Dynamic Optimization

Niels Kjølstad Poulsen
Informatics and Mathematical Modelling
The Technical Unversity of Denmark

Latest revision: 1st September 2004

Preface

These notes are intended for use in connection to the dynamic part of the course in **Static and Dynamic optimization** given at **Informatics and Mathematical Modelling, The Technical University of Denmark**.

The notes heavily rely on the presentation and basic approach to dynamic optimization in (Vidal 1981) and (Ravn 1994). Another very important source of inspiration are (Bryson & Ho 1975), (Lewis 1986*b*), (Lewis 1992), (Bertsekas 1995) and (Bryson 1999).

Many of the examples and figures in the notes has been produced with Matlab and the software that comes with (Bryson 1999).

Contents

Chapter 1

Introduction

Let us start this introduction with a citation from S.A. Kierkegaard which can be found in (Bertsekas 1995):

Life can only be understood going backwards,
but it must be lived going forwards

This citation will become more apparent later on when we are going to deal with the Euler-Lagrange equations and Dynamic Programming.

Dynamic optimization involve several components. Firstly, it involves something describing what we want to achieve. Secondly, it involves some dynamics and often some constraints.

In this context we formulate what we want to achieve in terms of a a mathematical model. We normally denote this as a *performance index*, a *cost function* (if we are minimizing) or an *objective function*. The type of mathematical model of the dynamics, we are using in these notes, are the so called *state space models*. A very important concepts in this connection is the *state* or more precisely the *state vector*, which is a vector containing the *state variables*. These variable can intuitively be interpreted as a summary of the system history or a sufficient statistics of the history. Knowing these variable and the future inputs to the system (together with the system model) we are able to determine the future path of the system or the trajectory of the state.

1.1 Discrete time

We will first consider the situation in which the index set is discrete. The index is normally the time, but can be a spatial parameter as well. For simplicity we will assume that the index, $i \in \{0,\ 1,\ 2,\ ...\ N\}$, since we can always transform the problem to this.

Example: 1.1.1 (**Optimal pricing**) *Assume we have started a production of a product. Let us call it brand A. On the marked the is already a competitor product, brand B. The basic problem is to determine a price profile is such a way that we earn as much as possible. We consider the problem in a period of time and subdivide the period into a number (N say) of intervals.*

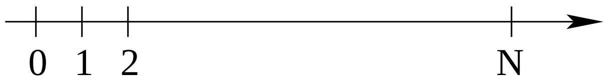

Figure 1.1. We consider the problem in a period of time divided into N intervals

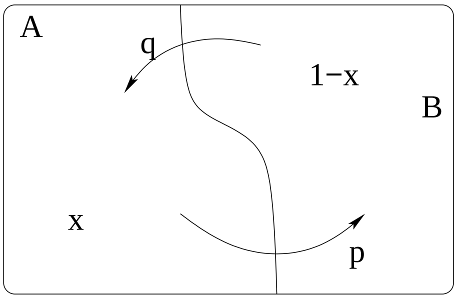

Figure 1.2. The marked shares

Let the marked share of brand A in the ith period be x_i, $i = 0, \ldots, N$ *where* $0 \le x_i \le 1$. *Since we start with no share of the marked* $x_0 = 0$. *We are seeking a sequence* u_i, $i = 0, 1, \ldots, N-1$ *of prices in order to maximize our profit. If* M *denotes the volume of the marked and* $\underline{u}$ *is production cost per units, then the performance index is*

$$J = \sum_{i=0}^{N} M x_i \big(u_i - \underline{u}\big) \tag{1.1}$$

Quite intuitively, a low price will results in a low profit, but a high share of the marked. On the other hand, a high price will give a high yield per unit but a few customers. I out simple setup, we assume that a customers i an interval is either buying brand A or brand B. In this context we can observe two kind of transitions. We will model this transition by means of probabilities.

The prices will effect the income in the present interval, but will also influence on the number of customers that will bye the brand in next interval. Let $p(u)$ *denote the probability for a customer is changing from brand A to brand B in next interval and let us denote that as the escape probability. The attraction probability is denotes as* $q(u)$. *We assume that these probabilities can be described the following logistic distribution laws:*

$$p(u) = \frac{1}{1 + exp(-k_p[u - u_p])} \qquad q(u) = \frac{1}{1 + exp(k_q[u - u_q])}$$

where k_p, u_p, k_q *and* u_q *are constants. This is illustrated as the left curve in the following plot.*

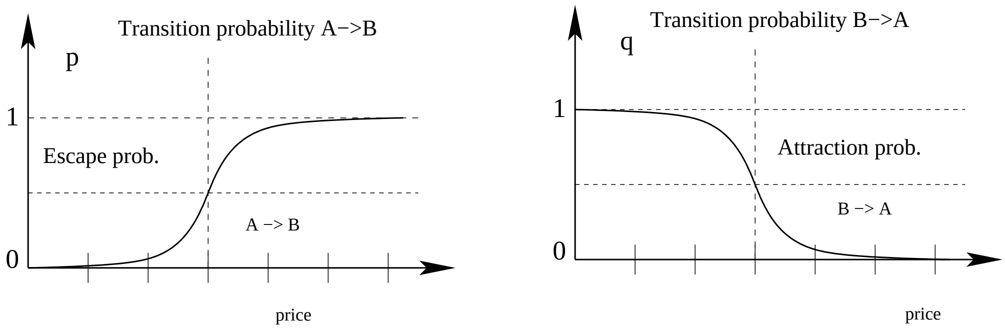

Figure 1.3. The transitions probabilities

Since $p(u_i)$ *is the probability of changing the brand from A to B,* $[1 - p(u_i)]\, x_i$ *will be the part of the customers that stays with brand A. On the other hand* $1 - x_i$ *is part of the marked buying brand B. With* $q(u_i)$ *being the probability of changing from brand B to A,* $q(u_i)\,[1 - x_i]$ *is the part of the customers who is changing from brand B to A. This results in the following dynamic model:*

Dynamics: $\qquad A \to A \qquad\qquad B \to A$

$$x_{i+1} = \big[1 - p(u_i)\big]x_i + q(u_i)\big[1 - x_i\big] \qquad x_0 = \underline{x}_0$$

or

$$x_{i+1} = q(u_i) + \big[1 - p(u_i) - q(u_i)\big]x_i \qquad x_0 = \underline{x}_0 \tag{1.2}$$

Notice, this is a discrete time model with no constraints on the decisions. The problem is determined by the objective function (1.1) and the dynamics in (1.2). The horizon N *is fixed. If we choose a constant price* $u_t = \underline{u} + 5$ ($\underline{u} = 6$,

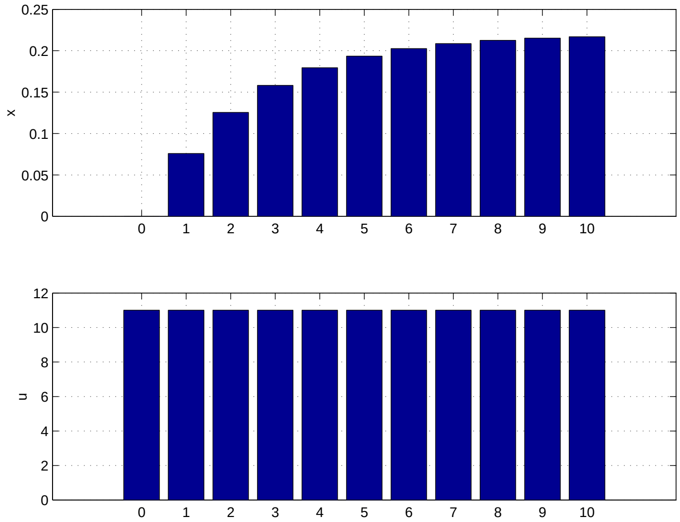

Figure 1.4. If we use a constant price $u_t = 11$ (lower panel) we will have a slow evolution of the marked share (upper panel) and a performance index equals (approx) $J = 9$.

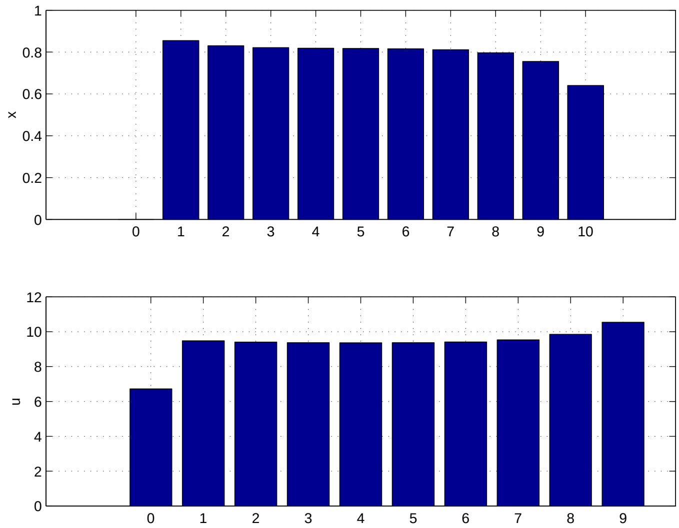

Figure 1.5. If we use an optimal pricing we will have a performance index equals (approx) $J = 26$. Notice, the introductory period as well as the final run, which is due to the final period.

$N = 10$) we get an objective equal $J = 9$ and a trajectory which can be seen in Figure 1.4. The optimal price trajectory (and path of the marked share) is plotted in Figure 1.5.

□

The example above illustrate a free (i.e. with no constraints on the decision variable or state variable) dynamic optimization problem in which we will find a input trajectory that brings the system given by the state space model:

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{1.3}$$

from the initial state, $\underline{x}_0$, in such a way that the performance index

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{1.4}$$

is optimized. Here N is fixed (given), J, ϕ and L are scalars. In general, the state vector, x_i is a n-dimensional vector, the dynamic $f_i(x_i, u_i)$ is vector (n dimensional) vector function and u_i is a (say m dimensional) vector of decisions. Also, notice there are no constraints on the decisions or the state variables (except given by the dynamics).

