - (F_i(x_i) + \pi_i^k(x_i)) \tag{8.18}$$

for any $(x_{i-1}, u_{i-1}) \in \{x_{i-1}^k, u_{i-1}^k\}^\diamond$ and any x_i for which $(x_i, u_i) \in \{x_i, u_i\}^\diamond$. The function F_i is defined in (3.32). We see that $d1(x_i)$ is the difference in criterion value in the solution to (8.7) - (8.8) according to whether maximization is constrained only by (8.8) (the first three terms), or constrained by (8.8) plus the requirement that x_i should be attained by $f_i(x_i, u_i)$ (the last two terms). If Step 5B is used when defining π_i^{k+1} then in the next forward run we will find that $\{x_{i-1}^{k+1}, u_{i-1}^{k-1}\}^\diamond$ will contain elements such that $x_i^k = f_{i-1}(x_{i-1}^{k+1}, u_{i-1}^{k-1})$. In other words, defining $d1(x_i)$ this way is an attempt to have the maximizing (x_i, u_i) in (8.5) - (8.10) satisfying the dynamic equation.

Proposition 8.1.2 *Suppose that all V_i contain only finitely many elements, that a solution to OCP exists and that the constant a has been chosen sufficiently large. Then the algorithm will find a solution in a finite number of steps.*

Proof. Since all V_i contain finitely many elements it will be possible to choose a such that $f_{N-1}(x_{i-1}, u_{i-1}) \in V_N$ for any maximizing (x_{i-1}, u_{i-1}). If there is a $(x_{N-1}, u_{N-1}) \in \{x_{N-1}^k, u_{N-1}^k\}$ such that $f_{N-1}(x_{N-1}, u_{N-1}) \in X_N$, then the sequence $\{x_i^k, u_i^k\}$, $i = 0, \ldots, N-1$ feasible and by

Proposition 3.5.2 optimal. In this case the algorithm stops (Step 3). If the dynamic equation is not fulfilled for all i by the maximizing values then at least once in each backwards run Step 5B is applied. Every time Step 5B is applied it will imply that the upper bound is decreased: the optimal value in (8.7) - (8.8) for $i-1$ is not affected by the change in π_i from π_i^k to π_i^{k+1}, and the optimal value in (8.7) - (8.8) for i is decreased by a positive amount. Since there are only finitely many elements in V_i we see that π_i can only take finitely many values during the iterations for any x_i. Since there is a lower bound (viz., the optimal criterion value) on the upper bound then eventually after a finite number of iterations the decrease will stop, implying that the dynamic equation is fulfilled for all i. □

We see that the idea in the algorithm is to try to make $f_{i-1}(x_{i-1}^{k+1}, u_{i-1}^{k+1})$ equal to x_i^k by increasing $\pi_i(x_i^k)$ by the amount $d1(x_i^k)$. Alternatively we may try to make x_i^{k+1} equal to $f_{i-1}(x_{i-1}^k, u_{i-1}^k)$ by reducing $\pi_i(f_{i-1}(x_{i-1}^k, u_{i-1}^k))$ by an amount $d2$. The determination of $d1$ in Step 5B, as specified above, requires the solution of a restricted problem, as seen in the definition of F_i in (3.32). Similarly, if we want to determine $d2$ we must find it as

$$d2(x_i) = r_i(x_i^k, u_i^k) - \pi_i^k(x_i^k) + \pi_k^{i+1}(f_i(x_i^k, u_i^k)) - (RF_i(x_i) - \pi_i^k(x_i)) \tag{8.19}$$

for any $x_i^k, u_i^k) \in \{x_i, u_i\}^\circ$ and any x_i for which $x_i = f_{i-1}(x_{i-1}, u_{i-1})$ for some $(x_{i-1}, u_{i-1}) \in \{x_{i-1}, u_{i-1}\}^\circ$. Again, to determine RF_i, defined in 3.33 we need to solve a problem but this will generally be simpler than the problem determining $d1$ since we when determining $d2$ have to maximize only with respect to u_i, because x_i is fixed. The modified algorithm will then be identical to the one above, expect for Step 5B, which should be:

Step 5B' If $\{x_i^k, u_i^k\}$ from Step 2 is empty then let $\pi_i^{k+1}(x_i) = \pi_i^k(x_i) - d2(x_i)$ for all those x_i for which there is a $(x_{i-1}, u_{i-1}) \in \{x_{i-1}^k, u_{i-1}^k\}$ such that $x_i = f_{i-1}(x_{i-1}, u_{i-1})$; and let $\pi_i^{k+1}(x_i) = \pi_i^k(x_i)$ for all other x_i.

A second modification is to apply both Step 5B and Step 5B'. Under the assumptions of Proposition 8.1.2 any one of the two modified algorithms will find a solution in a finite number of steps.

The algorithms above require the solution of optimization problems to determine the values of $d1$ and $d2$. The advantage of this is that we in this way evaluate the whole subdifferential, which in turn permits a decrease of the dual function in each iteration.

A weakness is then that the maximizing (x_i, u_i) will tend to become non-unique, implying a need to store an increasing number of solutions to (8.5) - (8.10). This is a consequence of using the values $d1$ or $d2$ which are determined such as "just exactly" to change π without making previous solutions become irrelevant. The algorithms can be interpreted as trying to construct good approximations to UB_i and RUB_i, respectively. For this, the functions F_i and RF_i are used. From Propositions 3.2.2 and 3.2.3 we see that with π equal to either of the upper boundaries there will be many optimizing (x_i, u_i) at each stage. It may therefore be expected that as the approximations improve an increasing number of solutions may have to be stored.

8.2 Lagrangian Relaxation

In Lagrangian relaxation the function π_i is linear, $\pi_i(x_i) = p_i x_i$. This makes the analysis particularly simple. For instance, any additive decomposability structure present in the original problem formulation is preserved. The weakness of Lagrangian relaxation is that it is not assured that the optimal solution may be found this way if the convexity structure of the problem is not present.

The simplest case of Lagrangian relaxation is attained when the solution to the relaxed problem is unique. We take this as underlying assumption in this section.

The problem considered is

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i)] \tag{8.20}$$

$$x_{i+1} = f_i(x_i, u_i) \tag{8.21}$$

$$(x_i, u_i) \in V_i \tag{8.22}$$

For notational simplicity we disregard a possible end constraint $x_N \in V_N$; if present in the original problem formulation, it may be incorporated into (8.22) by using the dynamic equation, as

$$f_{N-1}(x_{N-1}, u_{N-1}) \in V_N \tag{8.23}$$

Similarly any possible original term $r_N(x_N)$ is assumed incorporated in r_{N-1} using the dynamic equation. Consequently, $p_N \equiv 0$ in all the formulae below.

In most cases we shall assume that (8.22) is of the more structured form

$$g_i(x_i, u_i) \leq 0 \tag{8.24}$$

$$h_i(x_i, u_i) = 0 \tag{8.25}$$

Different Relaxations

We may consider relaxation of two types of constraints, either the local constraints (8.22) or the dynamic equation (8.21) (or both). Each choice may have its advantages and disadvantages. We shall consider all possibilities.

If the dynamic equation is relaxed, we get the Lagrangian

$$L = \sum_{i=0}^{N-1} r_i(x_i, u_i) + p_{i+1}(f_i(x_i, u_i) - x_{i+1}) \tag{8.26}$$

The maximization of the Lagrangian is subject to the local constraints (8.22). As seen, the problem decomposes into N independent problems, $i = 0, \ldots, N-1$,

$$\max_{x_i, u_i}[r_i(x_i, u_i) + p_{i+1} f_i(x_i, u_i) - p_i x_i] \tag{8.27}$$

$$(x_i, u_i) \in V_i \tag{8.28}$$

If the local constraints (8.22), of the assumed form (8.24) - (8.25), are relaxed, we get the Lagrangian function

$$L = \sum_{i=0}^{N-1} r_i(x_i, u_i) - \lambda_i g_i(x_i, u_i) - \mu_i h_i(x_i, u_i) \tag{8.29}$$

The maximization of the Lagrangian is subject to the dynamic constraints (8.21). As seen, the problem of maximization of the Lagrangian is an unconstrained optimal control problem.

Finally, if both the local constraints and the dynamics are relaxed, the Lagrangian is

$$L = \sum_{i=0}^{N-1} r_i(x_i,u_i) + p_{i+1}(f_i(x_i,u_i) - x_{i+1}) - \lambda_i g_i(x_i,u_i) - \mu_i h_i(x_i,u_i) \tag{8.30}$$

where the maximization decomposes into N independent unconstrained problems, $i = 0,\ldots,N-1$,

$$\max_{x_i,u_i}[r_i(x_i,u_i) + p_{i+1}f_i(x_i,u_i) - p_i x_i - \lambda_i g_i(x_i,u_i) - \mu_i h_i(x_i,u_i)] \tag{8.31}$$

The dual function D is defined as the optimal (with respect to (x,u)) value of the Lagrangian, i.e.

$$D = \max_{x,u}[L] \tag{8.32}$$

where L is defined in (8.26), (8.29) or (8.30). The constraints in (8.32) are those of the constraints (8.21) - (8.22) that are not relaxed, and D is a function of the multipliers corresponding to the relaxed constraints, i.e., it may be written as $D(p)$, $D(\lambda,\mu)$ and $D(p,\lambda,\mu)$ relative to (8.26), (8.29) and (8.30), respectively.

In Lagrangian relaxation the guiding principle for choice of the multipliers is that the dual function should be minimized. Let (x^o,u^o) be the optimal solution corresponding to a particular set of Lagrange multipliers. For the relaxation with Lagrangian (8.26) the dual function has subgradients with components

$$f_i(x_i^o,u_i^o) - x_{i+1}^o \tag{8.33}$$

corresponding to p_{i+1}. For the relaxation with Lagrangian (8.29) the dual function has subgradients with components

$$-g_i(x_i^o,u_i^o) \tag{8.34}$$
$$-h_i(x_i^o,u_i^o) \tag{8.35}$$

and for the relaxation with Lagrangian (8.30) the dual function has subgradients with components

$$f_i(x_i^o,u_i^o) - x_{i+1}^o \tag{8.36}$$
$$-g_i(x_i^o,u_i^o) \tag{8.37}$$
$$-h_i(x_i^o,u_i^o) \tag{8.38}$$

corresponding to p_{i+1}, λ_i and μ_i, respectively.

If the solution to the relaxed problem is unique then the values in (8.33) - (8.38) are gradients and the dual function is in fact differential.

From this, any method for gradient or subgradient minimization of the convex function D may be applied, see e.g. Kiwiel (1985) or Shor (1985). Necessary and sufficient conditions for a dual optimum are that the gradient vanishes - i.e. that the values in the expressions (8.33) - (8.38) vanish - or, in case of non-differential dual function that zero is included in the subdifferential.

Newton Iterations

Next consider the Newton iterations in (p, λ, μ) for the relaxation with Lagrangian (8.30). We may identify these as follows. Assume a given (p^k, λ^k, μ^k) and the corresponding maximizing (x^k, u^k). Strict complementarity is assumed, such that the active inequality constraints may be treated as equalities, while the nonactive inequality constraints are disregarded. Define the QLE problem as in (7.118) - (7.122) on page 214, using the matrices defined in (7.114) - (7.117), and using $p = p^k$ (rather than the p specified in (7.113)).

Proposition 8.2.1 *Assume that in the problem (8.20) - (8.21), (8.24) - (8.25) all functions are twice continuously differentiable. Assume a given (p^k, λ^k, μ^k) and the corresponding maximizing (x^k, u^k), and assume that strict complementarity holds. Formulate and solve the QLE problem (7.118) - (7.122) and find the associated multipliers $(p^{k+1}, \lambda^{k+1}, \mu^{k+1})$. If the solution and $(p^{k+1}, \lambda^{k+1}, \mu^{k+1})$ are unique then $(p^{k+1}, \lambda^{k+1}, \mu^{k+1})$ is the next (p, λ, μ) in the Newton iteration on the dual function.*

Proof. Consider the problem

$$\begin{aligned} &\max[r(u)] \\ &h(u) = 0 \end{aligned}$$

(corresponding to e.g. (7.22) - (7.24) on page 196; by suitable interpretation, this may be seen as the optimal control problem). This has the Lagrangian function

$$L(u, \mu) = r(u) - \mu h(u)$$

Assume that r and h are twice continuously differentiable. Assume a μ^k given with associated unique optimal relaxed solution u^k, implying $\nabla_u L(u^k, \mu^k) = 0$. The dual function has the gradient $-h(u^k)$ and the second order derivative $-\nabla h(u^k)[\nabla^2_{uu} L(u^k, \mu^k)]^{-1} \nabla h(u^k)'$, cf. Luenberger (1989), pp. 398-399. The Newton step in the dual problem then specifies the next μ^{k+1}, cf. Luenberger (1989), p. 225, as

$$\mu^{k+1} = \mu^k - (-\nabla h(u^k)[\nabla^2_{uu} L(u^k, \mu^k)]^{-1} \nabla h(u^k)')^{-1} h(u^k)$$

Consider alternatively the solution of the following quadratic-linear problem, formulated by approximation of the above problem around (u^k, μ^k):

$$\begin{aligned} \max_d [\tfrac{1}{2} d' \nabla_{uu} L(u^k, \mu^k) d + \nabla r(u^k) d] \\ \nabla h(u^k) d + h(u^k) = 0 \end{aligned}$$

and consider also finding the associated multiplier μ^{k+1}. The solution (d^*, μ^{k+1}) to this problem has the property that the derivative of the Lagrangian to this problem vanishes, i.e.,

$$d^{*\prime} \nabla^2_{uu} L(u^k, \mu^k) + \nabla r(u^k) - \mu^{k+1} \nabla h(u^k) = 0$$

and that the solution d is feasible. Therefore the quadratic-linear problem may be solved and the associated multiplier may be found by solving the system of linear equations

$$\begin{aligned} d' \nabla^2_{uu} L(u^k, \mu^k) + \nabla r(u^k) - \mu^{k+1} \nabla h(u^k) = 0 \\ \nabla h(u^k) d + h(u^k) = 0 \end{aligned}$$

However, if we let $u^{k+1} = u^k + d^*$, this is the application of Newton's method to the Lagrangian equations, and it follows that μ^{k+1} may be specified as

$$\mu^{k+1} = \mu^k + (\nabla h(u^k)[\nabla^2_{uu} L(u^k, \mu^k)]^{-1} \nabla h(u^k)')^{-1}$$
$$(h(u^k) - [\nabla h(u^k)(\nabla^2_{uu} L(u^k, \mu^k))^{-1}(-\nabla_u L(u^k, \mu^k))])$$

cf. Luenberger (1989), pp. 431-432. As u^k by assumption is optimizing the smooth function L, the gradient $\nabla_u L(u^k, \mu^k) = 0$ as already observed, and the last expression reduces to

$$\mu^{k+1} = \mu^k + (\nabla h(u^k)[\nabla^2_{uu} L(u^k, \mu^k)]^{-1} \nabla h(u^k)')^{-1} h(u^k)$$

As seen, the method of finding μ^{k+1} as the multiplier corresponding to the optimal solution in the quadratic-linear approximation specifies the same value as the Newton iteration in the dual function. □

It is interesting that although we formulate an optimization problem at the beginning of the proof, we do not actually have to find the solution; only the corresponding multipliers are needed. In Section 9.1 we shall specify a method, which actually finds the multipliers without finding (explicitly) the optimal solution.

We finally consider the Newton iteration for the two other types of relaxation, with Lagrangians (8.26) or (8.29); in the latter, we assume the local constraints of the form (8.24) - (8.25). We may also in these cases find the Newton iterations and attain the quadratic rate of convergence of the Newton method. It will be necessary to find all multipliers p^k, λ^k and μ^k, in addition to the optimal relaxed solution, such that the QLE problem (7.114) - (7.122) can be formulated.

The central steps of the Newton iterations with step length one for the three different relaxations therefore may therefore be formulated as follows:

- With Lagrangian (8.30): Step 1. Given $(p, \lambda, \mu) = (p^k, \lambda^k, \mu^k)$ find the optimal relaxed solution (x^k, u^k) by solving (8.31), $i = 0, \ldots, N-1$. Step 2. Formulate the QLE problem (7.114) - (7.122) page 214 and solve it to attain $(p^{k+1}, \lambda^{k+1}, \mu^{k+1})$. Let $k = k + 1$ and go to Step 1.

- With Lagrangian (8.26): Step 1. Given $p = p^k$ find the optimal relaxed solution (x^k, u^k) and the associated multipliers (λ^k, μ^k) by solving (8.24) - (8.25), (8.27). Step 2. Let $(\lambda, \mu) = (\lambda^k, \mu^k)$, find p from (7.113), formulate the QLE problem (7.114) - (7.122) page 214 and solve it to attain p^{k+1}. Let $k = k + 1$ and go to Step 1.

- With Lagrangian (8.29): Step 1. Given (λ^k, μ^k) find the optimal relaxed solution (x^k, u^k) and the associated multiplier p^k by solving the problem with criterion (8.29) and the dynamic constraint (8.21). Step 2. Formulate the QLE problem (7.114) - (7.122) using $(p, \lambda, \mu) = (p^k, \lambda^k, \mu^k)$ and solve it to attain $(\lambda^{k+1}, \mu^{k+1})$. Let $k = k + 1$ and go to Step 1.

Also the two last relaxations may attain the quadratic rate of convergence.

8.3 Linearity in the State Variables

It is a strong assumption in the above Proposition 8.2.1 that the optimal solution to the relaxed problem is unique. This is not fulfilled for many problems, typically because they are linear in the state. The implication of this is that the dual function is not differentiable, and both gradient and

Newton iterations are undefined. For the specific case of linear programming problems, Proposition 3.4.8 may be applied, and a unique solution is obtained. We now consider the case where the solution need not be unique.

Thus consider the problem that is additively separable in x and u and linear in x, i.e. the problem

$$\max[\sum_{i=0}^{N-1} R_i x_i + r_i^u(u_i) + R_N x_N] \tag{8.39}$$

$$x_{i+1} = F_i^x x_i + f_i^u(u_i) \tag{8.40}$$

$$G_i^x x_i + g_i^u(u_i) \leq 0 \tag{8.41}$$

$$H_i^x x_i + h_i^u(u_i) = 0 \tag{8.42}$$

For this problem the optimal x in a problem where the dynamic equation has been relaxed will not in general be unique. Therefore we may choose to relax the local constraints, in which case we work with the Lagrangian (8.29), as we shall do first in this section. Relaxation of the dynamic equation will require special attention and we consider this later in this section.

Relaxation of the Local Constraints

The idea may be seen as elimination of the state variables and then making a Lagrangian relaxation of the local constraints (8.41) - (8.42). Thus, we express x_i as

$$\begin{aligned} x_i = \quad & f_{i-1}(f_{i-2}(\ldots f_1(f_0(\underline{x}_0, u_0), u_1)\ldots, u_{i-2}), u_{i-1}) \\ = \quad & F_i^x(F_{i-1}^x(F_{i-2}^x(\ldots(F_1^x(F_0^x\underline{x}_0 + f_0^u(u_0)) + f_1^u(u_1)) \\ & \ldots) + f_{i-2}^u(u_{i-2})) + f_{i-1}^u(u_{i-1})) + f_i^u(u_i) \end{aligned} \tag{8.43}$$

It is here assumed that (8.41) - (8.42) for $i = 0$ specifies the unique initial state $\underline{x}_0$.

With this, and using the linearity with respect to x, the criterion may be reformulated as

$$\begin{aligned} & R_0\underline{x}_0 + \sum_{i=1}^{N} R_i^x(F_{i-1}^x(F_{i-2}^x(\ldots(F_1^x(F_0^x\underline{x}_0 + f_0^u(u_0)) + f_1^u(u_1)) \\ & \ldots) + f_{i-2}^u(u_{i-2})) + f_{i-1}^u(u_{i-1})) + \sum_{i=0}^{N-1} r_i^u(u_i) \\ & = R_0^x\underline{x}_0 + [R_1^x + (R_2^x + \ldots + (R_N^x F_{N-1}^x)\ldots)F_2^x)F_1^x]f_0^u(u_0) + r_0^u(u_0) \\ & + [R_2^x + (R_3^x + \ldots + (R_N^x F_{N-1}^x)\ldots)F_3^x)F_2^x]f_1^u(u_1) + r_1^u(u_1) \\ & \ldots \\ & + [R_{N-1}^x + (R_N^x F_{N-1}^x))]f_{N-2}^u(u_{N-2}) + r_{N-2}^u(u_{N-2}) \\ & + [R_N^x]f_{N-1}^u(u_{N-1}) + r_{N-1}^u(u_{N-1}) \end{aligned} \tag{8.44}$$

By introducing multipliers λ_i and μ_i corresponding to (8.41) - (8.42) the Lagrangian may be formulated by adding the following terms to the criterion

$$-\sum_{i=0}^{N-1} \lambda_i(G_i^x x_i + g_i^u(u_i)) - \sum_{i=0}^{N-1} \mu_i(H_i^x x_i + h_i^u(u_i)) \tag{8.45}$$

Again eliminating x using (8.43) the terms involving λ may be written

$$\begin{aligned}
&-\lambda_0 g_0^u(u_0) \\
&-[\lambda_1 G_1^x + \lambda_2 G_2^x F_1^x + \ldots + \lambda_{N-1} G_{N-1}^x F_{N-2}^x F_{N-3}^x \ldots F_1^x] f_0^u(u_0) \\
&-\lambda_1 g_1^u(u_1) \\
&-[\lambda_2 G_2^x + \lambda_3 G_3^x F_2^x + \ldots + \lambda_{N-1} G_{N-1}^x F_{N-2}^x F_{N-3}^x \ldots F_2^x] f_1^u(u_1) \\
&\ldots \\
&-\lambda_{N-3} g_{N-3}^u(u_{N-3}) \\
&-[\lambda_{N-2} G_{N-2}^x + \lambda_{N-1} G_{N-1}^x F_{N-2}^x] f_{N-3}^u(u_{N-3}) \\
&-\lambda_{N-2} g_{N-2}^u(u_{N-2}) \\
&-[\lambda_{N-2} G_{N-2}^x] f_{N-2}^u(u_{N-2}) \\
&-\lambda_{N-1} g_{N-1}^u(u_{N-1})
\end{aligned} \tag{8.46}$$

and similar reformulation hold for terms involving μ.