Example: 1.1.2 *(***Inventory Control Problem** *from (Bertsekas 1995) p. 3) Consider a problem of ordering a quantity of a certain item at each N intervals so as to meat a stochastic demand. Let us denote*

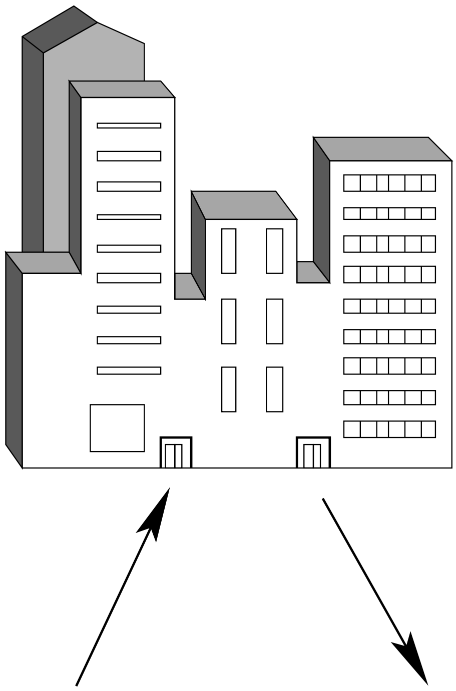

Figure 1.6. Inventory control problem

x_i *stock available at the beginning of the i'th interval.*

u_i *stock order (and immediately delivered) at the beginning of the i'th period.*

w_i *demand during the i'th interval*

We assume that excess demand is back logged and filled as soon as additional inventory becomes available. Thus, stock evolves according to the discrete time model (state space equation):

$$x_{i+1} = x_i + u_i - w_i \qquad i = 0, \ \dots\ N-1 \tag{1.5}$$

where negative stock corresponds to back logged demand. The cost incurred in period i consists of two components:

- *A cost $r(x_i)$representing a penalty for either a positive stock x_i (holding costs for excess inventory) or negative stock x_i (shortage cost for unfilled demand).*
- *The purchasing cost u_i, where c is cost per unit ordered.*

There is also a terminal cost $\phi(x_N)$ for being left with inventory x_N at the end of the N periods. Thus the total cost over N period is

$$J = \phi(x_N) + \sum_{i=0}^{N-1} (r(x_i) + cu_i) \tag{1.6}$$

We want to minimize this cost () by proper choice of the orders (decision variables) $u_0,\ u_1,\ \dots\ u_{N-1}$ *subject to the natural constraint*

$$u_i \geq 0 \qquad u = 0,\ 1,\ \dots\ N-1 \tag{1.7}$$

□

In the above example (1.1.2) we had the dynamics in (1.5), the objective function in (1.6) and some constraints in (1.7).

Example: 1.1.3 *(***Bertsekas two ovens** *from (Bertsekas 1995) page 20.) A certain material is passed through a sequence of two ovens (see Figure 1.7). Denote*

- x_0*: Initial temperature of the material*
- x_i $i = 1,\ 2$*: Temperature of the material at the exit of oven* i.
- u_i $i = 0,\ 1$*: Prevailing temperature of oven* i.

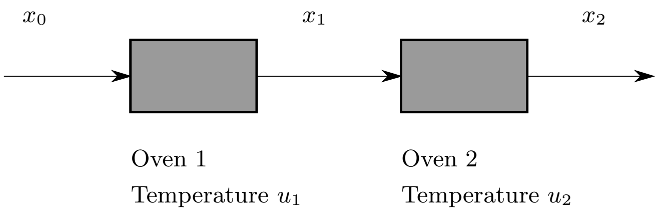

Figure 1.7. The temperature evolves according to $x_{i+1} = (1-a)x_i + au_i$ where a is a known scalar $0 < a < 1$

We assume a model of the form

$$x_{i+1} = (1-a)x_i + au_i \qquad i = 0,\ 1 \tag{1.8}$$

where a *is a known scalar from the interval* $[0,\ 1]$*. The objective is to get the final temperature* x_2 *close to a given target* T_g*, while expending relatively little energy. This is expressed by a cost function of the form*

$$J = r(x_2 - T_g)^2 + u_0^2 + u_1^2 \tag{1.9}$$

where r *is a given scalar.* □

1.2 Continuous time

In this section we will consider systems described in continuous time, i.e. when the the index, t, is continuous in the interval $[0,\ T]$. We assume the system is given in a state space formulation

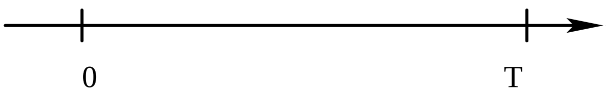

Figure 1.8. In continuous time we consider the problem for $t \in \mathbb{R}$ in the interval $[0,\ T]$

$$\dot{x} = f_t(x_t, u_t) \qquad t \in [0,\ T] \qquad x_0 = \underline{x}_0 \tag{1.10}$$

where $x_t \in \mathbb{R}^n$ is the state vector at time t, $\dot{x}_t \in \mathbb{R}^n$ is the vector of first order time derivative of the state at time t and $u_t \in \mathbb{R}^m$ is the control vector at time t. Thus, the system (1.10) consists of n coupled first order differential equations. We view x_t, $\dot{x}_t$ and u_t as column vectors and assume the system function $f : \mathbb{R}^{n\times m\times 1} \rightarrow \mathbb{R}^n$ is continuously differentiable with respect to x_t and continuous with respect to u_t.

We search for an input function (control signal, decision function) u_t, which takes the system from its original state $\underline{x}_0$ along a trajectory such that the cost function

$$J = \phi(x_T) + \int_0^T L_t(x_t, u_t)dt \tag{1.11}$$

is optimized. Here ϕ and L are scalar valued functions. The problem is specified by the functions ϕ, L and f, the initial state $\underline{x}_0$ and the length of the interval T.

Example: 1.2.1 *(***Motion control)** *from (Bertsekas 1995) p. 89). This is actually motion control in one dimension. An example in two or three dimension contains the same type of problems, but is just notationally more complicated.*

A unit mass moves on a line under influence of a force u. Let z and v be the position and velocity of the mass at times t, respectively. From a given $(z_0,\ v_0)$ we want to bring the the mass near a given final position-velocity pair $(\underline{z},\ \underline{v})$ at time T. In particular we want to minimize the cost function

$$J = (z - \underline{z})^2 + (v - \underline{v})^2 \tag{1.12}$$

subject to the control constraints

$$|u_t| \leq 1 \qquad \textit{for all} \qquad t \in [0,\ T]$$

The corresponding continuous time system is

$$\begin{bmatrix} \dot{z}_t \\ v_t \end{bmatrix} = \begin{bmatrix} v_t \\ u_t \end{bmatrix} \qquad\qquad \begin{bmatrix} z_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \underline{z}_0 \\ \underline{v}_0 \end{bmatrix} \tag{1.13}$$

We see how this example fits the general framework given earlier with

$$L_t(x_t, u_t) = 0 \qquad \phi(x_T) = (z - \underline{z})^2 + (v - \underline{v})^2$$

and the dynamic function

$$f_t(x_t, u_t) = \begin{bmatrix} v_t \\ u_t \end{bmatrix}$$

There are many variations of this problem; for example the final position andor velocity may be fixed. □

Example: 1.2.2 *(***Resource Allocation** *from (Bertsekas 1995).) A producer with production rate x_t at time t may allocate a portion u_t of his/her production to reinvestment and $1 - u_t$ to production of a storable good. Thus x_t evolves according to*

$$\dot{x}_t = \gamma u_t x_t$$

where γ is a given constant. The producer wants to maximize the total amount of product stored

$$J = \int_0^T (1 - u_t)x_t dt$$

subject to the constraint

$$0 \leq u_t \leq 1 \qquad \textit{for all} \qquad t \in [0,\ T]$$

The initial production rate x_0 is a given positive number. □

Example: 1.2.3 *(***Road Construction** *from (Bertsekas 1995)). Suppose that we want to construct a road*

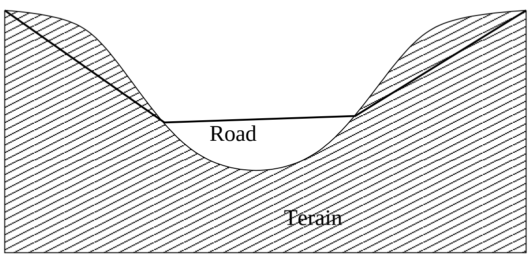

Figure 1.9. The constructed road (solid) line must lie as close as possible to the originally terrain, but must not have to high slope

over a one dimensional terrain whose ground elevation (altitude measured from some reference point) is known and is given by z_t, $t \in [0,\ T]$. Here is the index t not the time but the position along the road. The elevation of the *road is denotes as x_t, and the difference $z_t - x_i$ must be made up by fill in or excavation. It is desired to minimize the cost function*

$$J = \frac{1}{2}\int_0^T (x_t - z_t)^2 dt$$

subject to the constraint that the gradient of the road $\dot{x}$ lies between $-a$ and a, where a is a specified maximum allowed slope. Thus we have the constraint

$$|u_t| \leq a \qquad t \in [0,\ T]$$

where the dynamics is given as

$$\dot{x} = u_t$$

□

Chapter 2

Free Dynamic optimization

2.1 Discrete time free dynamic optimization

Let us in this section focus on the problem of controlling the system

$$x_{i+1} = f_i(x_i, u_i) \qquad i = 0,\ \dots\ , N-1 \qquad x_0 = \underline{x}_0 \tag{2.1}$$

such that the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{2.2}$$

is minimized. The solution to this problem is primarily a sequence of control actions or decisions, u_i, $i = 0,\ \dots\ N-1$. Secondarily (and knowing the sequence u_i, $i = 0,\ \dots\ N-1$), the solution is the path or trajectory of the state and the costate. Notice, the problem is specified by the functions f, L and ϕ, the horizon N and the initial state $\underline{x}_0$.

The problem is an optimization of (2.2) with $N+1$ set of equality constraints given in (2.1). Each set consists of n equality constraints. In the following there will be associated a vector, λ of Lagrange multipliers to each set of equality constraints. By tradition λ_{i+1} is associated to $x_{i+1} = f_i(x_i, u_i)$. These vectors of Lagrange multipliers are in the literature often denoted as costate or adjoint state.

The Hamiltonian function, which is a scalar function, is defined as

$$H_i(x_i, u_i, \lambda_{i+1}) = L_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i) \tag{2.3}$$

and facilitate a very compact formulation of the necessary conditions for an optimum.