Now assume a (λ, μ) given. Define

$$p_N = R_N^x \tag{8.47}$$

and then recursively backwards, $i = N-1, \ldots, 1$,

$$p_i = R_i^x + p_{i+1} F_i^x - \lambda_i G_i^x - \mu_i H_i^x \tag{8.48}$$

(Observe that the previous expressions constitute the dynamics of an optimal control problem with p_i as states, (λ_i, μ_i) as controls, (8.48) as dynamic equation. Control are constrained to $\lambda_i \geq 0$. The problem runs "backwards" over the stages, with (8.47) as "initial" condition and no "end" condition on p_0. Compare also the dual OCP (1.87) - (1.91) page 34.)

Using this and observing the reformulations (8.44) and (8.46) the problem of maximizing the Lagrangian may be formulated as

$$\max_u [R_0 \underline{x}_0 + \sum_{i=0}^{N-1} r_i^u(u_i) + p_{i+1} f_i^u(u_i)] \tag{8.49}$$

and this again separates into N independent unconstrained problems, $i = 0, \ldots, N-1$

$$\max_{u_i} [r_i^u(u_i) + p_{i+1} f_i^u(u_i)] \tag{8.50}$$

The components of the gradient of the dual in this formulation are given as

$$-G_i^x x_i^k - g_i^u(u_i^k) \tag{8.51}$$

$$-H_i^x x_i^k - h_i^u(u_i^k) \tag{8.52}$$

with respect to λ_i and μ_i, respectively, if u^k is the unique solution to the relaxed problem (8.49) and x^k is calculated from u^k using the dynamics. If u^k is not unique, (8.51) - (8.52) is a component of an element in the subdifferential. Using this, gradient or subgradient iterations aiming to minimize the dual are formulated using (8.47) - (8.48).

We may then sketch an algorithm as follows:

Algorithm

Step 0 Choose arbitrary (λ^0, μ^0). Let iteration counter $k = 0$.

Step 1 For given (λ^k, μ^k) calculate p_i from (8.47) - (8.48).

Step 2 Find the maximizing u_i^k from (8.50), $i = 0, \ldots, N-1$.

Step 3 Find x_i^k corresponding to u^k from the dynamic equation (8.40).

Step 4 Find new $(\lambda^{k+1}, \mu^{k+1})$, let $k = k+1$ and go to Step 1.

If a gradient or subgradient algorithm is used then for a given stepsize α^k the components of $(\lambda^{k+1}, \mu^{k+1})$ are calculated in Step 4 as

$$\lambda^{k+1} = \lambda^k + \alpha^k(G_i^x x_i^k + g_i^u(u_i^k)) \tag{8.53}$$

$$\mu^{k+1} = \mu^k + \alpha^k(H_i^x x_i^k + h_i^u(u_i^k)) \tag{8.54}$$

subject to the restriction that λ^{k+1} must be nonnegative.

Control of the stepsize is performed in Step 4, such that the dual function (8.32) attains a sufficient increase, for instance as evaluated by the Armijo step size rule.

Newton iterations may be used when the active inequality constraints have been identified; nonactive constraints are neglected.

As seen, even if the problem is linear in the dynamics it may be possible to circumvent this difficulty and attain a smooth dual function and hence apply gradient and possibly also Newton iterations.

Relaxation of the Dynamic Equation

Now consider the relaxation of the dynamic equation. For this, we consider the problem

$$\max[\sum_{i=0}^{N-1} r_i(u_i)] \tag{8.55}$$

$$x_{i+1} = F_i^x x_i + f_i(u_i) \tag{8.56}$$

$$u_i \in U_i \tag{8.57}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{8.58}$$

$$x_0 = \underline{x}_0 \tag{8.59}$$

$$x_N = \underline{x}_N \tag{8.60}$$

The criterion (8.55) is more general than immediately appearing, because it may be obtained from the criterion (8.44) using the dynamic equation to eliminate the states, cf. (8.43). Thus, the specificity of (8.55) - (8.60) in relation to (8.39) - (8.42) is the separability between x_i and u_i of the local constraints.

For given p the Lagrangian is (8.26). The maximizing u_i^o can be found for each stage i independently of the other stages and independently of x as solution to

$$\max_{u_i}[r_i(u_i) + p_{i+1} f_i(u_i)] \tag{8.61}$$

subject to (8.57). The maximizing x_i^o are given as follows:

$$\text{if} \quad p_i^j \begin{cases} > (F_i^x p_{i+1})^j & \text{then} \quad (x_i^j)^o = \underline{x}_i^j \\ = (F_i^x p_{i+1})^j & \text{then} \quad \underline{x}_i^j \leq (x_i^j)^o \leq \overline{x}_i^j \text{ (arbitrary)} \\ < (F_i^x p_{i+1})^j & \text{then} \quad (x_i^j)^o = \overline{x}_i^j \end{cases} \tag{8.62}$$

A non-unique x^o implies that the subdifferential S is not a singleton but given as the set

$$S = \{s \in R^{Nn} | s_{i+1}^j = (F_i^x x_i^o)^j + f_i^j(u_i^o) - (x_{i+1}^j)^o\} \tag{8.63}$$

We shall now specify a dual descent algorithm. First we specify how to choose x_i^o for given p in case of nonunique solution. The idea in this is that in case of non-unique solution we choose $(x_{i+1}^o)^j$ to satisfy the dynamic equation as closely as possible.

Subalgorithm A: Forwards Sequential Projection.

Step 0 Let $x_0^o = \underline{x}_0$. Perform steps 1 to 3 for $i = 0, \ldots, N-1$:

Step 1 Calculate u_i^o from (8.61).

Step 2 Calculate $x_{i+1} = F_i^x x_i^o + f_i(u_i^o)$.

Step 3 Calculate x_{i+1}^o as follows:
if $p_{i+1}^j \neq (F_{i+1}^x p_{i+2})^j$ then use (8.62)
if $p_{i+1}^j = (F_{i+1}^x p_{i+2})^j$ then $(x_{i+1}^j)^o = \min\{\overline{x}_{i+1}^j, \max\{\underline{x}_{i+1}^j, x_{i+1}^j\}\}$.

Next we specify how to calculate a new set of $(p_i)^{k+1}$ from a set of $(p_i)^k$ such that we get an improving direction of the dual. This is possible despite the fact that the dual function is not smooth. The direction chosen is defined from the subgradient with components s_i given as

$$s_i = x_i - x_i^o \tag{8.64}$$

where x_i is given in Step 2 and x_i^o is given in Step 3 of the above Subalgorithm A. As seen the idea in Step 2 below is that the relation (8.62) should not be violated by the new value p^{k+1}.

Subalgorithm B: Backwards Sequential Projection.

Step 0 Given a steplength $0 < \alpha$. Let $(p_N)^{k+1} = (p_N)^k - \alpha s_N$. Perform steps 1 to 2 for $i = N-1, \ldots, 1$:

Step 1 Calculate $d_i = (p_i)^k - \alpha s_i$

Step 2 Calculate $(p_i)^{k+1}$ as follows:
if $(x_i^j)^o = \underline{x}_i^j$ and $s_i^j \neq 0$ then let $(p_i^{k+1})^j = \max\{d_i^j, (F_i^x p_{i+1}^{k+1})^j\}$
if $(x_i^j)^o = \overline{x}_i^j$ and $s_i^j \neq 0$ then let $(p_i^{k+1})^j = \min\{d_i^j, (F_i^x p_{i+1}^{k+1})^j\}$
otherwise let $(p_i^{k+1})^j = ((F_i^x p_{i+1})^j)^{k+1}$.

We can now specify the following Forwards Backwards Sequential Projection algorithm.

Algorithm

Step 0 Let $k = 0$. Choose $0 < \alpha_o$. Let $\alpha = \alpha_o$. Choose $0 < \beta < 1$. Choose p^0. Perform Subalgorithm A: Forwards Sequential Projection.

Step 1 Calculate s from (8.64).

Step 2 If $s = 0$ then stop, else go to Step 3.

Step 3 Perform Subalgorithm B: Backwards Sequential Projection.

Step 4 Perform Subalgorithm A: Forwards Sequential Projection.

Step 5 Calculate $D(p^{k+1})$.

Step 6 If $D(p^{k+1}) < D(p^k)$ then go to Step 7 else go to Step 8.

Step 7 Store $D(p^{k+1})$, and the associated solution, let $k = k + 1$, let $\alpha = \alpha_o$ and go to Step 1.

Step 8 Let $\alpha = \alpha\beta$ and go to Step 3.

Proposition 8.3.1 *Assume for the problem (8.55) - (8.60), where F_i^x is diagonal with positive elements in the diagonal, that the solution u^o to (8.61) is unique and Lipschitz continuous as a function of p and that all f_i^u are Lipschitz continuous. Then the algorithm either stops with a dual optimal p^* and the corresponding solution (x^*, u^*) to (8.55) - (8.60), or constructs a sequence p^k for which any accumulation point p^* solves the dual, and the corresponding (x^*, u^*) solves (8.55) - (8.60).*

Proof: The assumptions of Lipschitz continuous solution u^o and functions f_i^u imply that also $f_i^j(u_i(p)^o)$ is Lipschitz continuous in p. If α is sufficiently small and $(p_i^j)^k \neq ((F_i^x p_{i+1})^j)^k$ then also $(p_i^j)^{k+1} \neq ((F_i^x p_{i+1})^j)^{k+1}$, and therefore also $(x_i^j)^o = \underline{x}_i^j$ or $(x_i^j)^o = \overline{x}_i^j$ for this α and any α smaller than this. If $(p_i^j)^k = ((F_i^x p_{i+1})^j)^k$ then $(p_i^j)^{k+1} < ((F_i^x p_{i+1})^j)^{k+1}$ only if $(x_i^j)^o = \overline{x}_i^j$, and in this case $(x_i^j)^o$ remains $\overline{x}_i^j$ if α is sufficiently small. If $(p_i^j)^k = ((F_i^x p_{i+1})^j)^k$ then $(p_i^j)^{k+1} < ((F_i^x p_{i+1})^j)^{k+1}$ only if $(x_i^j)^o = \underline{x}_i^j$, and in this case $(x_i^j)^o$ remains $\underline{x}_i^j$ if α is sufficiently small. If $(p_i^j)^k = ((F_i^x p_{i+1})^j)^k$ and $(p_i^j)^{k+1} = ((F_i^x p_{i+1}^j)^j)^{k+1}$ then $(x_{i+1}^j)^o$ is a Lipschitz continuous function of (x_i^j) and $f_i^j(u_i(p))^o)$ since the function $\min\{\overline{x}_i^j, \max\{\underline{x}_i^j, x_{i+1}^j\}\}$ is Lipschitz continuous. In conclusion, all $(x_i^j)^o$ and (x_i^j) are Lipschitz continuous functions of p, for α sufficiently small. Therefore also $s = s(p(\alpha))$ defined in (8.64) is a Lipschitz continuous function of α for α sufficiently small, and the dual function $D(p(\alpha))$ is continuously differentiable with a Lipschitz bound on the change of the gradient as a function of α.

We now show that the direction $\delta p_i^j = (p_i^j)^{k+1} - (p_i^j)^k$ chosen is a improving direction for the dual function if $s \neq 0$. We show that this direction is improving by showing that

$$(\max_{\sigma \in S}[\sum_{i=0}^{N-1} \delta p_{i+1}\sigma_{i+1}]) < 0 \tag{8.65}$$

where $\sigma = (\sigma_1', \ldots, \sigma_N')'$, $\sigma_i \in R^n$ and S is the subdifferential (8.63).

Using (8.63), the optimization in (8.65) may be written

$$\max_{\chi}[\sum_{i=0}^{N-1} \delta p_{i+1}(F_i^x \chi_i + f_i(u_i^o) - \chi_{i+1})] \tag{8.66}$$

or, by rearrangement and reduction using (8.59) - (8.60),

$$\max_{\chi}[\sum_{i=1}^{N-1}(\delta p_{i+1}F_i^x - \delta p_i)\chi_i + f_i(u_i^o)] \tag{8.67}$$

subject to $\chi_i^j = (x_i^j)^o$, where u_i^o and $(x_i^j)^o$ are solutions to (8.61) - (8.62).

The coefficient to χ_i^j in (8.67) is seen to be $((\delta p_{i+1}F_i^x)^j - \delta p_i^j)$. If this coefficient is positive then the optimal χ_i^j, $(\chi_i^j)^o$, is uniquely given as $\chi_i^j = \overline{x}_i^j$; however, when the coefficient is positive, it is because $(x_i^j)^o = \overline{x}_i^j$ (Subalgorithm B, Step 2), and hence $(\chi_i^j)^o = (x_i^j)^o$. If the coefficient is negative then the optimal χ_i^j is uniquely given as $\chi_i^j = \underline{x}_i^j$; however, when the coefficient is negative, it is because $(x_i^j)^o = \underline{x}_i^j$ (Subalgorithm B, Step 2), and hence $(\chi_i^j)^o = (x_i^j)^o$. When the coefficient is zero, then the maximizing χ_i^j in (8.67) is arbitrary (within (8.58)), and in particular $\chi_i^j = (x_i^j)^o$ is optimal. As seen, in all cases $\chi_i^j = (x_i^j)^o$ is optimal in (8.67), and hence $\sigma_i = s_i$, where s_i is given in (8.64), is optimal in (8.65).

From Subalgorithm B (Step 1, Step 2) and the assumption on F_i^x it follows that $\delta p_i^j s_i^j \leq 0$, and therefore (8.65) holds if $s \neq 0$ and $\delta p \neq 0$. From Subalgorithm B it is seen that if $s \neq 0$ then $\delta p \neq 0$. We conclude that if $s \neq 0$ then (8.65) holds. This in turn implies that the direction δp is an improving direction, cf. Kiwiel (1985) p. 12.

Therefore with $p(\alpha)^{k+1}$ constructed from p^k by Backwards Sequential Projection, $p(\alpha)^{k+1} - p^k$ is a descent direction for the dual function. After a finite number of reductions of α in Step 8 of the algorithm we therefore have that $D(p(\alpha)^{k+1}) - D(p^k) < 0$. With this and the observation that the dual is continuous the result concerning the dual follows from Polak (1971) p. 14.

The result concerning (x^*, u^*) follows, since p^* is dual optimal only if a subgradient is vanishing. A vanishing subgradient means that the relaxed constraint (8.56) is fulfilled, due to the way that x^o is chosen, and therefore (x^*, u^*) is optimal in (8.55) - (8.60). □

A possible advantage of the relaxation of the dynamic equation, rather than of the local constraints is that the former may give a better indication of the indexes i and j for which stage constraints (8.58) will be active.

8.4 Conclusions

Decomposition may take two general forms, primal decomposition and dual decomposition. Primal decomposition with respect to the stage index i was treated in Section 5, and dual decomposition is the subject of this chapter.

Application of linear dual (price) functions, Lagrangian relaxation, has its obvious advantages. Solution of the relaxed subproblems is in general simple, the dual functions are represented by few parameters p_i, and the updating of the dual functions may be based on minimization of the dual with respect to the parameter values. In particular this is attractive if the solutions to the relaxed problems are known to be unique. This is so if the relaxed problems have strictly concave criterion functions. This situation was dealt with in Section 8.2 which discussed various relaxations and derived Newton iterations.

If the relaxed problems have concave criterion functions, but the solutions to the relaxed problems are not unique, special care must be taken. Still Lagrangian relaxation is valid for the derivation of necessary and sufficient optimality conditions, but it may be difficult to design algorithms for finding the optimal dual solution. This problem was dealt with in Section 8.3, where the case with linearity in the state variables was treated. Algorithms were designed for various types

of Lagrangian relaxation, in part based on careful selection among the nonunique solutions. In the next chapter the approach taken has some similarity with Lagrangian relaxation in the sense that linear price functions are used. The difficulties with nonunique solution in case of problems with concave, but not strictly concave, relaxed criterion functions may also in that chapter be seen as solved by careful selection among the nonunique solutions.

If the relaxed problem does not have a concave criterion function the Lagrangian relaxation will not provide necessary optimality conditions. In this case there seem to be three possible approaches. One is to base the solution (exact or approximate) on further calculations, such as embedding within a branch-and-bound procedure or applying heuristics. The upper bound attained from Lagrangian relaxation may be useful in any case.

The second approach is to convexify the problem, cf. the references in the introduction to the present chapter. This may provide for convergent algorithms, however, the solution found need not be optimal.

The third approach is to apply nonlinear dual functions. This was followed in Section 8.1. The advantage of this is that price functions which permit solution of the problems by decomposition exist, Proposition 3.5.3. The question is if they can be found. In Section 8.1 a parameterization idea was used towards this. It was shown that for a weighted sum of given price functions the dual function is convex as a function of the weights. This result opens up for subgradient and gradient algorithms in the same way as for Lagrangian relaxation; it is shown that in fact Lagrangian relaxation may be seen as a special case.

The price functions applied obviously should be selected according to the properties of the upper boundaries, as this is discussed in Proposition 3.5.4. On the other hand, a compromise has to be made between simplicity of the calculations and solution accuracy. As Lagrangian relaxation exemplifies, the price functions may be simpler than the upper boundaries, and yet solve certain problems (viz., those with concave upper boundaries). Finally observe that also in the case of nonlinear price functions is it possible to apply branch-and-bound ideas, cf. Proposition 3.5.7.

Chapter 9

Forwards Algorithms

This chapter deals with forwards algorithms. The idea in these is to construct trajectories forwards from the given initial point $\underline{x}_0$. The aim is to find a trajectory that ends at the given $\underline{x}_N$. Typically the different trajectories are parameterized by p_1. Thus, we shoot forwards from $\underline{x}_0$, aiming with p_1 and trying to hit $\underline{x}_N$. The construction of the forwards trajectories is done (using sufficient optimality conditions) such that if $\underline{x}_N$ is indeed hit then the corresponding trajectory is optimal for the problem. In distinction to forwards DP, Section 4.7, UB_i is not constructed, only particular points of $\{UB_i\}$ are found.

The idea was developed in Roberts and Shipman (1972) for continuous time two points boundary value problems and the application to discrete time optimal control problems was demonstrated in Sethi and Thompson (1981). The methods did not apply to problems with state constraints. Kleindorfer and Lieber (1979) and Ravn (1987) extended the idea to problems with state constraints and $n = 1$. See the survey on forwards methods in mathematical programming in Aronson and Thompson (1984).

Forwards algorithms were also developed in relation to more specific areas, such as the production planning problem in e.g. Whagner and Whitin (1958), extended in Zabel (1964) and Eppen, Gould and Pashigian (1969), see further the review in Florian, Lenstra and Kan (1980).

The forwards methods may benefit from the existence and exploitation of decision and forecast horizons. Thus, if the solution u_i^* for $i = 0, \ldots, t_1$ is independent of data (r_i, f_i, V_i) for periods later than t_2 (where $t_1 \leq t_2$) then t_1 is a decision horizon and t_2 is a forecast horizon; sometimes this is discussed under the name of planning horizon. Thus, if a decision horizon t_1 is detected, we may solve the problem by considering only periods 0 through t_2, reset the initial index from $i = 0$ to $i = t_1 + 1$, and repeat the process. Therefore the existence and detection of decision and forecast horizons permit the decomposition of the original problem into a sequence of smaller problems.

In this chapter we develop forwards methods based on the sufficient maximum principle (Lagrangian relaxation), Proposition 3.5.6. In Section 9.1 we apply it to the QLE problem by developing an algorithm and finding its computational complexity.