Theorem 1: Consider the free dynamic optimization problem of bringing the system (2.1) from the initial state such that the performance index (2.2) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $i = 0,\ \dots\ , N-1$):

$x_{i+1} = f_i(x_i, u_i)$	State equation	(2.4)
$\lambda_i^T = \dfrac{\partial}{\partial x_i} H_i$	Costate equation	(2.5)
$0^T = \dfrac{\partial}{\partial u_i} H_i$	Stationarity condition	(2.6)

and the boundary conditions

$$x_0 = \underline{x}_0 \qquad \lambda_N^T = \frac{\partial}{\partial x}\phi(x_N) \tag{2.7}$$

which is a split boundary condition. □

Proof: Let λ_i, $i = 1, \dots, N$ be N vectors containing n Lagrange multiplier associated with the equality constraints in (2.1) and form the Lagrange function:

$$J_L = \phi(x_N) + \sum_{i=0}^{N-1} \Big[L_i(x_i, u_i) + \lambda_{i+1}^T \big(f_i(x_i, u_i) - x_{i+1} \big) \Big] + \lambda_0^T (\underline{x}_0 - x_0)$$

Stationarity w.r.t. λ_i gives (for $i = 1, \dots N$) as usual the equality constraints i.e. the state equations (2.4). Stationarity w.r.t. x_i gives (for $i = 0, \dots N-1$)

$$0 = \frac{\partial}{\partial x} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial x} f_i(x_i, u_i) - \lambda_i^T$$

or the costate equations (2.5), when the Hamiltonian function, (2.3), is applied. Stationarity w.r.t. x_N gives the terminal condition:

$$\lambda_N^T = \frac{\partial}{\partial x}\phi[x(N)]$$

i.e. the costate part of the boundary conditions in (2.7). Stationarity w.r.t. u_i gives the stationarity condition (for $i = 0, \dots N-1$):

$$0 = \frac{\partial}{\partial u} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial u} f_i(x_i, u_i)$$

or the stationarity condition, (2.6), when the definition, (2.3) is applied. □

The necessary condition can also be expressed in a more condensed form as

$$\boxed{x_{i+1}^T = \frac{\partial}{\partial \lambda} H_i \qquad \lambda_i^T = \frac{\partial}{\partial x} H_i \qquad 0^T = \frac{\partial}{\partial u} H_i} \tag{2.8}$$

with the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_N^T = \frac{\partial}{\partial x}\phi(x_N)$$

The Euler-Lagrange equations express the necessary conditions for optimality. The state equation (2.4) is inherently forward in time, whereas the costate equation, (2.5) is backward in time. The stationarity condition (2.6) links together the two set of recursions as indicated in Figure 2.1.

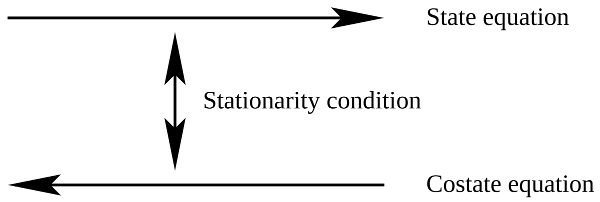

Figure 2.1. The state equation (2.4) is forward in time, whereas the costate equation, (2.5), is backward in time. The stationarity condition (2.6) links together the two set of recursions.

Example: 2.1.1 (Optimal stepping) *Consider the problem of bringing the system*

$$x_{i+1} = x_i + u_i$$

from the initial position, $\underline{x}_0$, such that the performance index

$$J = \frac{1}{2} p x_N^2 + \sum_{i=0}^{N-1} \frac{1}{2} u_i^2$$

is minimized. The Hamiltonian function is in this case

$$H_i = \frac{1}{2}u_i^2 + \lambda_{i+1}(x_i + u_i)$$

and the Euler-Lagrange equations are simply

$$x_{i+1} = x_i + u_i \tag{2.9}$$
$$\lambda_t = \lambda_{i+1} \tag{2.10}$$
$$0 = u_i + \lambda_{i+1} \tag{2.11}$$

with the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_N = p x_N$$

These equations are easily solved. Notice, the costate equation (2.10) gives the key to the solution. Firstly, we notice that the costate are constant. Secondly, from the boundary condition we have:

$$\lambda_i = p x_N$$

From the Euler equation or the stationarity condition, (2.11), we can find the control sequence (expressed as function of the terminal state x_N), which can be introduced in the state equation, (2.9). The results are:

$$u_i = -p x_N \qquad x_i = x_0 - i p x_N$$

From this, we can determine the terminal state as:

$$x_N = \frac{1}{1+Np} x_0$$

Consequently, the solution to the dynamic optimization problem is given by:

$$u_i = -\frac{p}{1+Np}x_0 \qquad \lambda_i = \frac{p}{1+Np}x_0 \qquad x_i = \frac{1+(N-i)p}{1+Np}x_0 = x_0 - i\frac{p}{1+Np}x_0$$

□

Example: 2.1.2 (simple LQ problem). *Let us now focus on a slightly more complicated problem of bringing the linear, first order system given by:*

$$x_{i+1} = a x_i + b u_i \qquad x_0 = \underline{x}_0$$

along a trajectory from the initial state, such the cost function:

$$J = \frac{1}{2}p x_N^2 + \sum_{i=0}^{N-1} \frac{1}{2} q x_i^2 + \frac{1}{2} r u_i^2$$

is minimized. Notice, this is a special case of the LQ problem, which is solved later in this chapter.

The Hamiltonian for this problem is

$$H_i = \frac{1}{2} q x_i^2 + \frac{1}{2} r u_i^2 + \lambda_{i+1}\left[a x_i + b u_i\right]$$

and the Euler-Lagrange equations are:

$$x_{i+1} = a x_i + b u_i \tag{2.12}$$
$$\lambda_i = q x_i + a\lambda_{i+1} \tag{2.13}$$
$$0 = r u_i + \lambda_{i+1} b \tag{2.14}$$

which has the two boundary conditions

$$x_0 = \underline{x}_0 \qquad \lambda_N = p x_N$$

The stationarity conditions give us a sequence of decisions

$$u_i = -\frac{b}{r}\lambda_{i+1} \tag{2.15}$$

if the costate is known.

Inspired from the boundary condition on the costate we will postulate a relationship between the state and the costate as:

$$\lambda_i = s_i x_i \tag{2.16}$$

If we insert (2.15) and (2.16) in the state equation, (2.12), we can find a recursion for the state

$$x_{i+1} = a x_i - \frac{b^2}{r} s_{i+1} x_{i+1}$$

or

$$x_{i+1} = \frac{1}{1 + \frac{b^2}{r} s_{i+1}} a x_i$$

From the costate equation, (2.13), we have

$$s_i x_i = q x_i + a s_{i+1} x_{i+1} = \left[q + a s_{i+1} \frac{1}{1 + \frac{b^2}{r} s_{i+1}} a\right] x_i$$

which has to fulfilled for any x_i. This is the case if s_i is given by the back wards recursion

$$s_i = a s_{i+1} \frac{1}{1 + \frac{b^2}{r} s_{i+1}} a + q$$

or if we use identity $\frac{1}{1+x} = 1 - \frac{x}{1+x}$

$$s_i = q + s_{i+1} a^2 - \frac{(a b s_{i+1})^2}{r + b^2 s_{i+1}} \qquad s_N = p \tag{2.17}$$

where we have introduced the boundary condition on the costate. Notice the sequence of s_i can be determined by solving back wards starting in $s_N = p$ (where p is specified by the problem).

With this solution (the sequence of s_i) we can determine the (sequence of) costate and control actions

$$u_i = -\frac{b}{r} \lambda_{i+1} = -\frac{b}{r} s_{i+1} x_{i+1} = -\frac{b}{r} s_{i+1} (a x_i + b u_i)$$

or

$$u_i = -\frac{a b s_{i+1}}{r + b^2 s_{i+1}} x_i \qquad \text{and for the costate} \qquad \lambda_i = s_i x_i$$

□

Example: 2.1.3 (Discrete Velocity Direction Programming for Max Range). *From (Bryson 1999). This is a variant of the* **Zermelo problem**.

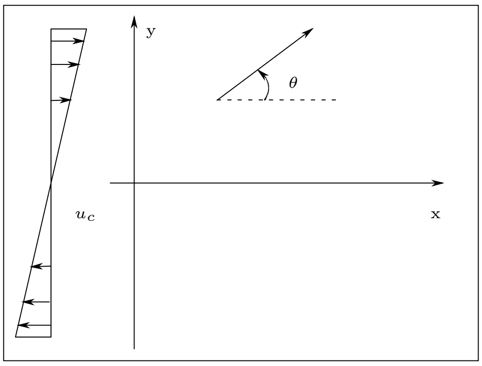

Figure 2.2. Geometry for the Zermelo problem

A ship travels with constant velocity with respect to the water through a region with current. The velocity of the current is parallel to the x-axis but varies with y, so that

$$\begin{aligned} \dot{x} &= V\ cos(\theta) + u_c(y) \qquad x_0 = 0 \\ \dot{y} &= V\ sin(\theta) \qquad y_0 = 0 \end{aligned}$$

where θ is the heading of the ship relative to the x-axis. The ship starts at origin and we will maximize the range in the direction of the x-axis.

Assume that

$$u_c = \beta y$$

and that θ is constant for time intervals of length $h = T/N$. Here T is the length of the horizon and N is the number of intervals.

The system is in discrete time described by

$$\begin{aligned} x_{i+1} &= x_i + V h\ cos(\theta_i) + \beta\left[h y_i + \frac{1}{2} V h^2\ sin(\theta_i)\right] \\ y_{i+1} &= y_i + V h\ sin(\theta_i) \end{aligned} \tag{2.18}$$

(found from the continuous time description by integration). The objective is to maximize the final position in the direction of the x-axis i.e. to maximize the performance index

$$J = x_N \tag{2.19}$$

Notice, the L term n the performance index is zero, but $\phi_N = x_N$.