The remaining part of the chapter deals with the more difficult problems that have inequality constraints. In Section 9.2 we pursuit the case with $n = 1$ and upper and lower limits on the control and state variables. Computational complexity results are developed for the quadratic criterion case. The section contains a discussion of the case of non-convex problems, where the close connection to price relaxation is exploited for deriving properties of approximate solutions. Also solution structure is derived and linked to the concept of supermodularity.

In Section 9.3 the linear problem is treated, first with $n = 1$, then in more generality. The expected difficulties of nonuniqueness (singularities) of the intermediate solutions are handled by carefully keeping track of all the solutions to the intermediate problems.

The algorithms of Sections 9.2 and 9.3 exploit decision and forecast horizons, whenever they are detected. Traditionally, results on these horizons are derived for $n = 1$; in Section 9.4 we give results for $n > 1$.

Finally observe that we have already in Section 1.7 given an implementation of a forwards algorithm for the isotone regression problem.

9.1 The QLE Problem

In this section we consider the problem with quadratic criterion function, linear dynamics and linear local equality constraints (the QLE problem).

This problem is interesting in itself, and also as a subproblem in a solution strategy for more complicated problems, based on repeated solution of quadratic-linear approximations and an active set strategy.

We therefore consider the QLE problem given as

$$\begin{aligned}
&\max[\sum_{i=0}^{N-1} \tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^x x_i + R_i^u u_i && (9.1)\\
&+\tfrac{1}{2}x_N'R_N^{xx}x_N + R_N^x x_N]\\
&x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i && (9.2)\\
&H_i^x x_i + H_i^u u_i - \overline{h}_i = 0 && (9.3)\\
&x_0 = \underline{x}_0 && (9.4)
\end{aligned}$$

with end conditions either free or fixed at $x_N = \underline{x}_N$.

We may express sufficient optimality conditions for a concave criterion function according to Proposition 3.5.6 as follows. The Lagrangian at stage i is defined as

$$\begin{aligned}
L = \quad & \tfrac{1}{2}x_i'R_i^{xx}x_i + x_i'R_i^{xu}u_i + \tfrac{1}{2}u_i'R_i^{uu}u_i + R_i^x x_i + R_i^u u_i && (9.5)\\
&+p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i) - p_i x_i - \mu_i(H_i^x x_i + H_i^u u_i - \overline{h}_i)
\end{aligned}$$

The conditions for stationarity with respect to u_i and x_i are

$$\begin{aligned}
x_i'R_i^{xu} + u_i'R_i^{uu} + R_i^u + p_{i+1}F_i^u - \mu_i'H_i^u = 0 && (9.6)\\
x_i'R_i^{xx} + u_i'R_i^{ux} + R_i^x + p_{i+1}F_i^x - \mu_i H_i^x - p_i = 0 && (9.7)
\end{aligned}$$

and this, together with the local feasibility condition (9.3) may we written as

$$\begin{aligned}
&\begin{pmatrix} R_i^{uu} & F_i^{u\prime} & -H_i^{u\prime} \\ R_i^{xu} & F_i^{x\prime} & -H_i^{x\prime} \\ H_i^u & 0 & 0 \end{pmatrix} \begin{pmatrix} u_i \\ p_{i+1}' \\ \mu_i \end{pmatrix} = && (9.8)\\
&\begin{pmatrix} -R_i^{xu} \\ -R_i^{xx} \\ -H_i^x \end{pmatrix} x_i + \begin{pmatrix} 0 \\ I^n \\ 0 \end{pmatrix} p_i' + \begin{pmatrix} -R_i^u \\ -R_i^x \\ \overline{h}_i \end{pmatrix}
\end{aligned}$$

where I^n is the $n \times n$ identity matrix.

Assume that x_i and p_i may be expressed as linear functions of p_1, i.e.,

$$x_i = K_i^x + L_i^x p_1' \tag{9.9}$$
$$p_i' = K_i^p + L_i^p p_1' \tag{9.10}$$

where K_i^x, L_i^x, K_i^p, and L_i^p are matrices of dimensions $n \times 1$, $n \times n$, $n \times 1$, and $n \times n$, respectively.

The solution of (9.8) for (u_i, p_{i+1}, μ_i) may then be written

$$\begin{pmatrix} u_i \\ p_{i+1}' \\ \mu_i \end{pmatrix} = \begin{pmatrix} R_i^{uu} & F_i^{u\prime} & -H_i^{u\prime} \\ R_i^{xu} & F_i^{x\prime} & -H_i^{x\prime} \\ H_i^u & 0 & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} R_i^{xu} \\ R_i^{xx} \\ -H_i^x \end{pmatrix} (K_i^x + L_i^x p_1') + \begin{pmatrix} 0 \\ I^n \\ 0 \end{pmatrix} (K_i^p + L_i^p p_1') + \begin{pmatrix} -R_i^u \\ -R_i^x \\ \overline{h}_i \end{pmatrix} \right] \tag{9.11}$$

The right hand side is seen to be a linear expression in p_1. Identifying the solution for u_i, corresponding to the upper m rows by matrices K_i^u and L_i^u of dimensions $m \times 1$, and $m \times n$, respectively, it may be written as

$$u_i = K_i^u + L_i^u p_1' \tag{9.12}$$

Similarly p_{i+1} may be identified from (9.11) by two $n \times 1$, and $n \times n$ matrices as

$$p_{i+1}' = K_{i+1}^p + L_{i+1}^p p_1' \tag{9.13}$$

From the dynamic equation we may now find x_{i+1} as

$$\begin{aligned} x_{i+1} &= F_i^x x_i + F_i^u u_i + \overline{f}_i \\ &= F_i^x (K_i^x + L_i^x p_1') + F_i^u (K_i^u + L_i^u p_1') + \overline{f}_i \\ &\equiv K_{i+1}^x + L_{i+1}^x p_1' \end{aligned} \tag{9.14}$$

We see that p_{i+1} and x_{i+1} may be expressed as linear functions of p_1, cf. (9.13) and (9.14). Since x_1 through (9.11), (9.2) and (9.4) may be expressed as a linear function of p_1, we may continue this way forwards from given $\underline{x}_0$ and p_1 to get x_N expressed by p_1.

If the end condition is given as $x_N = \underline{x}_N$ we solve the equation

$$K_N^x + L_N^x p_1' = \underline{x}_N \tag{9.15}$$

to find the optimal p_1^* as

$$p_1^{*\prime} = (L_N^x)^{-1} (\underline{x}_N - K_N^x) \tag{9.16}$$

If the end point is free then from the term $r_N(x_N) = \frac{1}{2} x_N' R_N^{xx} x_N + R_N^x x_N$ we require $p_N = \nabla r_N(x_N)$, i.e.,

$$K_N^p + L_N^p p_1' = R_N^{xx} (K_N^x + L_N^x p_1') + R_N^{x\prime} \tag{9.17}$$

which we solve for p_1^* as

$$p_1^{*\prime} = (L_N^p - R_N^{xx} L_N^x)^{-1} (R_N^{xx} K_N^x + R_N^{x\prime} - K_N^p) \tag{9.18}$$

Knowing p_1^*, the optimal solution may be found from (9.12) and (9.14). If the corresponding p^* is needed this may be found from (9.13).

For initialization we use (9.3) and (9.6) with $i = 0$ and $x_i = \underline{x}_0$:

$$\begin{pmatrix} u_i \\ \mu_i \end{pmatrix} = \begin{pmatrix} R_i^{uu} & -H_i^{u\prime} \\ H_i^u & 0 \end{pmatrix}^{-1} \left[\begin{pmatrix} R_i^{xu} \\ -H_i^x \end{pmatrix} x_i + \begin{pmatrix} -R_i^u - p_{i+1}F_i^u \\ \overline{h}_i \end{pmatrix} \right] \tag{9.19}$$

and then K_1^x and L_1^x are identified from (9.14).

It is seen that the computations may be interpreted as exploitation of the band structure of the set of linear equations that constitute the necessary and sufficient optimality conditions.

We may summarize the above procedure as follows:

Algorithm

Step 1 Choose an arbitrary p_1. Find u_0 from (9.19) and x_1 from (9.14).

Step 2 Define recursively forwards, $i = 1, \ldots, N-1$, K_i^u, L_i^u, K_{i+1}^p, and L_{i+1}^p from (9.11) - (9.13) and x_{i+1} from (9.14).

Step 3 Find p_1^* from the end condition, (9.16) or (9.18)

Step 4 Find the optimal solution from (9.12) and (9.14) and the corresponding p^* from (9.13).

Now, if $R_i^{xu} = 0$ and $H_i^x = 0$ for all i, the solution simplifies because the system (9.8) may be solved independently in p_{i+1} and (u_i, μ_i). Thus, p_{i+1} is found as

$$p_{i+1}' = (F_i^x)^{-1} (p_i' - R_i^{xx} x_i + R_i^{x\prime}) \tag{9.20}$$

and u_i is then found as in (9.19).

We summarize as follows:

Proposition 9.1.1 *Assume that the QLE problem (9.1) - (9.4) has a concave criterion and that the matrix on the left in (9.8) is nonsingular for all i. Assume that the inverted matrix in (9.16) (if a fixed end point is given) or (9.18) (if the end point is free) is nonsingular. Then the described algorithm solves the problem. It has a computational complexity of $O(N(m+n+\ell)^3)$. If it is further assumed that $R_i^{xu} = 0$ and $H_i^x = 0$ for all i, then the computational complexity is $O(N((m+\ell)^3 + n^3))$.*

Proof. The optimality of the solution is based on Proposition 3.5.6. Due to concavity of the criterion function and linearity of all constraints, stationarity of the Lagrangian implies maximization of the Lagrangian. The stationarity conditions are secured in (9.11) and the feasibility also in (9.11) (the local constraints), as well as in (9.14) (the dynamic equation) and (9.16) (with fixed end point; with free end point, (9.18) assures optimality). The computational complexity results follows by considering the arithmetic operations involved and observing that the matrix inversions in (9.11) or (9.19) - (9.20) are dominating. □

Comparing with forwards dynamic programming as described in Section 4.7 the forwards maximum principle as described here is attractive in terms of computational complexity.

Finally observe that we may construct an explicit expression for UB_N. Thus, for given x_N we find p_1^* from (9.16) and then x_i and u_i from (9.14) and (9.12). Inserting these into the criterion function (9.1) we get the optimal criterion value for this particular x_N. It is seen to be a quadratic function which by appropriate definitions may be written as

$$UB_N(x_N) = x_N' Q_N x_N + P_N x_N + \rho_N \tag{9.21}$$

9.2 The Ansgar Algorithm

Now we consider the following problem with simple state constraints:

$$\max[\sum_{i=0}^{N-1} r_i(x_i, u_i) + r_N(x_N)] \tag{9.22}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{9.23}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{9.24}$$

$$\underline{u}_i \leq u_i \leq \overline{u}_i \tag{9.25}$$

$$x_0 = \underline{x}_0 \tag{9.26}$$

$$x_N = \underline{x}_N \tag{9.27}$$

In order to proceed with the analysis, it is assumed that $n = 1$; in Section 9.4 this will be relaxed. Also $m = 1$ is assumed, however, as discussed in Section 4.6 this may be the result of a transformation of a problem that originally had $m > 1$. It is assumed that the criterion function is concave. As in the previous section, we base the analysis on the sufficient maximum principle, Proposition 3.5.6

Thus, the optimal $u_i^*(x_i, p_{i+1})$ (depending on (x_i, p_{i+1})) maximizes the Hamiltonian, i.e., it solves

$$\max_{u_i}[r_i(x_i, u_i) + p_{i+1}(F_i^x x_i + F_i^u u_i + \overline{f}_i)] \tag{9.28}$$

subject to (9.25). Further, if (x_i^*, u_i^*) is maximizing the Lagrangian this implies under conditions of differentiability that

$$\nabla_x r_i(x_i^*, u_i^*) + p_{i+1} F_i^x - p_i \begin{cases} \leq 0 & \text{if} \quad \underline{x}_i = x_i^* \\ = 0 & \text{if} \quad \underline{x}_i < x_i^* < \overline{x}_i \\ \geq 0 & \text{if} \quad x_i^* = \overline{x}_i \end{cases} \tag{9.29}$$

Stated otherwise it implies, assuming $F_i^x > 0$,

$$p_{i+1} \begin{cases} \leq (F_i^x)^{-1}(p_i - \nabla_x r_i(x_i^*, u_i^*)) & \text{if} \quad \underline{x}_i = x_i^* \\ = (F_i^x)^{-1}(p_i - \nabla_x r_i(x_i^*, u_i^*)) & \text{if} \quad \underline{x}_i < x_i^* < \overline{x}_i \\ \geq (F_i^x)^{-1}(p_i - \nabla_x r_i(x_i^*, u_i^*)) & \text{if} \quad x_i^* = \overline{x}_i \end{cases} \tag{9.30}$$

As seen from this, it is not possible to derive p_{i+1} from p_i, except in the middle option of (9.30). The forwards idea therefore has to be adapted to this situation.

The idea in the Ansgar algorithm can be indicated as follows. Choose p_1 and construct a forwards trajectory. If we can in this way come to stage N without violating any of the state

restrictions (9.24) and if we hit $\underline{x}_N$ then the solution is optimal since it satisfies the sufficient optimality conditions. Otherwise, i.e. when state restrictions (9.24) are violated, we have to adjust p_1. The idea now is to adjust p_1 so that we can come as far as possible (in terms of the stage index i) without violating the state constraints.

Therefore, we aim at one of the following two situations, upper or lower tangency:

UT: (9.24) holds for $i = 0, \ldots, j-1$, but $x_j < \underline{x}_j$, and for some index t with $0 < t < j$, we have *upper tangency*, i.e. $x_t = \overline{x}_t$. (If the stage index t of tangency is not uniquely determined we take the largest index.)

LT: (9.24) holds for $i = 0, \ldots, j-1$, but $x_j > \overline{x}_j$, and for some index t with $0 < t < j$, we have *lower tangency*, i.e. $x_t = \underline{x}_t$. (If the stage index t of tangency is not uniquely determined we take the largest index.)

A key point is now the observation that with tangency at stage t we know the optimal strategy and trajectory from stage 0 up to and including x_t. More on this in Section 9.4.

We can in any of these situations shift the initial stage forwards. The index t becomes the initial index and the initial state is the x_t found ($\underline{x}_t$ or $\overline{x}_t$). We then shoot with p_{t+1} trying again to hit $\underline{x}_N$. If this brings us to the final stage N the problem is solved, provided (9.30) holds; otherwise we aim again at a tangency situation. Conditions that assure that (9.30) hold will be considered next.

Monotonicity relations

The forwards construction of the solution according to stationarity relations are guided by the attempt to keep x_i feasible with respect to (9.24). This is done by adjusting p_t. In order to know which way to adjust p_t we need to consider the relationships between p_t and x_i, $t \leq i$.

Proposition 9.2.1 *Assume that r_i is additively separable in x_i and u_i for all i, i.e., $r_i(x_i, u_i) = r_i^x(x_i) + r_i^u(u_i)$, that $r_i^x(x_i)$ is concave and continuously differentiable and r_i^u is concave (but not necessarily differentiable). Assume that $F_i^x > 0$. Consider a forwards trajectory, constructed disregarding the state constraints (9.24) and using for $u_i^*(p_{i+1})$ either the largest or the smallest value of the solution in (9.28) (in case of non-unique solution). Then the following hold.*

(i) $F_i^u u_i^*(p_{i+1})$ is a non-decreasing function of p_{i+1}. (ii) p_{i+1} as given by the middle option of the adjoint relation (9.30) is a linearly increasing function of p_i and a continuous non-decreasing function of x_i. (iii) x_i is a non-decreasing function of p_t, $t \leq i$.

Proof. (i) The condition that $u_i^*(p_{i+1})$ maximizes the Hamiltonian implies for given p_{i+1}^o and p_{i+1}^+ that

$$r_i^u(u_i^*(p_{i+1}^o)) + p_{i+1}^o F_i^u u_i^*(p_{i+1}^o) \geq r_i^u(u_i^*(p_{i+1}^+)) + p_{i+1}^o F_i^u u_i^*(p_{i+1}^+)$$

$$r_i^u(u_i^*(p_{i+1}^+)) + p_{i+1}^+ F_i^u u_i^*(p_{i+1}^+) \geq r_i^u(u_i^*(p_{i+1}^o)) + p_{i+1}^+ F_i^u u_i^*(p_{i+1}^o)$$

which can be combined and then reduced to

$$(p_{i+1}^o - p_{i+1}^+)(F_i^u u_i^*(p_{i+1}^o) - F_i^u u_i^*(p_{i+1}^+)) \geq 0$$

This shows that $p_{i+1}^o < p_{i+1}^+$ implies $F_i^u u_i^*(p_{i+1}^o) \leq F_i^u u_i^*(p_{i+1}^+)$. (ii) $F_i^x > 0$ implies the first result. The concavity and continuous differentiability of r_i^x gives the second result. (iii) The last result follows by induction and combination of the two previous results. □

Observe that any assumptions that can lead to these monotonicity properties are sufficient for the establishment of convergence below. Thus for instance the control dependent part of the dynamic function, f_i^u, need not be linear, if only the combination of f_i^u and r_i^u is such that the conclusion above may be otherwise established.

We attained the above results assuming additive separability between x_i and u_i. We now relax this assumption. We assume that the criterion function is concave and quadratic, as in (9.1), but still with $n = m = 1$.

For simplicity we first assume that there are no constraints (9.25) on u_i or x_i. For given x_i and p_i the stationarity and adjoint conditions are

$$x_i' R_i^{xu} + u_i' R_i^{uu} + R_i^u + p_{i+1} F^u = 0 \tag{9.31}$$

$$p_i = x_i' R_i^{xx} + R_i^{xu} u_i + R_i^x + p_{i+1} F_i^x \tag{9.32}$$

This system of two equations in the variables (u_i, p_{i+1}) may be written

$$\begin{pmatrix} R_i^{uu} & F_i^u \\ R_i^{xu} & F_i^x \end{pmatrix} \begin{pmatrix} u_i \\ p_{i+1} \end{pmatrix} = \begin{pmatrix} -R_i^{xu} \\ -R_i^{xx} \end{pmatrix} x_i + \begin{pmatrix} 0 \\ 1 \end{pmatrix} p_i + \begin{pmatrix} -R_i^u \\ -R_i^x \end{pmatrix} \tag{9.33}$$

We assume $R_i^{uu} < 0$ and $F_i^x R_i^{uu} - F_i^u R^{xu} \neq 0$ and then solve (9.33) as

$$\begin{pmatrix} u_i \\ p_{i+1} \end{pmatrix} = \begin{pmatrix} a \\ b \end{pmatrix} x_i + \begin{pmatrix} c \\ d \end{pmatrix} p_i + \begin{pmatrix} e \\ f \end{pmatrix} \tag{9.34}$$

Here a, b, c, d, e and f are given as

$$a = -(R_i^{uu})^{-1} \left(R_i^{xu} + F_i^u \left(\frac{(R_i^{xu})^2 - R_i^{xx} R_i^{uu}}{F_i^x R_i^{uu} - F_i^u R_i^{xu}} \right) \right) \tag{9.35}$$

$$b = \frac{(R_i^{uu})^2 - R_i^{xx} R_i^{uu}}{F_i^x R_i^{uu} - F_i^u R_i^{uu}} \tag{9.36}$$

$$c = \frac{-F_i^u}{F_i^x R_i^{uu} - F_i^u R_i^{xu}} \tag{9.37}$$

$$d = \frac{R_i^{uu}}{F_i^x R_i^{uu} - F_i^u R_i^{xu}} \tag{9.38}$$

$$e = -(R_i^{uu})^{-1} \left(R_i^u + F_i^u \left(\frac{-R_i^x R_i^{uu} + R_i^u R_i^{xu}}{F_i^x R_i^{uu} - F_i^u R_i^{xu}} \right) \right) \tag{9.39}$$

$$f = \frac{-R_i^x R_i^{uu} + R_i^u R^{xu}}{F_i^x R_i^{uu} - F_i^u R_i^{uu}} \tag{9.40}$$

We now ask for conditions under which p_{i+1} as given in (9.34) is non-decreasing with p_i and x_i and under which conditions x_{i+1} as given by the dynamic equation (9.23) is non-decreasing with p_i and x_i.

As presented in (9.34) - (9.38) the answer is simple: sufficient conditions are that $F_i^u a \geq 0$, $b \geq 0$, $F_i^u c \geq 0$ and $d \geq 0$.

Also in the case of simple constraints $\underline{u}_i \leq u_i \leq \overline{u}_i$ these conditions are seen to be valid.