Let us introduce a costate sequence for each of the states, i.e. $\lambda = \begin{bmatrix} \lambda_i^x & \lambda_i^y \end{bmatrix}^T$. *Then the Hamiltonian function is given by*

$$H_i = \lambda_{i+1}^x \big[x_i + Vh\ cos(\theta_i) + \beta\big(hy_i + \frac{1}{2}Vh^2 sin(\theta_i)\big)\big] + \lambda_{i+1}^y \big[y_i + Vh\ sin(\theta_i)\big]$$

The Euler -Lagrange equations gives us the state equations, (2.19), and the costate equations

$$\begin{aligned} \lambda_i^x &= \frac{\partial}{\partial x} H_i = \lambda_{i+1}^x \qquad \lambda_N^x = 1 \\ \lambda_i^y &= \frac{\partial}{\partial y} H_i = \lambda_{i+1}^y + \lambda_{i+1}^x \beta h \qquad \lambda_N^y = 0 \end{aligned} \tag{2.20}$$

and the stationarity condition:

$$0 = \frac{\partial}{\partial u} H_i = \lambda_{i+1}^x \big[-Vh\ sin(\theta_i) + \frac{1}{2}\beta Vh^2\ cos(\theta_i)\big] + \lambda_{i+1}^y Vh\ cos(\theta_i) \tag{2.21}$$

The costate equation, (2.21), has a quite simple solution

$$\lambda_i^x = 1 \qquad\qquad \lambda_i^y = (N-i)\beta h$$

which introduced in the stationarity condition, (2.21), gives us

$$0 = -Vh\ sin(\theta_i) + \frac{1}{2}\beta Vh^2\ cos(\theta_i) + (N-1-i)\beta Vh^2\ cos(\theta_i)$$

or

$$\boxed{tan(\theta_i) = (N - i - \frac{1}{2})\beta h} \tag{2.22}$$

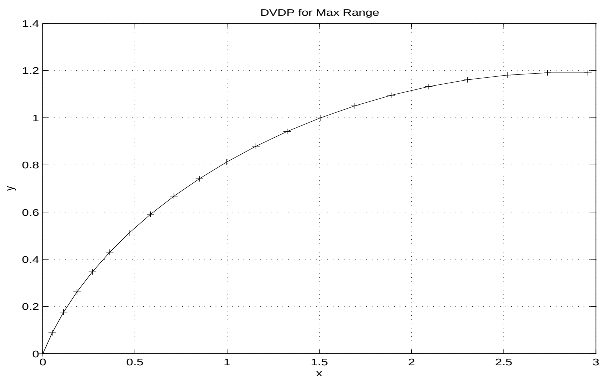

Figure 2.3. DVDP for Max Range with $u_c = \beta y$

□

Example: 2.1.4 (Discrete Velocity Direction Programming with Gravity). *From (Bryson 1999). This is a variant of the* **Brachistochrone problem**.

A mass m moves in a constant force field of magnitude g starting at rest. We shall do this by programming the direction of the velocity, i.e. the angle of the wire below the horizontal, θ_i *as a function of the time. It is desired to find the path that maximize the horizontal range in given time T.*

This is the dual problem to the famous Brachistochrone problem of finding the shape of a wire to minimize the time T to cover a horizontal distance (brachistocrone means shortest time in Greek). It was posed and solved by Jacob Bernoulli in the seventh century (more precisely in 1696).

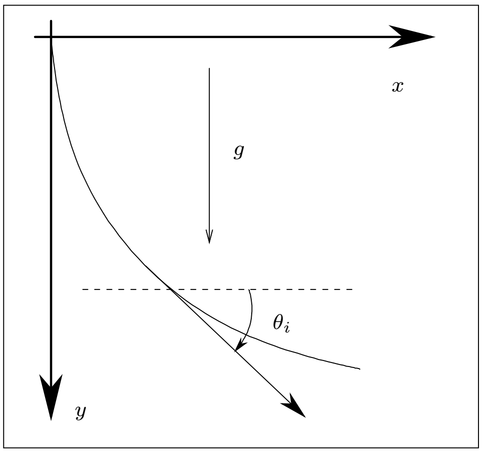

Figure 2.4. Nomenclature for the Velocity Direction Programming Problem

To treat this problem i discrete time we assume that the angle is kept constant in intervals of length $h = T/N$. *A little geometry results in an acceleration along the wire is*

$$a_i = g\ sin(\theta_i)$$

Consequently, the speed along the wire is

$$v_{i+1} = v_i + gh\ sin(\theta_i)$$

and the increment in traveling distance along the wire is

$$l_i = v_i h + \frac{1}{2} gh^2\ sin(\theta_i) \tag{2.23}$$

The position of the bead is then given by the recursion

$$x_{i+1} = x_i + l_i\ cos(\theta_i)$$

Let the state vector be $s_i = \begin{bmatrix} v_i & x_i \end{bmatrix}^T$.

The problem is then find the optimal sequence of angles, θ_i *such that system*

$$\begin{bmatrix} v \\ x \end{bmatrix}_{i+1} = \begin{bmatrix} v_i + gh\ sin(\theta_i) \\ x_i + l_i\ cos(\theta_i) \end{bmatrix} \begin{bmatrix} v \\ x \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{2.24}$$

such that performance index

$$J = \phi_N(s_N) = x_N \tag{2.25}$$

is minimized.

Let us introduce a costate or an adjoint state to each of the equations in dynamic, i.e. let $\lambda_i = \begin{bmatrix} \lambda_i^v & \lambda_i^x \end{bmatrix}^T$. *Then the Hamiltonian function becomes*

$$H_i = \lambda_{i+1}^v \big[v_i + gh\ sin(\theta_i)\big] + \lambda_{i+1}^x \big[x_i + l_i\ cos(\theta_i)\big]$$

The Euler-Lagrange equations give us the state equation, (2.24), the costate equations

$$\lambda_i^v = \frac{\partial}{\partial v} H_i = \lambda_{i+1}^v + \lambda_{i+1}^x h\ cos(\theta_i) \qquad \lambda_N^v = 0 \tag{2.26}$$

$$\lambda_i^x = \frac{\partial}{\partial x} H_i = \lambda_{i+1}^x \qquad \lambda_N^x = 1 \tag{2.27}$$

and the stationarity condition

$$0 = \frac{\partial}{\partial u} H_i = \lambda_{i+1}^v gh\ cos(\theta_i) + \lambda_{i+1}^x \big[-l_i\ sin(\theta_i) + cos(\theta_i) \frac{1}{2} gh^2\ cos(theta_i)\big] \tag{2.28}$$

The solution to the costate equation (2.27) is simply $\lambda_i^x = 1$ *which reduce the set of equations to the state equation, (2.24), and*

$$\lambda_i^v = \lambda_{i+1}^v + gh\ cos(\theta_i) \qquad \lambda_N^v = 0$$

$$0 = \lambda_{i+1}^v gh\ cos(\theta_i) - l_i\ sin(\theta_i) + \frac{1}{2} gh^2\ cos(\theta_i)$$

The solution to this two point boundary value problem can be found using several trigonometric relations. If $\alpha = \frac{1}{2}\pi/N$ the solution is for $i = 0, \; ... \; N-1$

$$\theta_i = \frac{\pi}{2} - \alpha(i + \frac{1}{2})$$
$$v_i = \frac{gT}{2Nsin(\alpha/2)} sin(\alpha i)$$
$$x_i = \frac{cos(\alpha/2)gT^2}{4Nsin(\alpha/2)} \Big[i - \frac{sin(2\alpha i)}{2sin(\alpha)} \Big]$$
$$\lambda_i^v = \frac{cos(\alpha i)}{2Nsin(\alpha/2)}$$

Notice, the y coordinate did not enter the problem in this presentation. It could have included or found from simple kinematics that

$$y_i = \frac{cos(\alpha/2)gT^2}{8N^2 sin(\alpha/2) sin(\alpha)} \big[1 - cos(2\alpha i) \big]$$

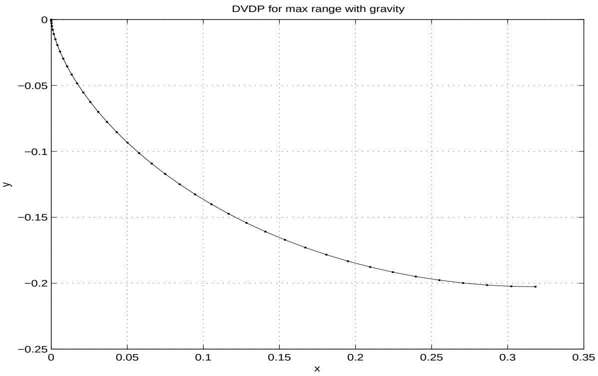

Figure 2.5. DVDP for Max range with gravity for $N = 40$.

□

2.2 The LQ problem

In this section we will deal with the problem of finding an optimal input sequence, $u_i, \; i = 0, \; ... \; N-1$ that take the Linear system

$$x_{i+1} = Ax_i + Bu_i \qquad\qquad x_0 = \underline{x}_0 \tag{2.29}$$

from its original state, $\underline{x}_0$, such that the Qadratic cost function

$$J = \frac{1}{2} x_N^T P x_N + \sum_{i=0}^{N-1} \Big(\frac{1}{2} x_i^T Q x_i + \frac{1}{2} u_i^T R u_i \Big) \tag{2.30}$$

is minimized.