Proposition 9.2.2 *Assume for the quadratic criterion (9.1) with $n = m = 1$ that $F_i^x > 0$, that $R_i^{uu} < 0$ and $F_i^x R_i^{uu} - F_i^u R^{xu} \neq 0$ and that $F_i^u a \geq 0$, $b \geq 0$, $F_i^u c \geq 0$ and $d \geq 0$. Consider a forwards trajectory, constructed disregarding the state constraints (9.24). Then p_{i+1} is non-decreasing with p_i and with x_i, and x_{i+1} is non-decreasing with p_i and with x_i.*

Proof. The assumption $R_i^{uu} < 0$ and $F_i^x R_i^{uu} - F_i^u R^{xu} \neq 0$ ensures that (9.33) may be solved as in (9.34). The result for p_{i+1} are obvious by considering (9.34) and the assumptions $b \geq 0$ and $d \geq 0$. Now, from the assumption $F_i^x > 0$, x_{i+1} is increasing with x_i. From (9.34), and from the assumptions $F_i^u a \geq 0$ and $F_i^u c \geq 0$, $F_i^u u_i$ is non-decreasing with p_i and with x_i, even considering the control constraints (9.25). Therefore from the dynamic equation (9.23) the result for x_{i+1} holds. □

Let us analyze the conditions (9.35) - (9.38) under a variant of the assumptions of Proposition 9.2.2, viz., the assumptions that $R_i^{uu} < 0$, $F_i^u a \geq 0$, $b \geq 0$, $F_i^u c \geq 0$, $d \geq 0$, $F_i^x = 1$ and $F_i^u = 1$. We see that (9.37) implies $R_i^{uu} \leq R_i^{xu}$, which, together with $R_i^{uu} < 0$, is also the implication of (9.38). (9.36) and $R_i^{uu} - R_i^{xu} < 0$ imply $R_i^{xx} R_i^{uu} - (R_i^{xu})^2 \geq 0$. We observe that with $R_i^{uu} < 0$ this is the condition ensuring that the matrix $\nabla^2 r_i \leq 0$, i.e., that the criterion function is concave. Finally (9.35), $R_i^{uu} < 0$ and $R_i^{uu} \leq R_i^{xu}$ imply $R_i^{xx} \leq R_i^{xu}$. In conclusion we see that the assumptions imply that $R_i^{uu} \leq R_i^{xu}$ and $R_i^{xx} \leq R_i^{xu}$.

Further insight into the meaning of the first one of these implied conditions may be attained as follows. Given the dynamics with $F_i^x = F_i^u = 1$ and disregarding constraints we get the smaller upper boundary function as follows by eliminating u_i:

$$\begin{aligned} ub_i^{i+1}(x_i, x_{i+1}) &= \tfrac{1}{2} R_i^{xx}(x_i)^2 + R_i^{xu}(x_{i+1} - x_i - \overline{f}_i)x_i \\ &+ \tfrac{1}{2} R_i^{uu}(x_{i+1} - x_i - \overline{f}_i)^2 + R_i^x x_i + R_i^u(x_{i+1} - x_i - \overline{f}_i) \end{aligned} \tag{9.41}$$

From this we find

$$\nabla^2_{x_i x_{i+1}} ub_i^{i+1}(x_i, x_{i+1}) = R_i^{xu} - R_i^{uu} \tag{9.42}$$

Now recall the definition of a supermodular function, cf. e.g. Denardo, Huberman and Rothblum (1982), Ross (1989). We say that $g : R^2 \to R$ is supermodular if

$$g(x_1, y_1) + g(x_2, y_2) \geq g(x_1, y_2) + g(x_2, y_1) \tag{9.43}$$

for all $x_1 > x_2$, $y_1 > y_2$.

If g is twice continuously differentiable then this is tantamount to

$$\nabla^2_{xy} g \geq 0 \tag{9.44}$$

An important consequence of supermodularity is that

$$\arg\max_y [g(x, y)] \tag{9.45}$$

is non-decreasing with x. Optimization problems of the form $\max[\sum_{i=1}^{N-1} g(x_i, x_{i+1})]$ display a nice solution structure linked to monotonicity, as may be guessed from (9.45), cf. e.g. Topkis (1978), Denardo, Huberman and Rothblum (1982), Vidal (1994).

The optimization problem considered may be written as $\max[\sum_{i=0}^{N-1} ub_i^{i+1}(x_i, x_{i+1})]$, cf. Proposition 3.3.5. These solution structures will therefore be satisfied for our problem if all ub_i^{i+1} are identical and ub_i^{i+1} supermodular. This is fulfilled if all ub_i^{i+1} are identical and if ub_i^{i+1} satisfies

(9.43) or (9.44). As seen from (9.42) the condition (9.44) is $R_i^{xu} - R_i^{uu} \geq 0$. But this is precisely one of the conditions implied. In other words, we have interpreted in the optimal control environment the identification and significance of supermodularity.

In the present context, with $F_i^x = 1$, $F_i^u = 1$, further assumptions as in Proposition 9.2.2 and all parameters independent of i, the above mentioned monotonicity solution structure translates to the insight that if $\underline{x}_N$ is increased (decreased) then u_i^* is not decreased (increased) for any i. Similar observations hold if $\underline{x}_0$ is taken as parameter.

Implementations

We now present some ways to implement the Ansgar algorithm. We first state the Ansgar algorithm as follows.

Algorithm

Step 0 Let the start index $s = 0$. Let the start state $x_s = \underline{x}_0$.

Step 1 Bound the costate vector p_{s+1}: Find $\underline{p}_{s+1}$ such that a forwards trajectory violates $\underline{x}_j \leq x_j$ first for some j. Find $\overline{p}_{s+1}$ such that a forwards trajectory violates $x_j \leq \overline{x}_j$ first for some j.

Step 2 Let $p_{s+1} = \frac{1}{2}(\overline{p}_{s+1} + \underline{p}_{s+1})$.

Step 3 Generate a forwards trajectory as far as necessary to detect one of the following cases:

Case 1 $\overline{x}_j \leq x_j$ is first violated for some j without prior lower tangency: Let $\overline{p}_{s+1} = p_{s+1}$ and go to Step 2.

Case 2 $\underline{x}_j \leq x_j$ is first violated for some j without prior upper tangency: Let $\underline{p}_{s+1} = p_{s+1}$ and go to Step 2.

Case 3 Upper tangency in period t and violation of $\underline{x}_j \leq x_j$ in a later period j: Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \overline{x}_s$. Go to Step 1.

Case 4 Lower tangency in period t and violation of $\overline{x}_j \leq x_j$ in a later period j: Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \underline{x}_s$. Go to Step 1.

Case 5 $\underline{x}_N$ is reached: Store the optimal strategy and trajectory from stage s to stage N. Stop.

Proposition 9.2.3 *Assume as in Proposition 9.2.1 or as in Proposition 9.2.2. Assume that tangency may be attained in a finite number of iterations. Then the algorithm constructs an optimal solution in a finite number of steps.*

Proof. Since there are only a finite number of periods and therefore a finite number of potential tangency points the algorithm will terminate in a finite number of steps.

All the u_i considered are feasible in (9.25). Further, the dynamic equation is kept feasible from $i = 0$ and up to the stage considered in the iterations, and therefore at termination the dynamic equations will be fulfilled for all i. At termination also all state constraints (9.24) will be fulfilled.

During all stages of the algorithm, the u_i considered are maximizing the Lagrangian for given x_i. If the adjoint relations (9.30) hold, the x_i found are maximizing the Lagrangian for the

u_i considered. Due to the concavity this means that the (x_i^*, u_i^*) constructed maximizes the Lagrangian for all i.

Thus we need only to show that the adjoint relations (9.30) hold in order to show that the solution constructed is optimal.

By the construction, (9.30) holds at stage i if $\underline{x}_i < x_i < \overline{x}_i$. Therefore now consider the situation when a tangency situation occurs. Assume first that it is upper tangency occurring at stage t, implying $x_j < \underline{x}_j$ (Step 3, Case 3 of the algorithm). In order to make x_j feasible with respect to (9.24) it is necessary to increase $F_i^u u_i$ for one or more i, $t \leq i < j$. As all $u_i^*(p_{i+1})$ are unique, due to the strict concavity of the criterion function with respect to u_i, it is necessary to increase p_{i+1}, $t \leq i < j$, cf. the monotonicity relations of Propositions 9.2.1 and 9.2.2. This means that at $i = t$ the adjoint relation will change from the second option in (9.30) to the third option, i.e., the relation is the one required for optimality, and it will remain this way during the remaining calculations. Similar argumentation holds for the case of lower tangency, and (9.30) will hold in all cases.

Therefore, when the algorithm stops then an optimal solution is found, since the sufficient optimality conditions of Proposition 3.5.6 are satisfied. □

The bisection method applied in the above implementation need not be the best way to determine p_{s+1}. Linear interpolation may be an attractive alternative. This in particular may be attractive when p_{s+1} is close to the optimal value. If then strict complementarity holds for all the control constraints (9.25), i.e. at the optimal points u_i^* either $\underline{u}_i < u_i < \overline{u}_i$ or the KKT multiplier corresponding to the binding constraint ($\underline{u}_i \leq u_i$ or $u_i \leq \overline{u}_i$) is strictly positive, this will also hold in a neighborhood of u_i^*. Therefore the attempt to get tangency at stage t by linear interpolation in p_{s+1} is essentially the same as solving by linear interpolation the smooth nonlinear equation $f_{t-1}(x_{t-1}, u_{t-1}) = \underline{x}_t$ (or $f_{t-1}(x_{t-1}, u_{t-1}) = \overline{x}_t$), where x_i, u_i, $i = s, \ldots, t-1$ are implicitly given by the variable p_{s+1}. Therefore, if all r_i are twice continuously differentiable, $\nabla^2 r_i^u < 0$ and strict complementarity holds then with linear interpolation the rate of convergence can be expected to be approximately 1.618 for p_{s+1} sufficiently close to p_{s+1}^*(cf. e.g. Luenberger (1989) p. 203). If in addition all r_i are assumed quadratic, then tangency will be obtained by the third value of p_{s+1} sufficiently close to p_{s+1}^*.

Clearly the two ideas could be combined, such that bisection is used during the first iterations, while linear interpolation is used when strict complementarity is assumed to hold.

If the problem is not specified with a given end point as in (9.27), the optimality conditions are modified accordingly. If x_N is free then $p_N^* = \nabla r_N(x_N^*)$ is required. If x_N is bounded as in (9.24) the end conditions for optimality are

$$p_N^* \begin{cases} \leq \nabla r_N(x_N^*) & \text{if} \quad x_N^* = \underline{x}_N \\ = \nabla r_N(x_N^*) & \text{if} \quad \underline{x}_N < x_N^* < \overline{x}_N \\ \geq \nabla r_N(x_N^*) & \text{if} \quad x_N^* = \overline{x}_N \end{cases} \tag{9.46}$$

The algorithm is easily modified accordingly, cf. also the previous section.

In the case that the criterion function is not strictly concave with respect to u_i the algorithm may be applied with the following interpretation. Among the optimal, possibly non-unique, $u_i^*(p_{s+1})$ select those that will not give any violation of the state constraints (9.24); if this is impossible, select those that will give violation at the stage with the largest index j and in addition with the smallest amount of violation (Step 3, Cases 1 through 4 of the algorithm). How this may in fact be done will be illustrated in Section 9.3.

Non-concave Problems

The optimality conditions on which the algorithm is built, Proposition 3.5.6, are sufficient for optimality also if r_i is not concave or if the dynamics is not linear or if the controls are not restricted to an interval. Therefore one may attempt to use the algorithm even if these three assumptions are not fulfilled.

Clearly in this case it cannot be expected that (9.24) hold. Therefore (9.24) should be attempted fulfilled only in an approximate sense in the algorithm. Even if the solution is not found this way, the strategy and trajectory found may be optimal in a modified problem. If this is sufficiently close to the original problem the solution may be meaningful for the original problem. We have:

Proposition 9.2.4 *Assume that r_i is additively separable in x_i and u_i for all i, i.e., $r_i(x_i,u_i) = r_i^x(x_i) + r_i^u(u_i)$ where r_i^x is concave and differentiable. Assume that the dynamic function is additively separable in x_i and u_i and linear in x_i for all i, i.e., $x_{i+1} = F_i^x x_i + f_i^u(u_i)$ rather than (9.23). Assume that the control is constrained as $u_i \in U_i$ where U_i is an arbitrary set, rather than (9.25). Assume that the algorithm constructs a forwards trajectory for this problem, where (9.24) and (9.27) are fulfilled only in an approximate sense. Let the constructed solution be (x^*,u^*), using p^*. Then (x^*,u^*) is an optimal solution to a problem where (9.24) for $i = 1,\dots,N-1$ are substituted by*

$$\begin{aligned} x_i^* \le x_i \le \max\{\overline{x}_i, x_i^*\} \quad &\text{if} \quad \nabla r_i^x(x_i^*) + p_{i+1}F_i^x \le p_i^j \\ \min\{\underline{x}_i, x_i^*\} \le x_i \le \max\{\overline{x}_i, x_i^*\} \quad &\text{if} \quad \nabla r_i^x(x_i^*) + p_{i+1}F_i^x = p_i^j \\ \min\{\underline{x}_i, x_i^*\} \le x_i \le x_i^* \quad &\text{if} \quad \nabla r_i^x(x_i^*) + p_{i+1}F_i^x \ge p_i^j \end{aligned}$$

and (9.27) is replaced by $x_N = x_N^*$.

Proof. The solution found is feasible in the modified problem. The criterion function is concave, and differentiable with respect to x_i. Therefore the stationarity (or in case of active state constraints the indicated adjoint inequalities (9.30)) with respect to x_i and maximization with respect to u_i is sufficient to guarantee that (x^*,u^*) maximizes the criterion, cf. Proposition 3.5.6. Therefore (x^*,u^*) is optimal in the modified problem. □

Small Step Ansgar

The Ansgar algorithm above tries to find p_{s+1} such that $\underline{x}_i \le x_i \le \overline{x}_i$. The indication of *which way* to move p_{s+1} is given by the relation of x_i to $\underline{x}_i$ and $\overline{x}_i$. The indication of *how far* to move p_{s+1} is given by previous values of p_{s+1} and bisection or interpolation ideas. The value of u_i that maximizes the Hamiltonian is recalculated for each new p_{i+1}.

We might also use a small step strategy. In this the indication of the direction to move p_{s+1} is given by the relation of x_i to $\underline{x}_i$ and $\overline{x}_i$ as above. But the indication of how much to move p_{s+1} would be given from an analysis of the maximization of the Hamiltonian.

Example 9.2.1 *Assume u_i and x_i to be scalars. Let $r_i^u(u_i) = -(u_i)^2$, $f_i^u(u_i) = 12u_i$ and $U_i = \{u_i \mid 0 \le u_i \le 2\}$. Then we find the optimal $u_i(p_{i+1})^*$ as*

$$u_i = \begin{cases} 0 & \text{if} \quad p_{i+1} \le 0 \\ p_{i+1}/2 & \text{if} \quad 0 \le p_{i+1} \le 4 \\ 2 & \text{if} \quad 4 \le p_{i+1} \end{cases}$$

Suppose $p_{i+1} = 3$ and we want to change p_{i+1} by δp_{i+1}. Then the change δu_i in $u_i(p^*_{i+1})$ is given as the linear function $\delta u_i = \delta p_{i+1}/2$. We see that this relationship is valid for $-3 \leq \delta p_{i+1} \leq 1$. If we want to e.g. increase p_{i+1} then we should limit the increase to 1 if we want the linear relationship $\delta u_i = \delta p_{i+1}/2$ to be valid. Under the assumptions of Proposition 4.3.mm we may also get linear relationships between p_{j+1} and all other variables for small variations δp_{i+1}. Thus for instance $\delta x_{i+1} = 12\delta u_i = 6\delta p_{i+1}$. The step size can then be established as the maximum stepsize for which a found linear relationship holds, and again the maximum of this value and the value by which tangency is obtained. □

This idea is only implementable if the maximization of the Hamiltonian is easily analytically tractable, in particular if r^u_i is linear or quadratic, f^u_i linear and U_i given by linear relations, e.g. as in (9.22) - (9.27).

The advantage of this method relative to the bisection and interpolation methods described above is that the maximization of the Hamiltonian may be easier since it is not done from scratch in every iteration, but only the first time. The disadvantage is that the steps taken may be small, even far from the tangency value. It is clear that the two stepsize strategies may be advantageously combined.

Now consider the problem, where r^u_i is concave and quadratic. For this case we may give the following small step Ansgar algorithm.

Algorithm

Step 0 Let the start index $s = 0$. Let $x_s = \underline{x}_0$. Choose p_{s+1} arbitrarily.

Step 1 Calculate recursively forwards from $i = s$:

- $u^*_i(p_{i+1}) = \text{argmax}[c_i + p_{i+1}F^u_i u_i]$ subject to (9.25). In case of non-unique solution, choose an arbitrary optimal $u^*_i(p_{i+1})$.
- x_{i+1} from the dynamic equation (9.23).
- p_{i+1} from the middle option in the adjoint equation (9.30).

until for some index v one of the following situations occurs

Case 1 $\underline{x}_i < x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, and $x_v > \overline{x}_v$. Let $\delta p^{nom}_{s+1} = -1$. Go to Step 2.

Case 2 $\underline{x}_i \leq x_i < \overline{x}_i$, $i = s+1, \ldots, v-1$, and $x_v < \underline{x}_v$. Let $\delta p^{nom}_{s+1} = 1$. Go to Step 2.

Case 3 Lower tangency: $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, $x_t = \underline{x}_t$ for some t, $s+1 \leq t \leq v-1$, and $x_v > \overline{x}_v$. Go to Step 3.

Case 4 Upper tangency: $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, $x_t = \overline{x}_t$ for some t, $s+1 \leq t \leq v-1$, and $x_v < \underline{x}_v$. Go to Step 4.

Case 5 $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = 1, \ldots, N-1$, $x_N = \underline{x}_N$: Store the optimal strategy and trajectory from stage s to stage N. Stop.

Step 2 Change p_i, u_i and x_i:

- Calculate δp^{nom}_i, $i = s+2, \ldots, v$ from the middle option in (9.30).
- Calculate $\delta u^{nom}_i(\delta p^{nom}_{i+1})$, for those i, $i = s, \ldots, v-1$, for which $\underline{u}_i < u_i < \overline{u}_i$. Find the minimal fraction α of the calculated δp^{nom}_i such that $u_i + \delta u^{nom}_i(\delta p^{nom}_{i+1})$ reaches a bound (9.25).

- Calculate δx_i^{nom}, $i = s+1,\ldots,v$ resulting from the changes $\delta u_i^{nom}(\delta p_{i+1}^{nom})$ using (9.23). Find the minimal fraction β of the calculated δp_i^{nom} such that $x_i + \delta x_i^{nom}$ reaches a bound (9.24).
- Find the minimal fraction γ of the calculated δp_i^{nom} such that a $u_i^*(p_{i+1})$, $i = s,\ldots,v-1$, which is at a bound (9.25), will leave this bound.
- Let $\epsilon = \min\{\alpha,\beta,\gamma\}$.
- Change p_i by $\epsilon\delta p_i^{nom}$, $i = s+1,\ldots,v$ and calculate a forwards trajectory according to these new p_i.
- Go to Step 1.

Step 3 (Lower tangency): Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \underline{x}_s$. Go to Step 1.

Step 4 (Upper tangency): Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \overline{x}_s$. Go to Step 1.

Now assume the problem has a criterion function which is piecewise quadratic in u_i; this may for instance be the result of a reformulation of a problem with $m > 1$, see Section 4.6. Suppose that r_i^u consists of K segments, each segment an interval with bounds $\underline{u}_k$. Then it may be specified as follows, $k = 1,\ldots,K-1$,

$$r_i^u(u_i) = d_i^k + c_i^k u_i + b_i^k (u_i)^2, \quad \text{if} \quad \underline{u}_k \le u_i \le \underline{u}_{k+1} \tag{9.47}$$

Here, the d_i^k's, c_i^k's and b_i^k's have values such that r_i^u is strictly concave (and hence continuous).

The algorithm is adapted to this case by considering, during the iterations, only one segment at a time (Steps 2 and 3 of the algorithm).

Proposition 9.2.5 *Consider the Ansgar algorithm for the quadratic linear problem (9.22) - (9.27) where $n = 1$, $F_i^x > 0$ and where r_i^u is concave, piecewise quadratic and consists of at most K segments. The computational complexity of the algorithm is $O(N^2K)$.*

Proof. Consider a subsequence of iterations where all δp_i^{nom}'s have the same sign (Step 1, Case 1, Case 2). The index v will in this subsequence attain a maximal value $\overline{v}$. During these iterations calculations will not be performed on stages outside the interval $i = s,\ldots,v$. The number of arithmetic operations will for each stage be proportional to K in order to find u_i^* (Step 1) (however, in all the remaining iteration it is known which segment is relevant, and the number of arithmetic operations will then be proportional to the number of stages). For this part the number of arithmetic operations will therefore in total be proportional to at most $(\overline{v} - s)K$.

In the remaining part of the iterations on this subsequence the arithmetic operations per iteration is at most $(\overline{v} - s)$. The number of iterations in the subsequence will be at most $(\overline{v} - s)K$ since then all segments of all r_i^u have been searched (and none are repeated since δp_i^{nom} has the same sign). Thus, for such subsequence of iterations the number of arithmetic operations is proportional to at most $(\overline{v} - s)^2K$.