In this case the Hamiltonian function is

$$H_i = \frac{1}{2} x_i^T Q x_i + \frac{1}{2} u_i^T R u_i + \lambda_{i+1}^T \Big[Ax_i + Bu_i \Big]$$

and the Euler-Lagrange equation becomes:

$$x_{i+1} = Ax_i + Bu_i \tag{2.31}$$
$$\lambda_i = Qx_i + A^T \lambda_{i+1} \tag{2.32}$$
$$0 = Ru_i + B^T u_i \tag{2.33}$$

with the (split) boundary conditions

$$x_0 = \underline{x}_0 \qquad\qquad \lambda_N = Px_N$$

Theorem 2: The optimal solution to the free LQ problem specified by (2.29) and (2.30) is given by a state feed back

$$\boxed{u_i = -K_i x_i} \tag{2.34}$$

where the time varying gain is given by

$$K_i = \left[R + B^T S_{i+1} B\right]^{-1} B^T S_{i+1} A \tag{2.35}$$

Here the matrix, S, is found from the following back wards recursion

$$\boxed{S_i = A^T S_{i+1} A - A^T S_{i+1} B \left(B^T S_{i+1} B + R\right)^{-1} B^T S_{i+1} A + Q \qquad S_N = P} \tag{2.36}$$

which is denoted as the (discrete time, control) *Riccati equation*. □

Proof: From the stationarity condition, (2.33), we have

$$u_i = -R^{-1} B^T \lambda_{i+1} \tag{2.37}$$

As in example 2.1.2 we will use the costate boundary condition and guess on a relation between costate and state

$$\lambda_i = S_i x_i \tag{2.38}$$

If (2.38) and (2.37) are introduced in (2.5) we find the evolution of the state

$$x_i = Ax_i - BR^{-1} B^T S_{i+1} x_{i+1}$$

or if we solves for x_{i+1}

$$x_{i+1} = \left[I + BR^{-1} B^T S_{i+1}\right]^{-1} Ax_i \tag{2.39}$$

If (2.38) and (2.39) are introduced in the costate equation, (2.6)

$$\begin{aligned} S_i x_i &= Qx_i + A^T S_{i+1} x_{i+1} \\ &= Qx_i + A^T S_{i+1} \left[I + BR^{-1} B^T S_{i+1}\right]^{-1} Ax_i \end{aligned}$$

Since this equation has to be fulfilled for any x_t, the assumption (2.38) is valid if we can determine the sequence S_i from

$$S_i = A^\top S_{i+1} \left(I + BR^{-1} B^\top S_{i+1}\right)^{-1} A + Q$$

If we use the inversion lemma (A.50) we can substitute

$$\left(I + BR^{-1} B^\top S_{i+1}\right)^{-1} = I - B\left(B^T S_{i+1} B + R\right)^{-1} B^T S_{i+1}$$

and the recursion for S becomes

$$S_i = A^T S_{i+1} A - A^T S_{i+1} B \left(B^T S_{i+1} B + R\right)^{-1} B^T S_{i+1} A + Q \tag{2.40}$$

The recursion is a backward recursion starting in

$$S_N = P$$

For determine the control action we have (2.37) or with (2.38) inserted

$$\begin{aligned} u_i &= -R^{-1} B^T S_{i+1} x_{i+1} \\ &= -R^{-1} B^T S_{i+1} (Ax_i + Bu_i) \end{aligned}$$

or

$$u_i = -\left[R + B^T S_{i+1} B\right]^{-1} B^T S_{i+1} Ax_i \tag{2.41}$$

□

The matrix equation, (2.36), is denoted as the **Riccati** equation, after Count Riccati, an Italian who investigated a scalar version in 1724.

It can be shown (see e.g. (Lewis 1986*a*) p. 54) that the optimal cost function achieved the value

$$J^* = V_o(x_o) = x_0^T S_0 x_o \tag{2.42}$$

i.e. is quadratic in the initial state and S_0 is a measure of the curvature in that point.

2.3 Continuous free dynamic optimization

Consider the problem related to finding the input function u_t to the system

$$\dot{x} = f_t(x_t, u_t) \qquad x_0 = \underline{x}_0 \qquad t \in [0,\ T] \tag{2.43}$$

such that the cost function

$$J = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt \tag{2.44}$$

is minimized. Here the initial state $\underline{x}_0$ and final time T are given (fixed). The problem is specified by the dynamic function, f_t, the scalar value functions ϕ and L and the constants T and $\underline{x}_0$.

The problem is an optimization of (2.44) with continuous equality constraints. Similarilly to the situation in discret time, we here associate a n-dimensional function, λ_t, to the equality constraints, $\dot{x} - f_t(x_t, u_t)$. Also in continuous time these multipliers are denoted as Costate or adjoint state. In some part of the litterature the vector function, λ_t, is denoted as *influence function*.

For convienence we can introduce the scalar Hamiltonian function as follows:

$$H_t(x_t, u_t, \lambda_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) \tag{2.45}$$

We are now able to give the necessary condition for the solution to the problem.

Theorem 3: Consider the free dynamic optimization problem in continuous time of bringing the system (2.43) from the initial state such that the performance index (2.44) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $t \in [0,\ T]$):

$$\dot{x}_t = f_t(x_t, u_t) \qquad \text{State equation} \tag{2.46}$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \text{Costate equation} \tag{2.47}$$

$$0^T = \frac{\partial}{\partial u_t} H_t \qquad \text{stationarity condition} \tag{2.48}$$

and the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_T = \frac{\partial}{\partial x}\phi_T(x_T) \tag{2.49}$$

□

Proof: Before we start on the proof we need two lemmas. The first one is the fundamental Lemma of calculus of variation, while the second is Leibniz's rule.

Lemma 1: **(The Fundamental lemma of calculus of variations)** Let h_t be a continuous real-values function defined on $a \le t \le b$ and suppose that:

$$\int_a^b h_t \delta_t \; dt = 0$$

for <u>any</u> $\delta_t \in C^2[a,b]$ satisfying $\delta_a = \delta_b = 0$. Then

$$h_t \equiv 0 \qquad t \in [a,b]$$

□

Lemma 2: **(Leibniz's rule for functionals):** Let $x_t \in \mathbb{R}^n$ be a function of $t \in \mathbb{R}$ and

$$J(x) = \int_s^T h_t(x_t) dt$$

where both J and h are functions of x_t (i.e. functionals). Then

$$dJ = h_T(x_T)dT - h_s(x_s)ds + \int_s^T \frac{\partial}{\partial x} h_t(x_t) \delta x \; dt$$

□

Firstly, we construct the Lagrange function:

$$J_L = \phi_T(x_T) + \int_0^T L_t(x_t, u_t) dt + \int_0^T \lambda_t^T \left[f_t(x_t, u_t) - \dot{x}_t \right] dt$$

Then we introduce integration by part

$$\int_0^T \lambda_t^T \dot{x}_t dt + \int_0^T \dot{\lambda}_t^T x_t = \lambda_T^T x_T - \lambda_0^T x_0$$

in the Lagrange function which results in:

$$J_L = \phi_T(x_T) + \lambda_0^T x_0 - \lambda_T^T x_T + \int_0^T \left(L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) + \dot{\lambda}_t^T x_t \right) dt$$

Using Leibniz rule (Lemma 2) the variation in J_L w.r.t. x, λ and u is:

$$\begin{aligned} dJ_L &= \left(\frac{\partial}{\partial x_T} \phi_T - \lambda_T^T \right) dx_T + \int_0^T \left(\frac{\partial}{\partial x} L + \lambda^T \frac{\partial}{\partial x} f + \dot{\lambda}^T \right) \delta x \; dt \\ &\quad + \int_0^T \left(f_t(x_t, u_t) - \dot{x}_t \right)^T \delta\lambda \; dt + \int_0^T \left(\frac{\partial}{\partial u} L + \lambda^T \frac{\partial}{\partial u} f \right) \delta u \; dt \end{aligned}$$

According to optimization with equality constraints the necessary condition is obtained as a stationary point to the Lagrange function. Setting to zero all the coefficients of the independent increments yields necessary condition as given in Theorem 3. □

We can express the necessary conditions as

$$\boxed{\dot{x}^T = \frac{\partial}{\partial \lambda} H \qquad -\dot{\lambda}^T = \frac{\partial}{\partial x} H \qquad 0^T = \frac{\partial}{\partial u} H} \tag{2.50}$$

with the (split) boundary conditions

$$x_0 = \underline{x}_0 \qquad \lambda_T^T = \frac{\partial}{\partial x} \phi_T$$

Furthermore, we have

$$\begin{aligned} \dot{H} &= \frac{\partial}{\partial t} H + \frac{\partial}{\partial u} H \dot{u} + \frac{\partial}{\partial x} H \dot{x} + \frac{\partial}{\partial \lambda} H \dot{\lambda} \\ &= \frac{\partial}{\partial t} H + \frac{\partial}{\partial u} H \dot{u} + \frac{\partial}{\partial x} H f + f^T \dot{\lambda} \\ &= \frac{\partial}{\partial t} H + \frac{\partial}{\partial u} H \dot{u} + \left[\frac{\partial}{\partial x} H + \dot{\lambda}^T \right] f \\ &= \frac{\partial}{\partial t} H \end{aligned}$$

Now, in the time invariant case, where f and L are not explicit functions of t, and so neither is H. In this case

$$\dot{H} = 0 \tag{2.51}$$

Hence, for time invariant systems and cost functions, the Hamiltonian is a constant on the optimal trajectory.

Example: 2.3.1 (Motion Control) *Let us consider the continuous time version of example 2.1.1. The problem is to bring the system*

$$\dot{x} = u_t \qquad x_0 = \underline{x}_0$$

from the initial position, $\underline{x}_0$, such that the performance index

$$J = \frac{1}{2} p x_T^2 + \int_0^T \frac{1}{2} u^2 dt$$

is minimized. The Hamiltonian function is in this case

$$H = \frac{1}{2} u^2 + \lambda u$$

and the Euler-Lagrange equations are simply

$$\begin{aligned} \dot{x} &= u_t & \qquad x_0 = \underline{x}_0 \\ -\dot{\lambda} &= 0 & \qquad \lambda_T = p x_T \\ 0 &= u + \lambda & \end{aligned}$$

These equations are easily solved. Notice, the costate equation here gives the key to the solution. Firstly, we notice that the costate is constant. Secondly, from the boundary condition we have:

$$\lambda = p x_T$$

From the Euler equation or the stationarity condition we find the control signal (expressed as function of the terminal state x_T) is given as

$$u = -p x_T$$

If this strategy is introduced in the state equation we have

$$x_t = x_0 - p x_T \; t$$

from which we get

$$x_T = \frac{1}{1 + pT} x_0$$

Finally, we have

$$x_t = \left(1 - \frac{p}{1 + pT} \; t \right) x_0 \qquad u_t = -\frac{p}{1 + pT} x_0 \qquad \lambda = \frac{p}{1 + pT} x_0$$

It is also quite simple to see, that the Hamiltonian function is constant and equal

$$H = -\frac{1}{2} \left[\frac{p}{1 + pT} x_0 \right]^2$$

□

Chapter 3

Dynamic optimization with end points constraints

In this chapter we will investigate the situation in which there is constraints on the final states. We will focus on equality constraints on (some of) the terminal states, i.e.

$$\psi_N(x_N) = 0 \qquad \text{(in discrete time)} \tag{3.1}$$

or

$$\psi_T(x_T) = 0 \qquad \text{(in continuous time)} \tag{3.2}$$

where ψ is a mapping from $\mathbb{R}^n$ to $\mathbb{R}^p$ and $p \leq n$, i.e. not fewer states than constraints.

3.1 Simple terminal constraints

Consider the discrete time system (for $i = 0,\ 1,\ \dots\ N-1$)

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{3.3}$$

the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{3.4}$$

and the simple terminal constraints

$$x_N = \underline{x}_N \tag{3.5}$$

where $\underline{x}_N$ (and $\underline{x}_0$) is given. In this simple case, the terminal contribution, ϕ, to the performance index could be omitted, since it has not effect on the solution (except a constant additive term to the performance index). The problem consist in bringing the system (3.3) from its initial state $\underline{x}_0$ to a (fixed) terminal state $\underline{x}_N$ such that the performance index, (3.4) is minimized.

The problem is specified by the functions f and L (and ϕ), the length of the horizon N and by the initial and terminal state $\underline{x}_0$, $\underline{x}_N$. Let us apply the usual notation and associate a vector of Lagrange multipliers λ_{i+1} to each of the equality constraints $x_{i+1} = f_i(x_i, u_i)$. To the terminal constraint we associate, ν which is a vector containing n (scalar) Lagrange multipliers.

Notice, as in the unconstrained case we can introduce the Hamiltonian function

$$H_i(x_i, u_i, \lambda_{i+1}) = L_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i)$$

and obtain a much more compact form for necessary conditions, which is stated in the theorem below.

Theorem 4: Consider the dynamic optimization problem of bringing the system (3.3) from the initial state, $\underline{x}_0$, to the terminal state, $\underline{x}_N$, such that the performance index (3.4) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $i = 0, \ \dots \ , N-1$):

$$x_{i+1} = f_i(x_i, u_i) \qquad \text{State equation} \tag{3.6}$$

$$\lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad \text{Costate equation} \tag{3.7}$$

$$0^T = \frac{\partial}{\partial u} H_i \qquad \text{Stationarity condition} \tag{3.8}$$

The boundary conditions are

$$x_0 = \underline{x}_0 \qquad x_N = \underline{x}_N$$

and the Lagrange multiplier, ν, related to the simple equality constraints is can be determined from

$$\lambda_N^T = \nu^T + \frac{\partial}{\partial x_N}\phi$$

□

Notice, ther performance index will rarely have a dependence on the terminal state in this situation. In that case

$$\lambda_N^T = \nu^T$$

Also notice, the dynamic function can be expressed in terms of the Hamiltonian function as

$$f_i^T(x_i, u_i) = \frac{\partial}{\partial \lambda} H_i$$

and obtain a more memotechnical form

$$x_{i+1}^T = \frac{\partial}{\partial \lambda} H_i \qquad \lambda_{i+1}^T = \frac{\partial}{\partial x} H_i \qquad 0^T = \frac{\partial}{\partial u} H_i$$

for the Euler-Lagrange equations, (3.6)-(3.8).

Proof: We start forming the Lagrange function:

$$J_L = \phi(x_N) + \sum_{i=0}^{N-1} \Big[L_i(x_i, u_i) + \lambda_{i+1}^T \big(f_i(x_i, u_i) - x_{i+1} \big) \Big] + \lambda_0^T (\underline{x}_0 - x_0) + \nu^T (x_N - \underline{x}_N)$$

As in connection to free dynamic optimization stationarity w.r.t.. λ_{i+1} gives (for $i = 0, \ \dots \ N-1$) the state equations (3.6). In the same way stationarity w.r.t. ν gives

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

Stationarity w.r.t. x_i gives (for $i = 1, \ \dots \ N-1$)

$$0^T = \frac{\partial}{\partial x} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial x} f_i(x_i, u_i) - \lambda_i^T$$

or the costate equations (3.7) if the definition of the Hamiltonian function is applied. For $i = N$ we have

$$\lambda_N^T = \nu^T + \frac{\partial}{\partial x_N}\phi$$

Stationarity w.r.t. u_i gives (for $i = 0, \ \dots \ N-1$):

$$0^T = \frac{\partial}{\partial u} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial u} f_i(x_i, u_i)$$

or the stationarity condition, (3.8), if the Hamiltonian function is introduced. □

Example: 3.1.1 (Optimal stepping) *Let us return to the system from 2.1.1, i.e.*

$$x_{i+1} = x_i + u_i$$

The task is to bring the system from the initial position, $\underline{x}_0$ *to a given final position,* x_N*, in a fixed number,* N *of steps, such that the performance index*

$$J = \sum_{i=0}^{N-1} \frac{1}{2} u_i^2$$

is minimized. The Hamiltonian function is in this case

$$H_i = \frac{1}{2} u_i^2 + \lambda_{i+1}(x_i + u_i)$$

and the Euler-Lagrange equations are simply

$$x_{i+1} = x_i + u_i \tag{3.9}$$
$$\lambda_t = \lambda_{i+1} \tag{3.10}$$
$$0 = u_i + \lambda_{i+1} \tag{3.11}$$

with the boundary conditions:

$$x_0 = \underline{x}_0 \qquad\qquad x_N = \underline{x}_N$$

Firstly, we notice that the costates are constant, i.e.

$$\lambda_i = c$$

Secondly, from the stationarity condition we have:

$$u_i = -c$$

and inserted in the state equation (3.9)

$$x_i = x_0 - ic \qquad \textit{and finally} \qquad x_N = x_0 - Nc$$

From the latter equation and boundary condition we can determine the constant to be

$$c = \frac{\underline{x}_0 - \underline{x}_N}{N}$$

Notice, the solution to the problem in Example 2.1.1 tens to this for $p \rightarrow \infty$ *and* $\underline{x}_N = 0$.

Also notice, the Lagrange multiplier to the terminal conditions is equal

$$\nu = \lambda_N = c = \frac{\underline{x}_0 - \underline{x}_N}{N}$$

and have an interpretation as a shadow price. □

Example: 3.1.2 Investment planning. *Suppose we are planning to invest some money during a period of time with* N *intervals in order to save a specific amount of money* $\underline{x}_N = 10000\$$. *If the the bank pays interest with rate* α *in one interval, the account balance will evolve according to*

$$x_{i+1} = (1+\alpha)x_i + u_i \qquad x_0 = 0 \tag{3.12}$$

Here u_i *is the deposit per period. This problem could easily be solved by the plan* $u_i = 0$ $i = 1, \ldots N-1$ *and* $u_{N-1} = \underline{x}_N$. *The plan might, however, be a little beyond our means. We will be looking for a minimum effort plan. This could be achieved if the deposits are such that the performance index:*

$$J = \sum_{i=0}^{N-1} \frac{1}{2} u_i^2 \tag{3.13}$$

is minimized.

In this case the Hamiltonian function is

$$H_i = \frac{1}{2} u_i^2 + \lambda_{i+1} \left((1+\alpha)x_i + u_i\right)$$

and the Euler-Lagrange equations become

$$\begin{aligned} x_{i+1} &= (1+\alpha)x_i + u_i \qquad x_0 = 0 \qquad x_N = 10000 && (3.14)\\ \lambda_i &= (1+\alpha)\lambda_{i+1} \qquad\qquad\qquad \nu = \lambda_N && (3.15)\\ 0 &= u_i + \lambda_{i+1} && (3.16) \end{aligned}$$

In this example we are going to solve this problem by means of analytical solutions. In example 3.1.3 we will solved the problem in a more computer oriented way.

Introduce the notation $a = 1+\alpha$ and $q = \frac{1}{a}$. From the Euler-Lagrange equations, or rather the costate equation (3.15), we find quite easily that

$$\lambda_{i+1} = q\lambda_i \qquad or \qquad \lambda_i = c\ q^i$$

where c is an unknown constant. The deposit is then (according to (3.16)) given as

$$u_i = -c\ q^{i+1}$$

$$\begin{aligned}
x_0 &= 0 \\
x_1 &= -c\ q \\
x_2 &= a(-c\ q) - cq^2 = -acq - cq^2 \\
x_3 &= a(-acq - cq^2) - cq^3 = -a^2cq - acq^2 - cq^3 \\
&\vdots \\
x_i &= -a^{i-1}cq - a^{i-2}cq^2 - \ \dots\ - cq^i = -c\sum_{k=1}^{i} a^{i-k}q^k \qquad 0 \le i \le N
\end{aligned}$$

The last part is recognized as a geometric series and consequently

$$x_i = -cq^{2-i}\frac{1-q^{2i}}{1-q^2} \qquad 0 \le i \le N$$

For determination of the unknown constant c we have

$$\underline{x}_N = -c\ q^{2-N}\frac{1-q^{2N}}{1-q^2}$$

When this constant is known we can determine the sequence of annual deposit and other interesting quantities such as the state (account balance) and the costate. The first two is plotted in Figure 3.1.

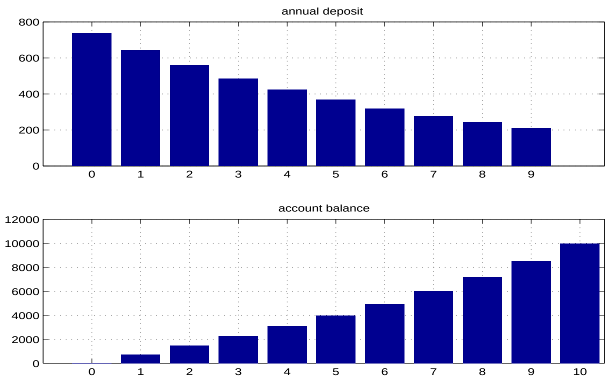

Figure 3.1. Investment planning. Upper panel show the annual deposit and the lower panel shows the account balance.

□

Example: 3.1.3 *In this example we will solve the investment planning problem from example 3.1.2 in a more computer oriented way. We will use a so called shooting method, which in this case is based on the fact that the costate equation can be reversed. As in the previous example (example 3.1.2) the key to the problem is the initial value of the costate (the unknown constant c in example 3.1.2).*

Consider the Euler-Lagrange equations in example 3.1.3. If $\lambda_0 = c$ is known, then we can determine λ_1 and u_0 from (3.15) and (3.16). Now, since x_0 is known we use the state equation and determine x_1. Further on, we can use (3.15) and (3.16) again and determine λ_2 and u_1. In this way we can iterate the solution until $i = N$. This is what is implemented in the file difference.m (see Table 3.1. If the constant c is correct then $x_N - \underline{x}_N = 0$.