Let $\overline{v}^1$ be the $\overline{v}$ relative to subsequence 1, let $\overline{v}^2$ be the $\overline{v}$ relative to subsequence 2 etc., with $\overline{v}^a$ being the last one (equal to N); let $\overline{v}^0 = 0$. Considering all subsequences, the number of arithmetic operations will be proportional to at most $\sum_{j=0}^{a-1}(\overline{v}^{j+1} - \overline{v}^j)^2K$. This expression is convex and the $\overline{v}^j$'s may attain values in a convex set (except for the integer requirement). The maximal value is therefore seen to be N^2K, attained at an extreme point. Hence, the computational complexity of

the algorithm is $O(N^2K)$. □

We observe that the computational complexity of the algorithm is the same as for the DP algorithm of Section 4.6. This result is better than the result reported in Florian, Lenstra and Kan (1980).

9.3 The Linear Problem

Now we consider the linear problem

$$\max[\sum_{i=0}^{N-1} R_i^x x_i + R_i^u u_i + R_N^x x_N] \tag{9.48}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{9.49}$$

$$G_i^u u_i - \overline{g}_i \leq 0 \tag{9.50}$$

$$H_i^u u_i - \overline{h}_i = 0 \tag{9.51}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{9.52}$$

$$x_0 = \underline{x}_0 \tag{9.53}$$

We specify the algorithm as a small step algorithm, cf. the previous section. As before, p and (x, u) are moved such that optimality in a truncated problem starting at stage 0 is maintained all the time. We shall show that the algorithm is closely related to linear programming duality and parametric programming.

For the linear problem (9.48) - (9.53) the adjoint relations will take the form

$$(R_i^x + p_{i+1}F_i^x - p_i)^j \begin{cases} \leq 0 & \text{if} \quad (\underline{x}_i)^j = (x_i^*)^j \\ = 0 & \text{if} \quad (\underline{x}_i)^j < (x_i^*)^j < (\overline{x}_i)^j \\ \geq 0 & \text{if} \quad (x_i^*)^j = (\overline{x}_i)^j \end{cases} \tag{9.54}$$

n=1

First consider the case with $n = 1$ where we assume that control constraints (9.50) - (9.51) are given as simple bounds (9.25). The following Ansgar algorithm is close to the one for quadratic criterion problems from the previous section. The main difference is that for the linear problems we must carefully handle the situations with non-unique solutions to the maximization of the Hamiltonian (Step 3).

Algorithm

Step 0 Let the start index $s = 0$. Let $x_s = \underline{x}_0$. Choose p_{s+1} arbitrarily.

Step 1 Calculate recursively forwards from $i = s$:

- $u_i^+(p_{i+1}) = \text{argmax}[R_i^u + p_{i+1}F_i^u u_i]$ subject to (9.25). In case of non-unique solution, choose an arbitrary optimal $u_i^+(p_{i+1})$.
- x_{i+1} from the dynamic equation (9.49).
- p_{i+1} from the adjoint relation (9.54).

until for some index v one of the following situations occurs

Case 1 $\underline{x}_i < x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, and $x_v > \overline{x}_v$. Let $\delta p_{s+1}^{nom} = -1$. Go to Step 2.

Case 2 $\underline{x}_i \leq x_i < \overline{x}_i$, $i = s+1, \ldots, v-1$, and $x_v < \underline{x}_v$. Let $\delta p_{s+1}^{nom} = 1$. Go to Step 2.

Case 3 Lower tangency: $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, $x_t = \underline{x}_t$ for some t, $s+1 \leq t \leq v-1$, and $x_v > \overline{x}_v$. Go to Step 4.

Case 4 Upper tangency: $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = s+1, \ldots, v-1$, $x_t = \overline{x}_t$ for some t, $s+1 \leq t \leq v-1$, and $x_v < \underline{x}_v$. Go to Step 5.

Case 5 $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = 1, \ldots, N-1$, $x_N = \underline{x}_N$: Store the optimal strategy and trajectory from stage s to stage N. Stop.

Step 2 (Violation without prior tangency): Change p_i:

- Calculate δp_i^{nom}, $i = s+2, \ldots, v$ from the middle option in the adjoint relation (9.54).
- Find the minimal fraction ϵ of the calculated δp_i^{nom} such that $R_i^u + (p_{i+1} + \epsilon \delta p_{i+1}^{nom}) F_i^u = 0$ for at least one index i, $s \leq i \leq v-1$; stages i for which $u_i = \underline{u}_i$ (if $\delta p_{s+1}^{nom} = -1$ and $F_i^u > 0$ or $\delta p_{s+1}^{nom} = 1$ and $F_i^u < 0$) or $u_i = \overline{u}_i$ (if $\delta p_{s+1}^{nom} = 1$ and $F_i^u > 0$ or $\delta p_{s+1}^{nom} = -1$ and $F_i^u < 0$) are excluded. I.e. $\epsilon = \min_i \{\max\{0, (-(F_i^u)^{-1} R_i^u - p_{i+1})/\delta p_i^{nom}\}\}$ over the relevant i. Let i^ϵ be the index at which ϵ is minimum. In case of several indexes, choose i^ϵ among those that have $\underline{u}_i < u_i < \overline{u}_i$, if possible.
- Change p_i by $\epsilon \delta p_i^{nom}$, $i = s+1, \ldots, v$.
- Go to Step 3.

Step 3 Change u_{i^ϵ} and x_i:

- Find the maximal (if $\delta p_i^{nom} = 1$) or minimal (if $\delta p_i^{nom} = -1$) change δu_{i^ϵ} such that the following hold when a forwards trajectory, $i = i^\epsilon + 1, \ldots, x_v$, using $u_{i^\epsilon} + \delta u_{i^\epsilon}$ (all other u_i's unchanged) is used:
  - $\underline{u}_{i^\epsilon} \leq u_{i^\epsilon} + \delta u_{i^\epsilon} \leq \overline{u}_{i^\epsilon}$
  - $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = i^\epsilon + 1, \ldots, v-1$.
  - $x_v \geq \overline{x}_v$ (if $\delta p_{s+1}^{nom} = -1$) or $x_v \leq \underline{x}_v$ (if $\delta p_{s+1}^{nom} = 1$).
- Change u_{i^ϵ} by δu_{i^ϵ}, calculate the resulting x_i, $i = i^\epsilon + 1, \ldots, v$.
- Go to Step 1.

Step 4 (Lower tangency): Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \underline{x}_s$. Go to Step 1.

Step 5 (Upper tangency): Store the optimal strategy and trajectory from stage s to stage t. Let $s = t$ and $x_s = \overline{x}_s$. Go to Step 1.

Now assume the problem has a criterion function which is piecewise linear in u_i. Suppose that r_i^u consists of K segments, each segment an interval with bounds $\underline{u}_k$. Then it may be specified as follows, $k = 1, \ldots, K-1$,

$$r_i^u(u_i) = d_i^k + c_i^k u_i, \quad \text{if} \quad \underline{u}_k \leq u_i \leq \underline{u}_{k+1} \tag{9.55}$$

Here, the d_i^k's and c_i^k's have values such that r_i^u is concave (and hence also continuous).

The algorithm is adapted to this case by considering, during the iterations, only one segment at a time (Steps 2 and 3 of the algorithm).

Proposition 9.3.1 *Consider the Ansgar algorithm for the linear problem (9.48) - (9.49), (9.25), (9.52) - (9.53) where $F_i^x > 0$ and where r_i^u is concave, piecewise linear and consists of at most K segments. The computational complexity of the algorithm is $O(N^2K)$.*

Proof. The proof is essentially the same as in the previous Proposition 9.2.5. □

$\mathbf{n \geq 1}$

Now consider the case of $n \geq 1$. For this, we consider first the general LP problem

$$\max[c'u] \tag{9.56}$$
$$Bu \leq b \tag{9.57}$$
$$Du \leq d \tag{9.58}$$

where $c \in R^m$, $u \in R^m$, $b \in R^n$ and B, D and d are matrices of appropriate dimensions. We assume that $\{u \mid Du \leq d\}$ is nonempty and compact and that the problem has a feasible and hence optimal solution.

We introduce the row vector multipliers $p \in R^n$, $p \geq 0$, relative to the first set of constraints (9.57) and μ, $\mu \geq 0$, of appropriate dimensions to the second one (9.58). Consider a $p^o \geq 0$ and the relaxed problem

$$\max[c'u - p^oBu] \tag{9.59}$$
$$Du \leq d \tag{9.60}$$

Let u^o be a solution to this.

Define $CSC(k, p^o, u^o)$ as the fulfillment of the following complementary slackness conditions for a k, $1 \leq k \leq n+1$:

$$(Bu^o)^j \leq b^j,\ (p^o)^j(Bu^o - b)^j = 0,\ (p^o)^j \geq 0,\ 1 \leq j \leq k-1 \tag{9.61}$$
$$(p^o)^j = 0,\ k+1 \leq j \leq n \tag{9.62}$$
$$Du^o \leq d,\ \mu(Du^o - d) = 0,\ \mu \geq 0 \tag{9.63}$$

We shall consider the largest index $k \leq n+1$ such that $CSC(k,.,.)$ holds and, if $k \leq n$, $(Bu^o)^k > b^k$. If $k = n+1$ then u^o solves the problem (9.56) - (9.58), since (9.59) - (9.63) are the sufficient optimality conditions from Lagrangian relaxation. If $k \leq n$ then there are problems with index k. In the following we assume $k \leq n$ and show how to adjust p, or to select another optimal u^o, in order to solve (9.56) - (9.58) by solving the relaxations (9.59) - (9.60), and hence to increase k.

Define the dual function

$$D(p^o) = c'u^o - p^o(Bu^o - b) \tag{9.64}$$

Lemma 1 *Assume $p^o \geq 0$ and that u^o is any solution to (9.59) - (9.60). Assume that $k \leq n$, that $CSC(k, p^o, u^o)$ holds and that $(Bu^o)^k > b^k$. Then either there is another solution u^+ to (9.59) - (9.60) with $(Bu^+)^k < (Bu^o)^k$ and $CSC(k, p^o, u^+)$ fulfilled or there is another p^+ such that (i) u^o solves (9.59) - (9.60) also with $p = p^+$, (ii) $CSC(k, p^+, u^o)$ holds and (iii) $D(p^+) < D(p^o)$.*

Proof. If there is a u^+ such that $(Bu^+)^k < (Bu^o)^k$ and the $CSC(k, p^o, u^+)$ is fulfilled then we are done. Therefore assume this is not the case. Consider the problem

$$\max[-(Bu)^k] \tag{9.65}$$

$$\begin{aligned} Du &\leq d \\ (Bu)^j &\leq b^j,\ 1 \leq j \leq k-1 \\ p^j(Bu-b)^j &= 0,\ 1 \leq j \leq k-1 \\ -c'u + p^oBu &\leq -c'u^o + p^oBu^o \end{aligned} \tag{9.66}$$

This problem expresses the objective of finding the u that yields the smallest $(Bu)^k$ among those u that are optimal in (9.59) - (9.60). By assumption u^o is a solution to the problem (9.59) - (9.60).

The KKT conditions corresponding to the solution of (9.65) - (9.66) can be written

$$0 = -B^k - \mu D - \sum_{j=1}^{k-1} \lambda^j B^j - \lambda^k(-c + p^oB) \tag{9.67}$$

$$CSC(k, p^o, u^o)$$

$$\lambda_k \geq 0,\ \lambda^k(-c'u^o + p^oBu^o + c'u^o - p^oB) = 0 \tag{9.68}$$

In the problem (9.65) - (9.66) the last restriction is active and $\lambda^k > 0$ since by assumption no solution in (9.59) - (9.60) had $(Bu^+)^k < (Bu^o)^k$. Therefore the condition (9.67) can be rewritten to

$$0 = -(1/\lambda^k)B^k - (\mu/\lambda^k)D - \sum_{j=1}^{k-1}(\lambda^j/\lambda^k)B^j - (-c + p^oB)$$

Defining $p^+ = p^o + (\lambda^1/\lambda^k, \lambda^2/\lambda^k, \ldots, 1/\lambda^k, 0, \ldots, 0)$ we see that the optimality conditions of the problem (9.67) - (9.68) are the optimality conditions of

$$\max[c'u - p^+Bu]$$

$$Du \leq d$$

Therefore u^o is maximizing in this problem also.

Now consider $D(p^+) - D(p^o) = (c'u^o - p^+(Bu^o - b)) - (c'u^o - p^o(Bu^o - b))$. Since $CSC(k, p^o, u^o)$ and $CSC(k, p^+, u^o)$ are satisfied all terms except those corresponding to index k are zero or cancel. Since $(p^+ - p^o)^k = 1/\lambda^k > 0$ and $(Bu^o - b)^k > 0$ we have $D(p^+) - D(p^o) = (-p^+ + p^o)^k(Bu^o - b)^k < 0$, i.e., $D(p^+) < D(p^o)$. □

To apply the result first consider the case without state constraints (9.52). We observe that if we start with $p = 0$ and solve (9.59) - (9.60) then $CSC(1, p, u)$ holds. Thus, let $k = 1$. Then we attempt to get $(Bu)^k \leq 0$ while $CSC(k, ., .)$ remains fulfilled. This we do either by finding another solution to (9.59) - (9.60) (if u were not unique) or by increasing p^k while p^j, $1 \leq j \leq k-1$, are adjusted to maintain u optimal. This process will terminate in a finite number of steps provided no cycling (in the simplex algorithm sense, see e.g. Luenberger (1989)) takes place. When terminated we increment k by one. Since there are only n values for the index k we have now shown

Lemma 2 *Assume an optimal solution exists and that cycling does not take place. Then the described algorithm will find an optimal solution in a finite number of steps.*

If the first set of constraints (9.57) were given as equalities, i.e., $Bu = b$, then this problem is also solved, the only difference being that it is not required that $p \geq 0$.

An aspect which is essential for implementation is that throughout the process we maintain optimality of u with respect to the p currently used. We can therefore describe the process as *parametric programming in an optimal simplex tableau.* Changing p corresponds to parametric

programming of the cost coefficients. During this, p^k is increased while p^j, $1 \leq j \leq k-1$, are changed so that u remains optimal and $CSC(k,.,.)$ holds. Changing u corresponds to a parametric programming of the right hand side. During this, u is changed so that $(Bu)^k$ decreases while at the same time u remains optimal and $CSC(k,.,.)$ holds.

We now apply this to the linear control problem with restricted end point $x_N \leq \underline{x}_N$ and no intermediate state constraints (9.52). We consider the problem similar to (9.59) - (9.60), which is the problem of maximization of the Hamiltonian:

$$\max_{u_i}[R_i^u u_i + p_{i+1} F_i^u u_i] \tag{9.69}$$

$$G_i^u u_i \leq \overline{g}_i \tag{9.70}$$

$$H_i^u u_i = \overline{h}_i \tag{9.71}$$

A given p_N permits the calculation of p_i recursively backwards according to the adjoint equation. The Hamiltonians are then maximized with respect to u_i and finally x_i are calculated recursively forwards. Knowing then x_N we can change p_N according to the description above.

For this problem we therefore get the following *Primal-Dual Maximum Principle Algorithm for Linear Problems*:

Algorithm

Step 0 Let $p_N = 0$. Calculate p_i recursively backwards from the adjoint equations. Calculate the u_i that maximize the Hamiltonians. Calculate x_{i+1} recursively forwards to obtain x_N. Let $k = 1$.

Step 1 If $k = n+1$ then stop. Else find for all i the u_i that maximize the Hamiltonians.

Step 2 Find the subset $\{u\}$ of u from Step 1 that is closest to make $x_N^k \leq \underline{x}_N^k$ while the $CSC(k,.,.)$ hold with respect to u (and with the corresponding x_{i+1} calculated recursively forwards).

Step 3 If $CSC(k+1,.,.)$ holds then let $k = k+1$ and go to Step 1. Else go to Step 4.

Step 4 For a nominal increase of $(\delta p_N^{nom})^k = 1$ in p_N^k find the changes $(\delta p_N^{nom})^j$ in p_N^j, $j < k$, necessary to maintain optimality of $\{u\}$ and to maintain the $CSC(k,.,.)$ fulfilled with respect to p. All p_i are calculated recursively backwards from the adjoint equations. For $k < j$ all $(\delta p_N^{nom})^j$ are zero.

Step 5 Find the maximal fraction δ^* of $(\delta p_N^{nom})^k$ such that $\{u\}$ (from Step 2) remains optimal.

Step 6 Change p_N by the amount $\delta^*((\delta p_N^{nom})^1, (\delta p_N^{nom})^2, \ldots, (\delta p_N^{nom})^k, 0, \ldots, 0)$. Change p_i recursively backwards according to the adjoint equations and go to Step 1.

Proposition 9.3.2 *Assume that an optimal solution to the linear OCP (9.48) - (9.51), (9.53) exists and that cycling does not occur. Then the algorithm finds an optimal solution in a finite number of steps.*

Proof. The algorithm implements the process described in connection with the above Lemma 2 and the result therefore follows from this. □

State Constraints

If we have intermediate state constraints $\underline{x}_i \leq x_i \leq \overline{x}_i$, (9.52), then the algorithm may be modified to handle this. It can be done in accordance with the forwards idea as follows. We add a loop around steps 1 - 5 in the above algorithm. Initially we solve a one stage truncated problem (i.e. $N = 1$) by the inner loop in steps 1 - 5. When this is solved the final index is incremented from 1 to 2. Now the problem is again solved by the steps 1 - 5 *maintaining optimality and feasibility with respect to* $\underline{x}_i \leq x_i \leq \overline{x}_i$. When this is done the final index is again incremented by one. In this way it continues until the problem is solved with the final index is equal to the original N.

In order to maintain feasibility and optimality we will have to elaborate on the algorithm. We consider state constraints (9.52) and therefore also must take the adjoint relations (9.54) into account.

We therefore introduce the following modifications in the above algorithm relative to $\underline{x}_i \leq x_i$ (similar considerations hold for $x_i \leq \overline{x}_i$):

- Suppose $p_i^j < (R_i^x + F_i^x p_{i+1})^j$ (and therefore $x_i^j = \overline{x}_i^j$ by the CSC). Then the changes are
  - If p_{i+1}^j is decreased then it should not be decreased more than to make $p_i^j < (R_i^x + p_{i+1}F_i^x)^j$ (Step 6).
  - When u is changed (Step 1) is should be in such a way as to maintain $x_i^j = \overline{x}_i^j$ (this is part of the CSC condition)
- Suppose $x_i^j < \underline{x}_i^j$ (and therefore $p_i^j = (R_i^x + p_{i+1}F_i^x)^j$ by the CSC). Then the changes are
  - When u is changed it should be in such a way as to maintain $x_i^j \leq \overline{x}_i^j$ (Step 5).
  - When p_{i+1}^j is changed p_i^j should change such that $p_i^j = (R_i^x + p_{i+1}F_i^x)^j$ (Step 6).

The change δu can be found stagewise. However, often the number of u_i^j that are not uniquely given by maximization of the Hamiltonian is very small compared to $N(n+m)$. Often it will be one more than the number of active state constraints. A direct application of the stagewise solution technique may therefore not be the best. We illustrate this by an example.

Example 9.3.1 *Assume* $n = 2$, $m = 2$, $x_N^1 = \underline{x}_N^1$, $x_N^2 \neq \underline{x}_N^2$ *(but required to hold),* $x_t^t = \underline{x}_t^t$ *and binding. Assume the dynamics given by*

$$x_{i+1} = \begin{pmatrix} x_i^1 + a_i^1 u_i^1 + a_i^2 u_i^2 \\ x_i^2 + a_i^3 u_i^1 \end{pmatrix}$$

where a_i^j *are scalars. Assume that all* u_i *maximizing the Hamiltonians are unique except for* u_0^1 *and* u_s^2 *where* $0 < s < t < N$. *The relationship between* δu_0^1, δu_s^2, δx_s^1 *and* δx_N^2 *is seen to be*

$$\begin{aligned} a_0^1 \delta u_0^1 + a_s^2 \delta u_s^2 &= \delta x_t^1 \\ a_0^3 \delta u_0^1 &= \delta x_N^2 \end{aligned}$$

With $\delta x_0^1 = 0$ *and* $\delta x_N^2 = \underline{x}_N^2 - x_N^2$ *we solve this system very easily.*

If we solve it recursively, we would have to involve the $2N$ *variables* δx_i *in addition to* δu_0^1 *and* δu_s^2. □

Finally observe that the algorithm can not be seen as operating completely in a stagewise manner. A key point in the algorithm is the appropriate selection of a particular value among those u_i^o that are solutions to the maximization of the Hamiltonian; the problem is singular, cf. page 111, and often the solution is nonunique. u_i^o must be selected such that the state constraints are fulfilled (Step 2), if possible. In general, the state constraints in question are located at stage t and the control u_i^o at stage i where $i < t$, but often $i + 1 < t$ and therefore the myopic stagewise approach is not operational. This is a consequence of the solution structure of the linear problem, rather than the particular algorithm presented here, as witnessed by e.g. the algorithms for linear problems presented in Propoi (1981).