The Matlab command fsolve is an implementation of a method for finding roots in a nonlinear function. For example the command(s)

```
alfa=0.15; x0=0; xN=10000; N=10;
opt=optimset('fsolve');

c=fsolve(@difference,-800,opt,alfa,x0,xN,N)
```

```
function deltax=difference(c,alfa,x0,xN,N)
lambda=c; x=x0;
for i=0:N-1,
 lambda=lambda/(1+alfa);
 u=-lambda;
 x=(1+alfa)*x+u;
end
deltax=(x-xN);
```

Table 3.1. The contents of the file, difference.m

will search for the correct value of c starting with -800. *The value of the parameters* `alfa,x0,xN,N` *is just passing through to difference.m*

3.2 Simple partial end point constraints

Consider a variation of the previously treated simple problem. Assume some of the terminal state variable, $\tilde{x}_N$, is constrained i a simple way and the rest of the variable, $\bar{x}_N$, is not constrained, i.e.

$$x_N = \begin{bmatrix} \tilde{x}_N \\ \bar{x}_N \end{bmatrix} \qquad \tilde{x}_N = \underline{\tilde{x}}_N$$

The rest of the state variable, $\bar{x}_N$, might influence the terminal contribution, $\phi_N(x_N)$. Assume for simplicity that $\tilde{x}_N$ do not influence on ϕ_N, then $\phi_N(x_N) = \phi_N(\bar{x}_N)$. In that case the boundary conditions becomes:

$$x_0 = \underline{x}_0 \qquad\qquad \tilde{x}_N = \underline{\tilde{x}}_N \quad \tilde{\lambda}_N = \nu^T \quad \bar{\lambda}_N = \frac{\partial}{\partial \bar{x}}\phi_N$$

3.3 Linear terminal constraints

In the previous section we handled the problem with fixed end point state. We will now focus on the problem when only a part of the terminal state is fixed. This has, though, as a special case the simple situation treated in the previous section.

Consider the system ($i = 0, \ \dots \ , N-1$)

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{3.17}$$

the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{3.18}$$

and the linear terminal constraints

$$Cx_N = \underline{r}_N \tag{3.19}$$

where C and $\underline{r}_N$ (and $\underline{x}_0$) are given. The problem consist in bringing the system (3.3) from its initial state $\underline{x}_0$ to a terminal situation in which $Cx_N = \underline{r}_N$ such that the performance index, (3.4) is minimized.

The problem is specified by the functions f, L and ϕ, the length of the horizon N, by the initial state $\underline{x}_0$, the $p \times n$ matrix C and $\underline{r}_N$. Let us apply the usual notation and associate a Lagrange multiplier λ_{i+1} to the equality constraints $x_{i+1} = f_i(x_i, u_i)$. To the terminal constraints we associate, ν which is a vector containing p (scalar) Lagrange multipliers.

Theorem 5: Consider the dynamic optimization problem of bringing the system (3.17) from the initial state to a terminal state such that the end point constraints in (3.19) is met and the performance index (3.18) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $i = 0, \ \dots \ , N-1$):

$$xn = f_i(x_i, u_i) \quad \text{State equation} \tag{3.20}$$

$$\lambda_i^T = \frac{\partial}{\partial x_i} H_i \quad \text{Costate equation} \tag{3.21}$$

$$0^T = \frac{\partial}{\partial u} H_i \quad \text{Stationarity condition} \tag{3.22}$$

The boundary conditions are the initial state and

$$x_0 = \underline{x}_0 \qquad Cx_N = \underline{r}_N \qquad \lambda_N^T = \nu^T C + \frac{\partial}{\partial x_N}\phi \tag{3.23}$$

□

Proof: Again, we start forming the Lagrange function:

$$J_L = \phi(x_N) + \sum_{i=0}^{N-1} \Big[L_i(x_i, u_i) + \lambda_{i+1}^T \big(f_i(x_i, u_i) - x_{i+1} \big) \Big] + \lambda_0^T (\underline{x}_0 - x_0) + \nu^T (Cx_N - \underline{r}_N)$$

As in connection to free dynamic optimization stationarity w.r.t.. λ_{i+1} gives (for $i = 0, \ \dots \ N-1$) the state equations (3.20). In the same way stationarity w.r.t. ν gives

$$Cx_N = \underline{r}_N$$

Stationarity w.r.t. x_i gives (for $i = 1, \ \dots \ N-1$)

$$0 = \frac{\partial}{\partial x} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial x} f_i(x_i, u_i) - \lambda_i^T$$

or the costate equations (3.21), whereas for $i = N$ we have

$$\lambda_N^T = \nu^T C + \frac{\partial}{\partial x_N}\phi$$

Stationarity w.r.t. u_i gives the stationarity condition (for $i = 0, \ \dots \ N-1$):

$$0 = \frac{\partial}{\partial u} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial u} f_i(x_i, u_i)$$

□

Example: 3.3.1 *(**Orbit injection problem** from (Bryson 1999)).*

A body is initial' at rest in the origin. A constant specific thrust force, a, is applied to the body in a direction that makes an angle θ with the x-axis (see Figure 3.2). The task is to find a sequence of directions such that the body in a finite number, N, of intervals

1 is injected into orbit i.e. reach a specific height H

2 has zero vertical speed (y-direction)

3 has maximum horizontal speed (x-direction)

This is also denoted as a Discrete Thrust Direction Programming (DTDP) problem.

Let u and v be the velocity in the x and y direction, respectively. The equation of motion (EOM) is (apply Newton 2 law):

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = a \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix} \qquad \frac{d}{dt}y = v \qquad \begin{bmatrix} u \\ v \\ y \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.24}$$

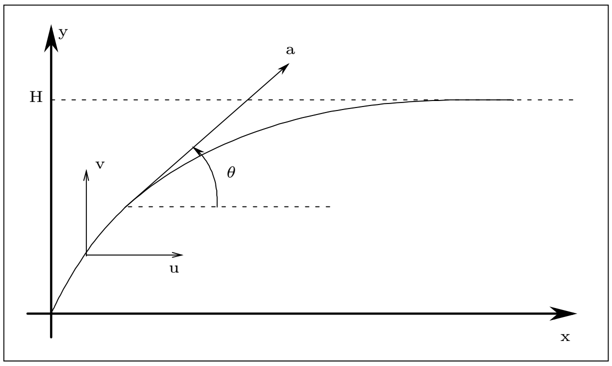

Figure 3.2. Nomenclature for Thrust Direction Programming

If we have a constant angle in the intervals (with length h) then the discrete time state equation is

$$\begin{bmatrix} u \\ v \\ y \end{bmatrix}_{i+1} = \begin{bmatrix} u_i + ah\ cos(\theta_i) \\ v_i + ah\ sin(\theta_i) \\ y_i + v_i h + \frac{1}{2}ah^2\ sin(\theta_i) \end{bmatrix} \qquad \begin{bmatrix} u \\ v \\ y \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.25}$$

The performance index we are going to maximize is

$$J = u_N \tag{3.26}$$

and the end point constraints can be written as

$$v_N = 0 \qquad y_N = H \qquad \text{or as} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ y \end{bmatrix}_N = \begin{bmatrix} 0 \\ H \end{bmatrix} \tag{3.27}$$

In terms of our standard notation we have

$$\phi = u_N = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ y \end{bmatrix}_N \qquad L = 0 \qquad C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad r = \begin{bmatrix} 0 \\ H \end{bmatrix}$$

We assign one (scalar) Lagrange multiplier (or costate) to each of the dynamic elements of the dynamic function

$$\lambda_i = \begin{bmatrix} \lambda^u & \lambda^v & \lambda^y \end{bmatrix}_i^T$$

and the Hamiltonian function becomes

$$H_i = \lambda_{i+1}^u \big[\ u_i + ah\ cos(\theta_i)\ \big] + \lambda_{i+1}^v \big[\ v_i + ah\ sin(\theta_i)\ \big] + \lambda_{i+1}^y \big[\ y_i + v_i h + \frac{1}{2}ah^2 sin(\theta_i)\ \big] \tag{3.28}$$

From this we find the Euler-Lagrange equations

$$\begin{bmatrix} \lambda^u & \lambda^v & \lambda^y \end{bmatrix}_i = \begin{bmatrix} \lambda_{i+1}^u & \lambda_{i+1}^v + \lambda_{i+1}^y h & \lambda_{i+1}^y \end{bmatrix} \tag{3.29}$$

which clearly indicates that λ_i^u and λ_i^y are constant in time and that λ_i^v is decreasing linearly with time (and with rate equal λ^y h). If we for each of the end point constraints in (3.27) assign a (scalar) Lagrange multiplier, ν_v and ν_y, we can write the boundary conditions in (3.23) as

$$\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u \\ v \\ y \end{bmatrix}_N = \begin{bmatrix} 0 \\ H \end{bmatrix} \qquad \begin{bmatrix} \lambda^u \\ \lambda^v \\ \lambda^y \end{bmatrix}_N = \begin{bmatrix} \nu_v & \nu_y \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

or as

$$v_N = 0 \qquad y_N = H \tag{3.30}$$

and

$$\lambda_N^u = 1 \qquad \lambda_N^v = \nu_v \qquad \lambda_N^y = \nu_y \tag{3.31}$$

If we combines (3.31) and (3.29) we find

$$\lambda_i^u = 1 \qquad \lambda_i^v = \nu_v + \nu_y h(N-i) \qquad \lambda_i^y = \nu_y \tag{3.32}$$

From the stationarity condition we find (from the Hamiltonian function in (3.28))

$$0 = -\lambda_{i+1}^u ah\ sin(\theta_i) + \lambda_{i+1}^v ah\ cos(\theta_i) + \lambda_{i+1}^y \frac{1}{2} ah\ cos(\theta_i)$$

or

$$tan(\theta_i) = \frac{\lambda^v_{i+1} + \frac{1}{2}\lambda^y_{i+1} h}{\lambda^u_{i+1}}$$

or with the costate inserted

$$\boxed{tan(\theta_i) = \nu_v + \nu_y h(N + \frac{1}{2} - i) \qquad (3.33)}$$

The two constant, ν_v and ν_y must be determined to satisfy $y_N = H$ and $v_N = 0$. This can be done by establish the mapping from the two constants to y_N and v_N and solve (numerically or analytically) for ν_v and ν_y.