9.4 Planning Horizons

Decision and forecast horizons - jointly referred to as planning horizons - may be characterized as follows. If the solution u_i^* for $i = 0, \ldots, t_1$ is independent of data (r_i, f_i, V_i) for periods later than t_2 (where $t_1 < t_2$) then t_1 is a decision horizon and t_2 is a forecast horizon.

Decision and forecast horizons may in more abstract terms be characterized as follows. Consider a truncated problem ending at x_{i+1}. Let $u_i^*(x_{i+1})$ denote the optimal control relative to x_{i+1}, i.e., u_i^* is optimal for the truncated problem ending at x_{i+1}. Similarly, $x_i^*(x_{i+1})$ denotes the optimal predecessor to x_{i+1}. This may be used recursively, defining $u_s^*(u_{s+1}^*(\ldots(u_{k-1}^*(x_k))\ldots))$ or $u_s^*(x_k)$ for short, $s < k$. Similarly for $x_{s+1}^*(x_k)$. We then have the following planning horizon result: If $u_s^*(x_k) = u_s^o$ or $x_{s+1}^*(x_k) = x_{s+1}^o$, for all $x_k \in Y$ (i.e. the same u_s or x_{s+1} is optimal irrespective of which $x_k \in Y_k$ is relevant) then k is a forecast horizon and s is a decision horizon.

In the previous sections 9.2 - 9.3 we have used the detection of decision horizons as part of the algorithms, thus breaking the solution of the optimal control problem into the solution of a sequence of smaller problems.

The existence of decision and forecast horizons is fortunate also from the point of view of modeling planning problems. Often the data for the more distant future is uncertain and if a forecast horizon shorter than N exists then some of the effect of uncertainty is eliminated. Thus, in this situation a forwards approach may be appropriate, while often backwards dynamic programming is preferred for problems with uncertainties.

In this section we derive conditions under which planning horizons may be identified.

We consider the following problem

$$\max[\sum_{i=0}^{N-1} r_i^x(x_i) + r_i^u(u_i) + r_N(x_N)] \tag{9.72}$$

$$x_{i+1} = F_i^x x_i + F_i^u u_i + \overline{f}_i \tag{9.73}$$

$$\underline{x}_i \leq x_i \leq \overline{x}_i \tag{9.74}$$

$$u_i \in U_i \tag{9.75}$$

$$x_0 = \underline{x}_0 \tag{9.76}$$

All functions r_i^x are assumed continuously differentiable (although generalization to concave nonsmooth r_i^x is straightforward). The criterion function (9.72) is concave, the dynamic function (9.73) is linear and the local control constraint set U_i in (9.75) is convex and so are the state constraints (9.74).

From the previous sections it appears that monotonicity relations are fundamental for the forwards algorithms. In particular the relationships between movements in p_{i+1} and the movements

in the associated x_{i+1} (via u_i and the dynamic equation) and p_{i+2} (via the adjoint relations). For $n = 1$ this was investigated in Section 9.2, see e.g. Proposition 9.2.1; here we treat the situation with $n \geq 1$. We may indicate the idea for the analysis as follows.

In the construction of the forwards trajectories the control u_i^* is found by maximization of the Hamiltonian

$$\max_{u_i}[r_i^u(u_i) + p_{i+1}F_i^u u_i] \tag{9.77}$$

subject to (9.75). Observe that x_i does not enter here due to the additive separability between x_i and u_i. x_{i+1} is found by the dynamic equation (9.73) and a forwards trajectory is considered only as far as x_{i+1} fulfills (9.74).

The states are linked by the dynamic equation (9.73) and the costate variables are linked by the adjoint relations

$$(\nabla r_i^x(x_i) + p_{i+1}F_i^x - p_i)^j \begin{cases} \leq 0 & \text{if} \quad (\underline{x}_i)^j = (x_i^*)^j \\ = 0 & \text{if} \quad (\underline{x}_i)^j < (x_i^*)^j < (\overline{x}_i)^j \\ \geq 0 & \text{if} \quad (x_i^*)^j = (\overline{x}_i)^j \end{cases} \tag{9.78}$$

Thus, a forwards trajectory up to stage t is feasible, and if a truncated problem covering periods $i = 0, \ldots, t$ ending at x_t is considered, this trajectory would be optimal for this particular x_t.

Consider first the case where the $u_i^*(x_i, p_{i+1})$ that maximize in (9.77) for given x_i and p_{i+1} is unique, hence also $x_{i+1}^*(x_i, p_{i+1}) = f_i(x_i, u_i^*(x_i, p_{i+1}))$ is unique. We assume that for any fixed x_i and any p_{i+1}, any j and any $k \in \{1, \ldots, n\}$, $k \neq j$, $(x_{i+1}^*(x_i, p_{i+1}))^j$ is non-decreasing and $(x_{i+1}^*(x_i, p_{i+1}))^k$ is non-increasing as a function of p_{i+1}^j, i.e.,

$$(x_{i+1}^*(x_i, p_{i+1}^1, \ldots, p_{i+1}^j, \ldots, p_{i+1}^n))^j \leq \tag{9.79}$$
$$(x_{i+1}^*(x_i, p_{i+1}^1, \ldots, p_{i+1}^j + \epsilon, \ldots, p_{i+1}^n))^j$$
$$(x_{i+1}^*(x_i, p_{i+1}^1, \ldots, p_{i+1}^j, \ldots, p_{i+1}^n))^k \geq \tag{9.80}$$
$$(x_{i+1}^*(x_i, p_{i+1}^1, \ldots, p_{i+1}^j + \epsilon, \ldots, p_{i+1}^n))^k$$

for any $\epsilon \geq 0$. (Condition (9.79) is actually not an assumption but a characteristics, cf. the first part of Proposition 9.2.1 which is easily seen to imply (9.79).)

The desired implication of this assumption is the property that if p_{i+1}^j is increased (decreased) then p_{i+1}^k need not be decreased (increased) in order for $(x_{i+1}^*(x_i, p_{i+1}))^k$ to remain fixed for some or all $k \neq j$.

For the case of a non-unique solution to (9.77) we shall by $(x_{i+1}^*(x_i, p_{i+1}))$ in (9.79) - (9.80) mean $f_i(x_i, u_i^*)$ for a particular value u_i^* maximizing in (9.77) for given (x_i, p_{i+1}). For this more general situation we use the above property directly as assumption, i.e., we assume that if p_{i+1}^j is increased (decreased) then p_{i+1}^k need not be decreased (increased) in order for $(x_{i+1}^*(x_i, p_{i+1}))^k$ to remain fixed for some or all $k \neq j$.

Thus we assume that for any x_i, p_{i+1} and $\epsilon \geq 0$ and any $j \in \{1, \ldots, n\}$, any $k \in \{1, \ldots, n\}$, $k \neq j$, there are $\delta^k \geq 0$, such that

$$(x_{i+1}^*(x_i, p_{i+1}^1 + \delta^1, \ldots, p_{i+1}^j + \epsilon, \ldots, p_{i+1}^n + \delta^n))^k = \tag{9.81}$$
$$(x_{i+1}^*(x_i, p_{i+1}^1, \ldots, p_{i+1}^j, \ldots, p_{i+1}^n))^k$$

We refer to (9.81) as assumption A^+. For smooth concave $ub_i^{i+1}(x_i, .)$ the assumption is equivalent to the condition that $\partial ub_i^{i+1}(x_i, x_{i+1})/\partial x_{i+1}^k$ is non-increasing as a function of x_{i+1}^j, $j \neq k$. For

concave twice continuously differentiable $ub_i^{i+1}(x_i,.)$ it is equivalent to $\nabla^2_{x_{i+1}^j x_{i+1}^k} ub_i^{i+1}(x_i,x_{i+1}) \leq 0$, $j \neq k$. For non-smooth $ub_i^{i+1}(x_i,.)$ it is equivalent to $D(ub_i^{i+1}(x_i,x_{i+1};s))$ being non-increasing and $D(ub_i^{i+1}(x_i,x_{i+1};-s))$ being non-decreasing as a function of x_{i+1}^j where $D(ub_i^{i+1}(x_i,x_{i+1};s))$ is the directional derivative of $ub_i^{i+1}(x_i,x_{i+1})$ in the direction $s=(0,\ldots,0,1,0,\ldots,0)' \in R^n$ (where the single 1 is at the k-th position, $j \neq k$).

Example 9.4.1 *As an example we take the following case which may be seen as a version of the hydropower scheduling problem of Example 2 on page 16. We assume $n=2$, $m=2$ and the dynamics are given as*

$$\begin{pmatrix} x_{i+1}^1 \\ x_{i+1}^2 \end{pmatrix} = \begin{pmatrix} x_i^1 \\ x_i^2 \end{pmatrix} + \begin{pmatrix} -u_i^1 \\ u_i^1 \end{pmatrix} + \begin{pmatrix} 0 \\ -u_i^2 \end{pmatrix} + \begin{pmatrix} d_i^1 \\ d_i^2 \end{pmatrix}$$

As seen, u_i^1 is the water taken from reservoir 1 to reservoir 2 during period i; u_i^2 takes water from reservoir 2, this water is then lost. d_i^j is the natural inflow to reservoir j during period i, and x_i^j is the content of reservoir j at the beginning of period i.

The criterion function may be given as

$$\max[\sum_{i=0}^{51} -(u_i^1)^2 - (u_i^2)^2 + c_i(u_i^1+u_i^2) - (x_{52}^1)^2 - (x_{52}^2)^2 + c_N(x_{52}^1+x_{52}^2)]$$

expressing the desire to maximize the income from sale of hydropower produced via u_i in each of the 52 weeks of the year plus the value of the final storage volume x_N.

Local constraints may be given as (9.74) while (9.75) is specified as

$$\underline{u}_i \leq u_i \leq \overline{u}_i$$

It is readily verified that if the local constraints (9.74) - (9.75) are disregarded, then we find the optimal control from (9.77) as

$$\begin{aligned} u_i^1 &= \tfrac{1}{2}(c_i - p_{i+1}^1 + p_{i+1}^2) \\ u_i^2 &= \tfrac{1}{2}(c_1 - p_{i+1}^2) \end{aligned}$$

and consequently from (9.73)

$$\begin{aligned} x_{i+1}^1 &= x_i^1 + \tfrac{1}{2}p_{i+1}^1 - \tfrac{1}{2}p_{i+1}^2 + d_i^1 - \tfrac{1}{2}c_i \\ x_{i+1}^2 &= x_i^2 - \tfrac{1}{2}p_{i+1}^1 + p_{i+1}^2 + d_i^2 \end{aligned}$$

Thus, if we increase (decrease) p_{i+1}^1, we increase (decrease) x_{i+1}^1 and decrease (increase) x_{i+1}^2. In order to keep x_{i+1}^2 fixed we must therefore also increase (decrease) p_{i+1}^2. Similarly, if we increase (decrease) p_{i+1}^2, we increase (decrease) x_{i+1}^2 and decrease (increase) x_{i+1}^1 and in order to keep x_{i+1}^1 fixed we must therefore also increase (decrease) p_{i+1}^1. Thus, the condition (9.81) is verified.

It may be further verified that $\nabla_{x_{i+1}} ub_i^{i+1} = (4x_i^1 - 4x_{i+1}^1 + 2x_i^2 - 2x_{i+1}^2 + 4d_i^1 + 2d_i^2 - 2c_1, 2x_i^1 - 2x_{i+1}^1 + 2x_i^2 - 2x_{i+1}^2 + 2d_i^1 + 2d_i^2 - c_1)$ such that $\nabla_{x_{i+1}^j} ub_i^{i+1}$ is decreasing with increasing x_{i+1}^k, $j \neq k$. Further, $\nabla^2_{x_{i+1}^j x_{i+1}^k} ub_i^{i+1} = -2$, $j \neq k$, i.e., negative. □

By a forwards algorithm we shall in the following understand an algorithm which is based on the construction of forwards trajectories which are optimal up to a certain stage t, for the particular state x_t reached. In particular for $n > 1$ the algorithm may work with such forwards trajectories for $i = 0, \ldots, t$ and eliminate any state infeasibilities at $i = t+1$ one by one, $j = 1, \ldots, n$, such that once x_{t+1}^j is feasible in (9.74) then it remains feasible, cf. the algorithms in Section 9.3. All algorithms in this chapter are forwards algorithms in this sense.

Proposition 9.4.1 *Consider the problem (9.72) - (9.76) and assume that r_i is concave and additively separable with respect to x_i^j, $j = 1, \ldots, n$ and u_i (i.e. $r_i(x_i, u_i) = \sum_{j=1}^n r_i^{x^j}(x_i^j) + r_i^u(u_i)$, $i = 0, \ldots, N-1$, $r_N(x_n) = \sum_{j=1}^n r_N^j(x_N^j)$), continuously differentiable with respect to x_i and that F_i^x is diagonal with positive diagonal elements. Assume that A^+, (9.81), holds for all i.*

Apply a forwards algorithm. Assume that at a certain time during calculations the following hold for a forwards trajectory up to stage t_2: Upper (lower) tangency occurs at $i = t_1 + 1$ and lower (upper) tangency occurs at $i = t_2$, $t_1 < t_2$, for $j = 1, \ldots, n$, i.e., $\underline{x}_i \leq x_i \leq \overline{x}_i$, $i = 0, \ldots, t_2$ and either $x_{t_1+1} = \overline{x}_{t_1+1}$ and $x_{t_2} = \underline{x}_{t_2}$, or $x_{t_1+1} = \underline{x}_{t_1+1}$ and $x_{t_2} = \overline{x}_{t_2}$.

Then t_1 is a decision horizon, i.e., the controls $\{u_0^*, \ldots, u_{t_1}^*\}$ and the trajectory $\{x_0^*, \ldots, x_{t_1+1}^*\}$ are optimal.

Proof. We first show certain monotonicity relations. Due to the structure of r_i^x and F_i^x (with positive diagonal elements $F_i^{x^j}$) the adjoint relation (9.78) may be written independently for $j = 1, \ldots, n$ as

$$\nabla_{x_i^j} r_i^{x^j}(x_i^j) + p_{i+1}^j F_i^{x^j} - p_i^j \begin{cases} \leq 0 & \text{if} \quad \underline{x}_i^j = x_i^{*j} \\ = 0 & \text{if} \quad \underline{x}_i^j < x_i^{*j} < \overline{x}_i^j \\ \geq 0 & \text{if} \quad x_i^{*j} = \overline{x}_i^j \end{cases}$$

Consider a forwards trajectory starting at $i = 0$. Assume first that the middle option in (9.78) holds. Since $F_i^{x^j} > 0$ and $r_i^{x^j}$ is concave an increase (decrease) in p_i^j and/or x_i^j will permit that p_{i+1}^j need not be decreased (increased) in order to keep (9.78) fulfilled. Further, x_i^j is non-decreasing as a function of p_t^j, $t \leq i$, as is easily seen by adapting the first part of Proposition 9.2.1. Now consider the case where a state constraint is active (and the middle option in (9.78) is no longer relevant) such that $x_s^j = \overline{x}_s^j$ (similar argumentation will hold for $x_x^j = \underline{x}_s^j$). An increase in p_{s+1}^j need not be followed by an increase in p_s^j, cf. the lower option in (9.78). In conclusion, for each dimension j in x_i and p_i there are monotonicity and adjoint relations that are valid irrespective of the other dimensions k.

Now assume that all u_i^* from (9.77) are unique. Consider a forwards trajectory such that $x_{t_1+1} = \overline{x}_{t_1+1}$ and $x_{t_2} = \underline{x}_{t_2}$ (similar argumentation applies if the other bounds are relevant). Now p_{t_2+1} may be defined according to the middle option of the adjoint relation. If this produces a feasible x_{t_2+1} then this process may be repeated forwards, and if $\underline{x}_N$ is reached, the optimal solution is found. (If the final point x_N is free or bounded like in (9.74) the argumentation is easily extended, cf. also (9.46).) If x_s is infeasible for some s, $t_2 < s$, then a p_s^j must be changed. If p_s^j is increased this is propagated backwards as increases in p_i^j, $i < s$, however, as described above, no longer back than to $i = t_1 + 2$. In order that the constraints $\underline{x}_{t_2}^k \leq x_{t_2}^k$ are not violated $p_{t_2+1}^k$, $k \neq j$, need not be decreased, cf. (9.81), so that also $p_{t_1+1}^k$, $k \neq j$, are left unchanged. If on the other hand p_s^j is decreased, this is propagated backwards as decreases in p_i^j, $i < s$, however, no longer back than to $i = t_2 + 1$ (implying that $p_{t_1+1}^j$ is unchanged and that also $p_{t_1+1}^k$, $k \neq j$, are left unchanged). In conclusion, p_i will be left unchanged for $i \leq t_1 + 1$ and therefore also u_i^* and x_{i+1}^* are unchanged for $i \leq t_1$, i.e. $\{u_0^*, \ldots, u_{t_1}^*\}$ and $\{x_0^*, \ldots, x_{t_1+1}^*\}$ are optimal.

If not all u_i^* and hence not x_{t_2} are unique we enlarge the above argumentation with the following observations (cf. also the reasoning in relation to Lemma 1 on page 256). If $x_{t_2+1}^j < \underline{x}_{t_2+1}^j$ then first $x_{t_2+1}^j$ is attempted increased by selection of another optimal u_s, $0 \leq s \leq t_2$ which is nonunique. As $x_{t_1+1} = \overline{x}_{t_1+1}$, then obviously $t_1 + 1 \leq s \leq t_2$ and x_{t_1+1} will not be changed. If it is not possible to make $x_{t_2+1}^j$ feasible this way, $p_{t_2+1}^j$ must be increased and the above argumentation applies. Similarly if $x_{t_2+1}^j > \overline{x}_{t_2+1}^j$ then first $x_{t_2+1}^j$ is attempted decreased by selection of another optimal u_s, $0 \leq s \leq t_2$ which is nonunique. As $x_{t_2} = \underline{x}_{t_2}$, obviously $t_2 \leq s$ and x_{t_1+1} will not be changed. If it is not possible to make $x_{t_2+1}^j$ feasible this way, $p_{t_2+1}^j$ must be decreased and the above argumentation applies. In conclusion, the calculations of the algorithm motivated by stages later than t_2 will influence neither x_i nor p_i for $i \leq t_1 + 1$. Therefore $\{u_0^*, \ldots, u_{t_1}^*\}$ and $\{x_0^*, \ldots, x_{t_1+1}^*\}$ are optimal and t_1 is a decision horizon. □

Once a decision horizon is detected, the problem (9.72) - (9.76) may be reformulated such that the initial stage is $i = t_1 + 1$ and the initial state is that particular x_{t_1+1} found ($\overline{x}_{t_1+1}$ or $\underline{x}_{t_1+1}$). Thus, the detection of decision horizons permits that the problem is solved as a sequence of smaller problems.

The key idea in the above development was the monotonicity relations (9.81). The physical background may in relation to e.g. Example 9.4.1 be interpreted such that the state variables compete for (some of) the controls, i.e., a change in u_i^1 will change x_{i+1}^1 and x_{i+1}^2 in opposite directions. The conditions of Proposition 9.4.1 that all $x_{t_1}^j$ should be at the same boundary ($\overline{x}_{t_1}^j$ or $\underline{x}_{t_1}^j$) and all $x_{t_2}^j$ should be at the opposite boundary ($\underline{x}_{t_2}^j$ or $\overline{x}_{t_2}^j$, respectively) may be interpreted in relation to the same idea.

In Section 1.5 we considered a different case with two state variables (page 45 ff.). Here the state variables do not compete for any of the controls, rather they are influenced similarly by the control u_i^1 (parameters γ_i^a and γ_i^b are both positive):

$$x_{i+1}^a = \alpha^a x_i^a + \gamma_i^a u_i^1 + u_i^2 - u_i^3 - \beta_i^a \tag{9.82}$$

$$x_{i+1}^b = \alpha^b x_i^b + \gamma_i^b u_i^1 - u_i^4 - \beta_i^b \tag{9.83}$$

For such case it is again possible to formulate decision horizon results. For $n = 2$ we formulate assumption A^- as the following conditions, cf. (9.81): to any x_i, p_{i+1} and $\epsilon \geq 0$ there is a $\delta^1 \geq 0$ such that

$$\begin{aligned}&(x_{i+1}^*(x_i, p_{i+1}^1 - \delta^1, p_{i+1}^2 + \epsilon))^1 = \\ &(x_{i+1}^*(x_i, p_{i+1}^1, p_{i+1}^2))^1\end{aligned} \tag{9.84}$$

and a $\delta^2 \geq 0$ such that

$$\begin{aligned}&(x_{i+1}^*(x_i, p_{i+1}^1 + \epsilon, p_{i+1}^2 - \delta^2))^2 = \\ &(x_{i+1}^*(x_i, p_{i+1}^1, p_{i+1}^2))^2\end{aligned} \tag{9.85}$$

We then have

Proposition 9.4.2 *Consider the problem (9.72) - (9.76) with $n = 2$ and assume that r_i is concave and additively separable with respect to x_i^j, $j = 1, \ldots, n$ and u_i (i.e. $r_i(x_i, u_i) = \sum_{j=1}^n r_i^{xj}(x_i^j) + r_i^u(u_i)$, $i = 0, \ldots, N-1$, $r_N(x_n) = \sum_{j=1}^n r_N^j(x_N^j)$), continuously differentiable with respect to x_i*

and that F_i^x is diagonal with positive diagonal elements. Assume that A^-, (9.84) - (9.85), holds for all i.

Apply a forwards algorithm. Assume that at a certain time during calculations the following hold for a forwards trajectory up to stage t_2: Upper (lower) tangency occurs at $i = t_1 + 1$ for $j = 1$, lower (upper) tangency occurs at $i = t_1 + 1$ for $j = 2$, lower (upper) tangency occurs at $i = t_2$ for $j = 1$, upper (lower) tangency occurs at $i = t_2$ for $j = 2$, $t_1 < t_2$.

Then t_1 is a decision horizon, i.e., the controls $\{u_0^*, \ldots, u_{t_1}^*\}$ and the trajectory $\{x_0^*, \ldots, x_{t_1+1}^*\}$ are optimal.

Proof. The proof may follow the ideas of Proposition 9.4.1 with suitable interchanges of the direction of movements, according to the differences in signs between (9.81) and (9.84) - (9.85), and bounds (9.74) reached. □

9.5 Conclusions

Forwards algorithms highlight the advantages of applying the stagewise perspective on the optimal control problem. They contribute towards the understanding of the nature of the problem and they contribute towards designing efficient algorithms. As analyzed in Section 9.2, certain monotonicity relations hold for the optimal control problem with $n = 1$ under lenient assumptions. This adds to the characterization of the solution structure in particular where the end point $\underline{x}_N$ or the initial point $\underline{x}_0$ is taken as parameter.

The planning horizon results, developed in Section 9.4 for multistate problems, are also expedient for the characterization of the solution. The detection of decision horizon results by forwards algorithms will permit that the problem is split into two subproblems. Of these, the first one is already solved; moreover, the solution is valid irrespective of the data beyond the forecast horizon. Thus, planning horizon results are expedient in relation to data for which uncertainties increases with time index i.

The forwards perspective also permits the development of efficient algorithms as observed for the problem of isotone regression of Section 1.7. For the QLE problems the computational complexity is the same as for the backwards DP algorithm, cf. Section 4.3. For the problem with inequality constraints the computational complexity results for the Ansgar algorithm of Sections 9.2 and 9.3 are again the same as for DP, cf. Section 4.4.

In summary, the forwards perspective elegantly contributes to the purpose of optimization in relation to mathematical modeling that were formulated in the Introduction on page 3: to characterize the optimal solution, to interpret it and to find it.

Chapter 10

Conclusions and Further Research

The background and motivation for the thesis is that the OCP is appropriate for modeling many real life problems of analysis and decision making within the operations research framework. Therefore it is relevant to increase the understanding of the OCP in order to obtain further benefit from this model within the context of the application.

The thesis has exploited relevant results from modern mathematical programming, while insisting on the specificity of the OCP. Partial conclusions have been formulated at the end of the previous chapters, and it therefore seems relevant to point out here more general results and perspectives.

Conclusions

It is demonstrated in the thesis that the OCP is a widely facetted problem that permits a detailed analysis along many perspectives, from formulations and interpretations of optimality conditions to formulation of algorithms. It has further been demonstrated that central theoretical perspectives and results from mathematical programming, including decomposition ideas, duality theory, sensitivity analysis and algorithm analysis, may be applied in a nontrivial ways to enhance the understanding of the OCP.

It is shown in the thesis that all classical optimality principles for the OCP may be interpreted in relation to the upper boundaries. This in turn permits the interrelationships between the optimality conditions to be more clearly seen and appreciated, and consequently classical distinctions, such as between maximum principles and dynamic programming ideas, become less absolute. This in turn simplifies and enhances interpretations, sensitivity analyses and other characterizations of the optimal solution and its neighborhood.

Instrumental in this is application of nonlinear auxiliary functions. While such linear functions (i.e., given by the costate vectors) are natural from the classical maximum principle perspective, the generalization is precisely what opens up for the unified perspective on maximum principle and dynamic programming approaches. Also nonlinear functions are seen to be applicable in duality analyses, while maintaining the stagewise decomposition property. As revealed by this perspective, the auxiliary functions may be interpreted as stagewise price functions.

Also in relation to algorithms the upper boundaries/price functions perspective proves beneficial. Primal and dual decomposition, for instance, may be interpreted alternatively as a distinction between application of the smaller upper boundaries and application of stagewise price functions; at the optimal solution, the two sets of functions have intimately relationships.

The thesis also demonstrates that the complex of optimality conditions, expressed alternatively in maximum principle and dynamic programming formulations, naturally lead to the formulation of a variety of algorithms which have be formulated, analyzed and interpreted stagewise.

Further Research

While many of the results in the thesis may be relatively directly applied in operations research activities there have also been identified a number of directions in which further research may prove beneficial. We shall here point to those areas that to our opinion seem most promising.

First, we have identified a number of algorithms, presented, motivated and interpreted mainly according to the underlying optimality conditions and decomposition principles. Each particular algorithm has been given a preliminary analysis, typically in the form of theoretical convergence analysis. However, in order to appropriately evaluate the algorithms other aspect should be investigated, in terms of numerical aspects, implementation detail, and practical performance on relevant classes of test problems.

Second, throughout the thesis nonlinear price functions have been used as an integral part of the analysis in relation to optimality conditions and in relation to classification of algorithms. Potentials for application of nonlinear price functions are presented, as witnessed by the derived results on optimality conditions, in particular that application of nonlinear price functions are not prohibitive for the stagewise decomposition, and the linkage to duality. However, much is lacking on the application side. The analysis shows that there are strong relations between the upper boundaries, i.e., properties of the problem, and the price functions that are applied. Therefore specific problems should be analyzed in order to characterize appropriate price functions, and more general principles of construction of applicable price functions should be attempted. Obviously also this will have implications for algorithmic development.

Third, one area where dynamic problems are particularly relevant is where stochastic elements are present. While the thesis does not explicitly treat such problems, it is obvious that the presented extensions on the results on the deterministic OCP will have potential applications also to the stochastic discrete time optimal control problem. The thesis emphasizes relationships between various optimality conditions and draws on basic optimality conditions and results. Therefore the recent renewed attacks on the stochastic problem, such as stochastic dual dynamic programming and scenario analysis, may be naturally interpreted within the framework of the thesis, as these techniques rely heavily on classical principles (nested Benders' decomposition and Lagrangian relaxation, respectively). This highlights the need for identifying and extending the underlying structures and theories and indicates that the results from the thesis may be extended in the direction of the stochastic discrete time optimal control problem.

Chapter 11

Literature

Author's related publications in operations research, discrete time optimal control and energy systems analysis

Optimal Control of a Servomechanism, Proceedings, 1st International Conference on Applied Modeling and Simulation, Lyon, 1981, pp. 143-149, with R.V.V.Vidal.

Planning of the Danish Natural Gas System, in: Large Scale Systems: Theory, Applications and Impacts, IFAC/IFORS Conference Proceedings, Warsaw, 1983.

Optimization of Discrete-Time Systems, The Upper Boundary Approach, Lecture Notes in Control and Information Sciences 51, Springer-Verlag, 1983, with Zbigniew Nahorski and R.V.V.Vidal.

The Discrete-Time Maximum Principle: A Survey and Some New Results, Int. J. Control, Vol. 40, No. 3, pp. 533-554, 1984, with Zbigniew Nahorski and R.V.V.Vidal.

Optimizing Combined Power and Heat Production with Dual Fuels, pp. 854-863 in J.-P. Brans (ed.): Operational Research '84, Elsevier Science Publishers B.V., 1984.

Investigation of a New Numerical Method for Control of a Watersupply Network, pp. 153-158 in L.V.Tavares and J.E. da Silva (eds.): Systems Analysis Applied to Water and Related Land Resources, Proceeding, Lisbon 2-4 October 1985, Pergamon Press, 1985, with C.S.Nielsen.

Optimal Electrical Dispatch and Unit Commitment with Nonconvex Costs, pp. 339-347 in E. Szelke and J. Browne (ed.): Advances in Production Management Systems, Elsevier Science Publishers B.V., 1986, with Mogens Pedersen.

A Forward Maximum Principle Algorithm with Decision Horizon Results, Applied Mathematics and Computation, 24: 65-75, 1987.

Introductory Remarks: Some Connections between Mathematical Programming, Continuous and Discrete Time Optimal Control Theory, Control and Cybernetics, Vol. 17, Nos. 2-3, pp. 107-113, 1988, with Zbigniew Nahorski.

The Upper Boundary Approach to Constrained Discrete Time Optimal Control, Control and Cybernetics, Vol. 17, Nos. 2-3, pp. 145 - 172, 1988, with Zbigniew Nahorski.

Comments on 'A Discrete Optimal Control Problem For Descriptor Systems', IEEE Trans. Automat. Contr., Vol. AC-35, pp. 985-987, 1990, Vol. 36, p. 767, 1991.

Engineering Applications of Discrete Time Optimal Control, European Journal of Operations Research, Vol. 45, pp. 241-250, 1990, with R.V.V.Vidal.

A Discrete-Time Optimal Control Algorithm Based on the Generalized Maximum Principle,

pp. 85-102 in Proceedings, First Nordic Meeting on Mathematical Programming, Copenhagen, IMSOR, Technical University of Denmark, 1990, with R.V.V.Vidal.

Lagrangian Relaxation of an Optimal Control Problem, pp. 103-114 in Proceedings, First Nordic Meeting on Mathematical Programming, Copenhagen, IMSOR, Technical University of Denmark, 1990.

Mathematical Modelling as a Basis for Operational Optimization of District Heating Systems, in Proceedings, 26. UNICHAL-Congress, 8.-10. June 1993, with Atli Benonysson and Benny Bøhm.

Stochastic Heat Storage Problem-Solved by the Progressive Hedging Algorithm, Energy Conversion and Management, Vol. 35, No. 12, pp. 1157 - 1171, 1994, with Ólafur P. Pálsson.

Optimal Scheduling of Coproduction with a Storage, Eng. Opt., Vol 22., pp. 267-281, 1994, with Jens M. Rygaard.

Optimal Scheduling of Heat Production with Storage and Stochastic Demands, pp. 411-418 in M.T.Kangas and P.D.Lund (eds.): Thermal Energy Storage: Better Economy, Environment, Technology, Proceedings, Calorstock '94, August 22-25, 1994, Helsinki, Finland, with Claus Jørgensen.

Operational Optimization of District Heating (DH) Systems-Using a DH as a Heat Storage, pp. 459-467 in M.T.Kangas and P.D.Lund (eds.): Thermal Energy Storage: Better Economy, Environment, Technology, Proceedings, Calorstock '94, August 22-25, 1994, Helsinki, Finland, with Hongping Zhao and Benny Bøhm.

Hydro-Thermal Scheduling. Computational Approaches for Optimal Scheduling of Mixed Hydro-Thermal Systems, pp. 309-320 in Proceedings, 'Hydropower Into the Next Century', Barcelona, 1995, Aqua-Media International (UK), 1995, with Peter Børre Eriksen, Jens Pedersen, Arne Johannesen and Arne Haugstad.

On Optimum Operation of a CHP Type District Heating System by Mathematical Modelling, Euroheat & Power-Fernvärme International, 11/1995, pp. 618-622, with Hongping Zhao and Benny Bøhm.

Operational Optimization in a District Heating System, Energy Conversion and Management, Vol. 36, No. 5, pp. 297-314, 1995, with Atli Benonysson and Benny Bøhm.

A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities and multiple objectives, Annals of Operations Research, Vol. 60, pp. 193-208, 1995, with Oli B.G.Madsen and Jens Moberg Rygaard.

A Method to Perform Probabilistic Production Simulation Involving CHP Units, IEEE Trans. PWRS, Vol. 11, No. 2, May 1996, pp. 1031-1036, with Charlotte Søndergren.

Power Network Design, Control and Cybernetics, Vol. 25, No. 1, pp. 97-120, 1996, with Jens M. Rygaard and Berno Wibbels.

Hydro and Thermal Scheduling by the Decoupling Method, Electric Power Systems Research, Vol. 38, Issue 3, pp. 43-49, 1996, with Peter Børre Eriksen and Claus Jørgensen.

Mathematical Models in Economic Environmental Problems, Risø-R-955(EN), 1996, with Zbigniew Nahorski.

Simulations of Combined Heat and Power Systems With Heat Storage, pp. 205-210 in K. Ochifuji and K. Nagano (eds.): Proceedings, Megastock '97, Japan, 1997, with Sturla Sæther.

Incorporation of Thermal Stochastic Elements Into a Hydro-Thermal Model, pp. 251-258 in Broch, Lysne, Flatabø and Helland-Hansen (eds.): Proceedings, Hydropower '97, Trondheim 1997, with Claus Jørgensen.

Optimal Feedback Solution of a Constrained Stochastic One-Storage Model, Optimal Control Applications and Methods, Vol. 18, pp. 445-452, 1997, with Claus Jørgensen.

Simulation Tool for Expansion Planning of Combined Heat and Power, Proceedings, 6th International Symposium on District Heating and Cooling Simulation, August 28-30, 1997, Reykjavik, Iceland, with Helge V. Larsen and Halldór Pálsson.

Sequential Probabilistic Methods for Power System Operation and Planning, Task Force 13 of Advisory Group 38.03, Electra, No. 179, pp. 68-99, August 1998, with J.van Hecke, R. N. Allan, E. Dialynas, T. Doan, J. Gheury, G. Lovas, S. Panichelli, E. Pelgrum, V. Ringeissen, M. Roggenbau, J. Roman, M. T. Schilling, W. Welssow.

Probabilistic production simulation including combined heat and power plants, Electric Power Systems Research, Vol. 48, pp. 45-56, 1998, with Helge V. Larsen and Halldór Pálsson.

References

Aris, Rutherford; Richard Bellman and Robert Kalaba: Some Optimization Problems in Chemical Engineering, Chemical Engineering Progress Symposium Series, No. 31, Vol. 56, 1960, pp. 95-102.

Aronson, J.E. and G.L. Thompson: A Survey on Forward Methods in Mathematical Programming, Large Scale Systems 7, 1984, pp. 1-16.

Ashchepkov, L.T. and R. Gabasov: Optimization of Discrete Systems, Differencyalnye Uravnenya, Vol. 8, No. 6, 1972 (in Russian, communicated by Z. Nahorski).

Aubin, J.P. and I. Ekeland: Estimates of the Duality Gap in Nonconvex Optimization, Math. Oper. Res., Vol. 1, 1976, pp. 225-245.

Bannister, C. H., and Kaye, R.J.: A rapid method for optimization of linear systems with storage, Operations Research Vol. 39, No. 2, March-April 1991, pp. 220-232.

Barlow, R.E. and H.D.Bruuk: The Isotone Regression Problem and Its Dual, J. American Statistical Association, Vol. 67, No. 337, pp. 140-147, 1972.

Bazaraa, Mokhtar and C.M. Shetty: Nonlinear Programming, Theory and Algorithms, Wiley, 1979.

Beckmann, Martin J.: A Note on the Optimal Rates of Resource Exhaustion, Review of Economic Studies, Symposium Issue, 1974, pp. 121-122.

Bellman, R.: Dynamic Programming, Princeton University Press, New Jersey, 1957.

Bellman, R. and S.E. Dreyfus: Applied Dynamic Programming, Princeton University Press, New Jersey, 1962.

Bertsekas, Dimitri P.: Partial Conjugate Gradient Method for a Class of Optimal Control Problems, IEEE Trans. Automatic Control, Vol. AC-19, pp. 209-217, 1974.

Bertsekas, Dimitri P.: Convergence of Discretization Procedures in Dynamic Programming, IEEE Transaction on Automatic Control, Vol. AC-20, June 1975, pp. 415-419.

Bertsekas, Dimitri P.: Convexification Procedures and Decomposition Methods for Nonconvex Optimization Problems, J. Optimiz. Theory Appl., Vol 29, pp. 169-179, 1979.

Bertsekas, Dimitri P.: Constrained Optimization and Lagrange Multiplier Methods, Academic Press, 1982.

Bertsekas, Dimitri P.: Projected Newton Methods for Optimization Problems with Simple Constraints, SIAM J. Control and Optimization, Vol. 20, No. 2, pp. 221-246, 1982a.

Bertsekas, Dimitri P.: Dynamic Programming: Deterministic and Stochastic Models, Prentice-Hall, 1987.

Best, M.J. and Chakravarti, N.: Active Set Algorithms for Isotonic Regression: A Unifying Framework, Mathematical Programming, Vol. 47, pp. 425-439, 1990.

Boltyanskii, V.G.: Optimal Control of Discrete Systems, Wiley, 1978 (Russian original 1973).

Butkovski, A.G.: Necessary and Sufficient Conditions of Optimality of Discrete Control Systems, Automation and Remote Control, Vol. 24, No. 8, pp. 963-970, 1963.

Butkovski, A.G.: Theory of Optimal Control of Distributed Parameter Systems, Nauka, Moskov, 1965 (in Russian, communicated by Z. Nahorski).

Canon, M.D., C.D. Cullum Jr., and E. Polak: Theory of Optimal Control and Mathematical Programming, McGraw-Hill, 1970.

Chang, S.S.L.: Digitized Maximum Principle, Proc. IRE, pp. 2030-2031, 1960.

Chang, S.S.L.: Synthesis of Optimal Control Systems, McGraw-Hill, New York, 1961.

Chang, Shi-Chung; Chang, Tsu-Shuan and Luh, Peter B.: A Hierarchical Decomposition for Large-scale Optimal Control Problems with Parallel Processing Structure, Automatica, Vol. 25, No. 1, 1989, pp. 77-86.

Clarke, Frank H.: Optimization and Nonsmooth Analysis, John Wiley & Sons, 1983.

Cullum, Jane: Finite-dimensional approximations of state-constrained continuous optimal control problems, SIAM J. Control, Vol. 10, No. 4, pp. 649-670, 1972.

Debreau, Gerard and Koopmans, Tjalling C.: Additively decomposed quasiconvex functions, Mathematical Programming Vol. 24, pp. 1-38, 1982.

Denardo, Eric V.: Dynamic Programming, Models and Applications, Prentice-Hall, 1982.

Denardo, Eric V., Gur Hubermam and Uriel G. Rothblum: Optimal Locations on a Line Are Interleaved, Operations Research Vol. 30, pp. 745-759, 1982.

Doležal, J.: Necessary optimality conditions for a class of nondifferentiable discrete control problems, Problems Control Inf. Theory, Vol. 11, pp. 77-83, 1982.

Doležal, J.: Non-smooth and non-convex problems in discrete optimal control, Int. J. System Sci., Vol. 13, pp. 969-978, 1982.

Doležal, J.: Necessary conditions for Pareto optimality in non-differentiable discrete control problems, Control and Cybernetics, Vol. 17, No. 2-3, pp. 213-223, 1988.

Dreyfus, Stuart E.: Dynamic Programming and the Discrete Maximum Principle, Symposia Mathematica, Vol. 19, pp. 310-312, Academic Press, 1976.

Dreyfus, S.E. and Y.C.Kan: A General Dynamic Programming Solution of Discrete-Time Linear Optimal Control Problems, IEEE Trans. Automatic Control Vol. AC-18, No. 3, pp. 286-289, 1973.

Dreyfus, Stuart E. and Averill M. Law: The Art and Theory of Dynamic Programming, Academic Press, 1977.

Dunn, J.C. and Bertsekas, D.P.: Efficient Dynamic Programming Implementations of Newton's Method for Unconstrained Optimal Control Problems, JOTA, Vol. 63, No. 1, pp. 23- 38,1989.

Dyer, Peter and McReynolds, Stephen R.: The Computation and Theory of Optimal Control, Academic Press, 1970.

Edmunds, T.A. and Bard, J.F.: Time-Axis Decomposition of Large-Scale Optimal Control Problems, JOTA, Vol. 67, No. 2, November 1990, pp. 259-277.

Eppen, G.D., F.J. Gould, B.P. Pashigian: Extension of the Planning Horizon Theorem in the Dynamic Lot Size Model, Management Science, Vol. 15, pp. 268-277, 1969.

Everett, H.: Generalized Lagrange Multiplier Method for Solving Problems of Optimum Allocation of Resources, Opns. Res., Vol. 2, No. 3, pp. 399-417, 1963.

Fan, L.T. and Ch. S. Wang: The Discrete Maximum Principle, Wiley, 1964.

Feichtinger, Gustav and Richard F. Hartl: Optimale Kontrolle Ökonomischer Prozesse, Walter de Gruyter, 1986.

Feng, X.; Mukai, H. and Brown, R.H.: New Decomposition and Convexification Algorithm for Nonconvex Large-Scale Primal-Dual Optimization, JOTA, Vol. 67, No. 2, 1990, pp. 279-296.

Ferreira, José: Optimal Control of Discrete-Time Systems, With Applications, The Institute of Mathematical Statistics and Operations Research, Technical University of Denmark, 1984.

Ferreira, José A. Soeiro: Maximum Principles Applied to A Model of Consumer Brand Choice, Optimal Control Applications & Methods, Vol. 11, pp. 21-37, 1990.

Ferreira, José and R.V.V. Vidal: On the connections between mathematical programming and discrete optimal control, Proc. 12th IFIP Conf. Syst. Modelling and Optimization, Budapest 1985, pp. 234-243, Springer-Verlag, 1986.

Fiacco, Anthony V.: Sensitivity analysis for nonlinear programming using penalty methods, Mathematical Programming, Vol. 10, pp. 287-311, 1976.

Fiacco, A. V. and Kyparisis, J.: Convexity and Concavity Properties of the Optimal Value Function in Parametric Nonlinear Programming, JOTA, Vol. 48, No. 1, pp. 95- 126, 1986.

Fleming, Wendell H. and Raymond W. Rishel: Deterministic and Stochastic Optimal Control, Springer-Verlag, 1975.

Florian, M., J.K.Lenstra and A.G.H.Rinnoy Kan: Deterministic Production Planning: Algorithms and Complexity, Management Science Vol. 26, No. 7, pp. 669-679, 1980.

Fox, B.L.: Discretizing dynamic programs, JOTA, Vol. 11, No. 3, 228-234, 1973.

Gabasov, R. and F.M. Kirillova: Extending L.S. Pontryagin's Maximum Discrete Systems, Automation and Remote Control, Vol. 27, No. 11, pp. 1878-1882, 1966.

Gabasov, R.: Theory of Optimal Discrete Processes, Zurnal Vycisl. Mat. i Mat. Fiz., Vol. 8, No. 4, 1968 (in Russian, communicated by Z. Nahorski).

Gabasov, R and N.W. Tarasenko: Necessary High-Order Conditions of Optimality for Discrete Systems, Automation and Remote Control, Vol. 32, No. 1, pp. 50-57, 1971.

Gauvin, Jacques: A necessary and sufficient regularity condition to have bounded multipliers in nonconvex programming, Mathematical Programming 12, pp. 136-138, 1977.

Gauvin, Jacques: Shadow prices in nonconvex mathematical programming, Mathematical Programming, Vol. 19, pp. 300-312, 1980.

Gauvin, Jacques and Tolle, Jon W.: Differential stability in nonlinear programming, SIAM J. Control and Optimization, Vol. 15, No. 2, pp. 294-311, 1977.

Gauvin, Jacques and Debeau, Francois: Differential properties of the marginal function in mathematical programming, Mathematical Programming Study 19, pp. 101-119, 1982.

Gawande, M. and Dunn, J.C.: A Projected Newton Method in a Cartesian Product of Balls, JOTA Vol. 59, No.1, pp. 45-69, 1988.

Halkin, H.: Optimal Control for Systems Described by Difference Equations, In: C.T. Leondes (ed.): Advances in Control Systems, Academic Press, Vol. 1, pp. 171-196, 1964.

Halkin, H.: A Maximum Principle of the Pontryagin Type for Systems Described by Nonlinear Difference Equations, SIAM J. Control, Vol. 4. No. 1, pp. 90-111, 1966.

Hansen, Jens Anker: Indfasningsproblem løst ved relaksation, The Institute of Mathematical Statistics and Operations Research, Technical University of Denmark (M.Sc. Thesis No. 12/87), 1987.

Hartl, Richard F., Suresh P. Sethi and Raymond G. Vickson: A survey of the maximum principles for optimal control problems with state constraints, SIAM Review, Vol. 37, No. 2, pp. 181-218, 1995.

Holtzman, J.M.: Convexity and the Maximum Principle for Discrete Systems, IEEE Trans. Autom. Control, Vol. AC-11, No. 1, pp. 30-35, 1966a.

Holtzman, J.M.: On the Maximum Principle for Nonlinear Discrete-Time Systems, IEEE Trans. Autom. Control, Vol. AC-11, No. 2, pp. 273-274, 1966b.

Holtzman, J.M. and H. Halkin: Directional Convexity and the Maximum Principle for Discrete Systems, SIAM J. Control, Vol. 4, No. 2, pp. 263-275, 1966.

Horn, F. and R. Jackson: Discrete Maximum Principle, Ind. & Eng. Chem. Fundamentals, Vol. 4, No. 1, pp. 110-112, 1965. Vol. 4, No. 4, pp. 487-488, 1965.

Howson, H.R. and Sancho, N.G.F.: A New Algorithm for the Solution of Multi-Stage Dynamic Programming Problems, Mathematical Programming 8 (1975), pp. 104-116.

Jacobsen, David H. and Mayne, David Q.: Differential Dynamic Programming, American Elsevier, New York, 1970.

Jackson, R. and F. Horn: On Discrete Analogues of Pontryagin's Maximum Principle, Int. J. Control, Vol. 1, No. 4, pp. 389-395, 1965.

Johnsen, Henrik: A Newton Method for Solving Non-Linear Optimal Control Problems with General Constraints, Linköping Studies in Science and Technology Dissertations No. 104, 1983.

Johnson, Sharon A.; Stedinger, Jury R.; Shoemaker, Christine A.; Li, Ying; Tejada-Guibert, José Alberto: Numerical Solution of Continuous-State Dynamic Programs Using Linear and Spline Interpolation, Operations Research, Vol. 41, No. 3, May-June 1993, pp. 484-500.

Jørgensen, Claus: En metode til dynamisk programmering, The Institute of Mathematical Statistics and Operations Research, The Technical University of Denmark (M.Sc. Thesis), 1993.

Jørgensen, Claus, and Hans F. Ravn: Optimal Feedback Solution of a Constrained Stochastic One-Storage Model, Optimal Control Applications and Methods, Vol. 18, pp. 445-452, 1997.

Katz, S.: A Discrete Version of Pontryagin's Maximum Principle, J. Electronics and Control, Vol. 13, pp. 179, 1962a.

Katz, S.: Best Operating Points for Staged Systems, Ind. & Eng. Chem. Fundamentals, Vol. 1, No. 4, 1962b.

Kiwiel, Krzysztof C.: Methods of Descent for Nondifferentiable Optimization, Lecture Notes in Mathematics 1133, Springer-Verlag, 1985.

Kleindorfer, Paul R. and Zvi Lieber: Algorithms and Planning Horizon Results for Production Planning Problems with Separable Costs, Operations Research, Vol. 27, No. 5, pp. 874-887, 1979.

Koppelhus, Søren: Prisdekomposition anvendt i elsektoren, The Institute of Mathematical Statistics and Operations Research, The Technical University of Denmark (M.Sc. Thesis No. 6/91), 1991.

Kornai, J. and Lipták, Th.: Two-level planning, Econometrica, Vol. 33, No. 1, pp. 141-169, January 1965.

Krotov, V.F.: Sufficient optimality conditions for discrete systems, Dokl. Akad. Nauk SSSR, Vol. 172, No. 1, pp. 18-21, 1967 (in Russian, communicated by Z. Nahorski).

Krotov, V.F.: A technique of global bounds in optimal control theory, Control and Cybernetics, Vol. 17, No. 2-3, pp. 115-144, 1988.

Larson, R. E. and Korsak, A.J.: A Dynamic Programming Succesive Approximations Technique with Convergence Proofs, Automatica, Vol. 6, 1970, pp. 245-252 (Part I), pp. 253-260 (Part II).

Lee, Fred N.: A Method to Eliminate Solution Trapping in Applying Progressive Optimality Principle to Short-Term Hydrothermal Scheduling, IEEE Transactions on Power Systems, Vol. 4, No. 3, pp. 935-942, August 1989.

Liao, Li-Zhi and Shoemaker, Christine A.,: Convergence in Unconstrained Discrete-Time Differential Dynamic Programming, IEEE Trans. AC, Vol. 36, No. 6, pp. 692-706, 1991.

Luenberger, David G.: Mathematical Programming and Control Theory: Trends of Interplay, pp. 102-133 in Arthur D. Geoffrion (ed.): Perspectives on Optimization: A Collection of Expository Articles, Addison-Wesly, 1972.

Luenberger, David G.: Linear and Nonlinear Programming, Addison-Wesley 1989.

Luus, Rein: Optimal control by dynamic programming using systematic reduction in grid size, Int. J. Control, 1990, Vol. 51, No. 5, pp. 995-1013.

MacRae, Duncan C.: A Dual Maximum Principle for Discrete-Time Linear Systems with Economic Applications, IEEE Trans-AC, pp. 49-52, February 1969.

Mayne, D.: A Second-order Gradient Method for Determining Optimal Trajectories of Non-Linear Discrete-Time Systems, Int. J. Control, Vol. 3, pp. 85-95, 1966.

Morin, Thomas L.: Computational Advances in Dynamic Programming, pp. 53-90 in Martin L. Putermann (ed.): Dynamic Programming and Its Applications, Academic Press, 1978.

Murray, Daniel M. and Sidney J. Yakowitz: Constrained Differential Dynamic Programming and Its Application to Multireservoir Control, Water Resources Research, Vol. 15, No. 5, pp. 1017-1027, October 1979.

Murray, Daniel M. and Sidney J. Yakowitz: The Application of Optimal Control Methodology to Nonlinear Programming Problems, Mathematical Programming 21, pp. 331-347, North-Holland 1981.

Nahorski, Z., H.F. Ravn and René Victor Valqui Vidal: Optimization of Discrete-Time Systems, The Upper Boundary Approach, Lecture Notes in Control and Information Sciences 51, Springer-Verlag, 1983.

Nahorski, Z., H.F. Ravn and R.V.V. Vidal: The Discrete-Time Maximum Principle: A Survey and Some New Results, Int. J. Control 40, pp. 533-554, 1984.

Nahorski, Z., Hans F. Ravn and René Victor Valqui Vidal: Properties of Upper Boundary Functions in Discrete Time Optimal Control, IMSOR, Technical University of Denmark, 1987.

Nahorski, Z. and H.F. Ravn: The Upper Boundary Approach to Constrained Discrete Time Optimal Control, Control and Cybernetics, Vol. 17, No. 2-3, pp. 145-172, 1988.

Nemhauser, G. L.: Introduction to Dynamic Programming, Wiley 1966.

Nielsen, Claus Stefan: Løsningsmetoder i Diskret Optimal Kontrolteori, The Institute of Mathematical Statistics and Operations Research, The Technical University of Denmark (M.Sc. Thesis No. 3/85), 1985.

Nielsen, Claus Stefan, and Hans F. Ravn: Investigation of a New Numerical Method for Control of a Watersupply Network, pp. 153-158 in L.V.Tavares and J.E. da Silva (eds.): Systems Analysis Applied to Water and Related Land Resources, Proceeding, Lisbon 2-4 October 1985, Pergamon Press, 1985.

Ohno, Katsuhiza: A New Approach to Differential Dynamic Programming for Discrete Time Systems, IEEE Transactions on Automatic Control, Vol. AC-23, No. 1, pp. 37-47, 1978.

Ortega, J.M and W.C.Rheinboldt: Iterative Solutions of Nonlinear Equations in Several Variables, Academic Press, 1970.

Outrata, J.V.: Duality Theory for a Class of Discrete Optimal Control Problems. In: Proc. 1978 IFAC Congress, Pergamon Press, Vol. 2, pp. 1085-1092, 1978.

Outrata, J.V.: Minimization of nonsmooth nonregular functions: Applications to discrete-time optimal control problems, Problems Control Inf. Theory, Vol. 13, pp. 413-424, 1984.

Outrata, J.V. and J. Jarusek: Duality Theory in Mathematical Programming and Optimal Control, Supplement to Kybernetica, Vol. 20, 1984/1985.

Pantoja, F.A. de O.: Differential dynamic programming and Newton's method, Int. J. Control, Vol. 47, No. 5, pp. 1539-1553, 1988.

Papageorgieu, Markos: Optimal Multireservoir Network Control by the Discrete Maximum Principle, Water Resources Research, Vol. 21, No. 12, pp. 1824-1830, 1985.

Parlar, Mahmut: A Decomposition Technique for an Optimal Control Problem with 'PQDZ' Cost and Bounded Controls, IEEE Trans. AC-27, No. 4, pp. 947-951, 1982.

Parlar, Mahmut and Vickson, R.G.: An Optimal Control Problem with Piecewise Quadratic Cost Functional Containing a 'Dead-Zone', Optimal Control Applications and Methods Vol. 1, pp. 361-372, 1980.

Polak, E.: Computational Methods in Optimization: A Unified Approach, Academic Press, 1971.

Polak, Elijah: An Historical Survey of Computational Methods in Optimal Control, SIAM Rewiev, Vol. 15, No. 2, pp. 553-584, April 1973.

Pontryagin, L.S., V.G. Boltyanski, R.V. Gamkrelidze and E.F. Mishchenko: The Mathematical Theory of Optimal Processes, Interscience, New York, 1962.

Propoi, A.I.: On a Problem of Optimal Discrete Control, Doklady Akad. Nauk SSSR, Vol. 159, No. 6, 1964 (in Russian, communicated by Z. Nahorski).

Propoi, A.I.: The Maximum Principle for Discrete Control Systems, Automation and Remote Control, Vol. 26, No. 7, pp. 1167-1177, 1965.

Propoi, A.: Methods of dynamic linear programming, in: S. Walukiewicz and A.P.Wierzbicki (eds.): Methods of Mathematical Programming, Polish Scientific Publishers, pp. 253-260, 1981.

Rakshit, A. and S. Sen: Sequential Rank-One/Rank-Two Updates for Quasi-Newton Differential Dynamic Programming, Optimal Control Applications & Methods, Vol. 11, pp. 95-101, 1990.

Ravn, Hans F.: A Forward Maximum Principle Algorithm with Decision Horizon Results, Applied Mathematics and Computation 24, pp. 65-75, 1987.

Ravn, Hans F.: Lagrangian Relaxation of an Optimal Control Problem, pp. 103-114 in Proceedings, First Nordic Meeting on Mathematical Programming, Copenhagen, IMSOR, Technical University of Denmark, 1990.

Ravn, Hans F. and R.V.V. Vidal: Engineering Applications of Discrete Time Optimal Control, European Journal of Operations Research, Vol. 45, pp. 241-250, 1990.

Roberts, S.M. and J.S. Shipman: Two-Point Boundary Value Problems: Shooting Methods, American Elsevier, New York, 1972.

Rockafellar, R.T.: Lagrange multipliers and subderivatives of optimal value functions in nonlinear programming, Mathematical Programming Study 17, pp. 28-66, 1982.

Rockafellar, R.Tyrell: Marginal values and second-order necessary conditions for optimality, Mathematical Programming, Vol. 26, pp. 245-286, 1983.

Rockafellar, R.T.: Directional differentiability of the optimal value function in a nonlinear programming problem, Mathematical Programming Study 21, pp. 213-226, 1984.

Rockafellar, R.Tyrell: Multistage convex programming and discrete-time optimal control, Control and Cybernetics, vol. 17, Nos. 2-3, pp. 225-245, 1988.

Ross, Sheldon: Introduction to Stochastic Dynamic Programming, Academic Press, 1989.

Rozonoer, L.I.: The Maximum Principle of L.S. Pontryagin in the Theory of Optimal Systems, Automation and Remote Control, Nos. 10-12, 1959.

Sage, Andrew P. and Chelsea C. White III: Optimum Systems Control, Prentice-Hall, 1977.

Seierstad, Atle and Knut Sydsæter: Optimal Control Theory with Economic Applications, North-Holland, 1987.

Sen, S. and S.J. Yakowitz: A Quasi-Newton Differential Dynamic Programming Algorithm for Discrete-Time Optimal Control, Automatica, Vol. 23, No. 6, pp. 749-752, 1987.

Sethi, Sureshi P. and Gerald L. Thompson: Optimal Control Theory, Martinus Nijhoff Publishing, 1981.

Shor, N.Z.: Minimization Methods for Non-Differentiable Functions, Springer-Verlag, 1985.

Sniedovich, Moshe: Dynamic Programming, Marcel Dekker, 1992.

Stephanopoulos, G. and A.W. Westerberg: The Use of Hestenes' Method of Multipliers to Resolve Dual Gaps in Engineering System Optimization, J. Optimiz. Theory Appl., Vol. 15, pp. 285-309, 1975.

Stoer, Josef and Christoph Witzgall: Convexity and Optimization in Finite Dimensions, I, Springer-Verlag, 1970.

Tabak, Daniel and Benjamin C. Kuo: Optimal Control by Mathematical Programming, Prentice-Hall, 1971.

Tamura, Hiroyuki: Decentralized Optimization for Distributed-lag Models of Discrete Systems, Automatica, Vol. 11, pp. 593-602, 1975.

Tanikawa, Akio and Huro Mukai: A New Technique for Nonconvex Primal-Dual Decomposition of a Large-Scale Separable Optimization Problem, IEEE Trans. AC-30, No 2, pp. 133-143, 1985.

Tatjewski, P.: On-line hierarchical control of steadystate systems using the augmented interaction balance method with feedback, Large Scale Systems 8, pp. 1-18, 1985.

Tind, Jørgen and Laurence A. Wolsey: An Elementary Survey of General Duality Theory in Mathematical Programming, Mathematical Programming 21, pp. 241-261, 1981.

Topkis, Donald M.: Minimizing a Submodular Function on a Lattice, Operations Research Vol. 26, No. 2, March-April, pp. 305-321 , 1978.

Turgeon, André: Incremental Dynamic Programming May Yield Nonoptimal Solutions, Water Resources Research, Vol. 18, No. 6, December 1982, pp. 1599-1604.

Vidal, René Victor Vidal: A Global Maximum Principle for Discrete-Time Control Problems, Eng. Opt., Vol. 10, pp. 77-84, 1986.

Vidal, R.V. Valqui: On the Sufficiency of the Linear Maximum Principle for Discrete-Time Control Problems, JOTA, Vol. 54 No. 3, pp. 583-589, 1987.

Vidal, René Victor Vidal: A simple convex optimization problem with many applications, IMA J. Mathematics Applied in Business and Industry, Vol. 5, pp. 15-23, 1993/4, 1994a.

Vinter, R.: Optimality and sensitivity of discrete time processes, Control and Cybernetics, Vol. 17, No. 2-3, pp. 191-211, 1988.

Vidal, R.V.V.: On the Optimal Sizing problem, J. Opl. Res. Soc., Vol. 45, No. 6, pp. 714-719, 1994.

Wagner, H.M. and T.M. Whitin: Dynamic Version of the Economic Lot Size Model, Management Science 5, pp. 89-96, 1958.

Watanabe, N., Y. Nishimura and M. Matsubara: Decomposition in Large System Optimization Using the Method of Multipliers, J. Optimiz. Theory Appl., Vol. 25, pp. 181-193, 1978.

White, D.J.: Dynamic Programming and Duality in Linear Programming, J. Mathematical Analysis and Applications, 51, pp. 695-704, 1975.

Wolfe, P.: The Simplex Method for Quadratic Programming, Econometrica Vol. 27, pp. 382-398, 1959

Yakovlev, W.M.: On Discrete Maximum Principle, Problemy Kibernetiki, No. 34, 1978.

Yakowitz, S.J.: Convergence Rate Analysis of the State Increment Dynamic Progamming Method, Automatica, Vol. 19, No. 1, pp. 53-60, 1983.

Yakowitz, Sidney and Brian Rutherford: Computational Aspects of Discrete-Time Optimal Control, Applied Mathematics and Computation 15, pp. 29-45, 1984.

Yakowitz, Sidney J.: The Stagewise Kuhn-Tucker Condition and Differential Dynamic Programming, IEEE Transactions on Automatic Control, Vol. AC-31, No. 1, pp. 25-30, January 1986.

Yakowitz, Sidney J.: Theoretical and computational advances in differential dynamic programming, Control and Cybernetics, Vol. 17, No. 2-3, pp. 173-189, 1988.

Yakowitz, Sidney J.: Algorithms and Computational Techniques in Differential Dynamic Programming, Control and Dynamic Systems, Vol. 31, pp. 75-91, 1989.

Zabel, Edward: Some Generalizations of an Inventory Planning Horizon Theorem, Management Science, Vol. 10, No. 3, pp. 465-471, 1964.

Zuo, Zhao-Qin: Two New Techniques for Optimal Control, IEEE Trans. Automatic Control, Vol. 36, No. 11, November 1991, pp. 1307-1310.

Zuo, Z.Q. and Wu, C.P.: Successive Approximation Technique for a Class of Large-Scale NLP Problems and Its Application to Dynamic Programming, JOTA, Vol. 63, No. 3, September 1989, pp. 515-527.