In the following we measure time in units of $T = Nh$, velocities such as u and v in units of aT^2, then we can put $a = 1$ and $h = 1/N$ in the equations above.

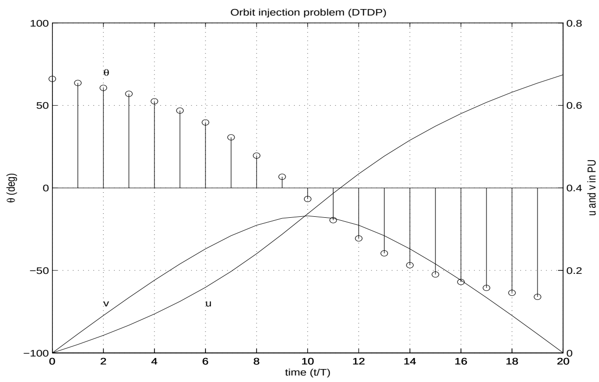

Figure 3.3. DTDP for max u_N with $H = 0.2$. Thrust direction angle, vertical and horizontal velocity.

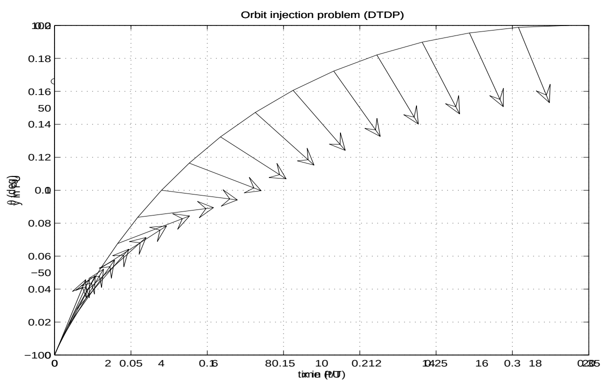

Figure 3.4. DTDP for max u_N with $H = 0.2$. Position and thrust direction angle.

□

3.4 General terminal equality constraints

Let us now solve the more general problem in which the end point constraints is given in terms of a nonlinear function ψ, i.e.

$$\psi(x_N) = 0 \qquad (3.34)$$

This has, as a special case, the previously treated situations.

Consider the discrete time system ($i = 0, \ ... \ , N - 1$)

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \qquad (3.35)$$

the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{3.36}$$

and the terminal constraints (3.34). The initial state, $\underline{x}_0$, is given (known). The problem consist in bringing the system (3.35) i from its initial state $\underline{x}_0$ to a terminal situation in which $\psi(x_N) = 0$ such that the performance index, (3.36) is minimized.

The problem is specified by the functions f, L, ϕ and ψ, the length of the horizon N and by the initial state $\underline{x}_0$. Let us apply the usual notation and associate a Lagrange multiplier λ_{i+1} to each of the equality constraints $x_{i+1} = f_i(x_i, u_i)$. To the terminal constraints we associate, ν which is a vector containing p (scalar) Lagrange multipliers.

Theorem 6: Consider the dynamic optimization problem of bringing the system (3.35) from the initial state such that the performance index (3.36) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $i = 0, \ \dots, N-1$):

$$\begin{aligned} x_{i+1} &= f_i(x_i, u_i) && \text{State equation} && (3.37) \\ \lambda_i^T &= \frac{\partial}{\partial x_i} H_i && \text{Costate equation} && (3.38) \\ 0^T &= \frac{\partial}{\partial u} H_i && \text{Stationarity condition} && (3.39) \end{aligned}$$

The boundary conditions are:

$$x_0 = \underline{x}_0 \qquad \psi(x_N) = 0 \qquad \lambda_N^T = \nu^T \frac{\partial}{\partial x}\psi + \frac{\partial}{\partial x_N}\phi$$

□

Proof: As usual, we start forming the Lagrange function:

$$J_L = \phi(x_N) + \sum_{i=0}^{N-1} \Big[L_i(x_i, u_i) + \lambda_{i+1}^T \big(f_i(x_i, u_i) - x_{i+1}\big) \Big] + \lambda_0^T(\underline{x}_0 - x_0) + \nu^T(\psi(x_N))$$

As in connection to free dynamic optimization stationarity w.r.t.. λ_{i+1} gives (for $i = 0, \ \dots N-1$) the state equations (3.37). In the same way stationarity w.r.t. ν gives

$$\psi(x_N) = 0$$

Stationarity w.r.t. x_i gives (for $i = 1, \ \dots N-1$)

$$0 = \frac{\partial}{\partial x} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial x} f_i(x_i, u_i) - \lambda_i^T$$

or the costate equations (3.38), whereas for $i = N$ we have

$$\lambda_N^T = \nu^T \frac{\partial}{\partial x}\psi + \frac{\partial}{\partial x_N}\phi$$

Stationarity w.r.t. u_i gives the stationarity condition (for $i = 0, \ \dots N-1$):

$$0 = \frac{\partial}{\partial u} L_i(x_i, u_i) + \lambda_{i+1}^T \frac{\partial}{\partial u} f_i(x_i, u_i)$$

□

3.5 Continuous dynamic optimization with end point constraints.

In this section we consider the continuous case in which $t \in [0;\ T] \in \mathbb{R}$. The problem is to find the input function u_t to the system

$$\dot{x} = f_t(x_t, u_t) \qquad\qquad x_0 = \underline{x}_0 \tag{3.40}$$

such that the cost function

$$J = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt \tag{3.41}$$

is minimized and the end point constraints in

$$\psi_T(x_T) = 0 \tag{3.42}$$

are met. Here the initial state $\underline{x}_0$ and final time T are given (fixed). The problem is specified by the dynamic function, f_t, the scalar value functions ϕ and L, the end point constraints through the function ψ and the constants T and $\underline{x}_0$.

As in section 2.3 we can for the sake of convenience introduce the scalar Hamiltonian function as:

$$H_t(x_t, u_t, \lambda_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) \tag{3.43}$$

As in the previous section on discrete time problems we, in addition to the costate (the dynamics is an equality constraints), introduce a Lagrange multiplier, ν associated with the end point constraints.

Theorem 7: Consider the dynamic optimization problem in continuous time of bringing the system (3.40) from the initial state and a terminal state satisfying (3.42) such that the performance index (3.41) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $t \in [0,\ T]$):

$$\dot{x}_t = f_t(x_t, u_t) \qquad \text{State equation} \tag{3.44}$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \text{Costate equation} \tag{3.45}$$

$$0^T = \frac{\partial}{\partial u_t} H_t \qquad \text{stationarity condition} \tag{3.46}$$

and the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \psi_T(x_T) = 0 \qquad \lambda_T = \nu^T \frac{\partial}{\partial x}\psi_T + \frac{\partial}{\partial x}\phi_T(x_T) \tag{3.47}$$

which is a split boundary condition. □

Proof: As in section 2.3 we first construct the Lagrange function:

$$J_L = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt + \int_0^T \lambda_t^T \left[f_t(x_t, u_t) - \dot{x}_t\right] dt + \nu^T \psi_T(x_T)$$

Then we introduce integration by part

$$\int_0^T \lambda_t^T \dot{x}_t dt + \int_0^T \dot{\lambda}_t^T x_t = \lambda_T^T x_T - \lambda_0^T x_0$$

in the Lagrange function which results in:

$$J_L = \phi_T(x_T) + \lambda_0^T x_0 - \lambda_T^T x_T + \nu^T \psi_T(x_T) + \int_0^T \Big(L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) + \dot{\lambda}_t^T x_t\Big)\, dt$$

Using Leibniz rule (Lemma 2) the variation in J_L w.r.t. x, λ and u is:

$$\begin{aligned} dJ_L &= \left(\frac{\partial}{\partial x_T}\phi_T + \nu^T \frac{\partial}{\partial x}\psi_T - \lambda_T^T\right) dx_T + \int_0^T \left(\frac{\partial}{\partial x}L + \lambda^T \frac{\partial}{\partial x} f + \dot{\lambda}^T\right) \delta x\ dt \\ &\quad + \int_0^T \left(f_t(x_t, u_t) - \dot{x}_t\right) \delta\lambda\ dt + \int_0^T \left(\frac{\partial}{\partial u}L + \lambda^T \frac{\partial}{\partial u} f\right) \delta u\ dt \end{aligned}$$

According to optimization with equality constraints the necessary condition is obtained as a stationary point to the Lagrange function. Setting to zero all the coefficients of the independent increments yields necessary condition as given in Theorem 7. □

We can express the necessary conditions as

$$\dot{x}^T = \frac{\partial}{\partial \lambda} H \qquad -\dot{\lambda}^T = \frac{\partial}{\partial x} H \qquad 0^T = \frac{\partial}{\partial u} H \tag{3.48}$$

with the (split) boundary conditions

$$x_0 = \underline{x}_0 \qquad \psi_T(x_T) = 0 \qquad \lambda_T^T = \nu^T \frac{\partial}{\partial x}\psi_T + \frac{\partial}{\partial x}\phi_T$$

The only difference between this formulation and the one given in Theorem 7 is the alternative formulation of the state equation.

If we have *simple end point constraints* where the problem is to bring the system from the initial state $\underline{x}_0$ to the final state $\underline{x}_T$ in a fixed period of time along a trajectory such that the performance index, (3.41), is minimized. In that case

$$\psi_T(x_T) = x_T - \underline{x}_T = 0$$

and the boundary condition in (3.47) becomes:

$$x_0 = \underline{x}_0 \qquad x_T = \underline{x}_T \qquad \lambda_T = \nu^T \qquad \left[+\frac{\partial}{\partial x}\phi_T(x_T)\right] \tag{3.49}$$

If we have *simple partial end point constraints* the situation is quite similar to the previous one. Assume some of the terminal state variable, $\tilde{x}_T$, is constrained i a simple way and the rest of the variable, $\bar{x}_T$, is not constrained, i.e.

$$x_T = \begin{bmatrix} \tilde{x}_T \\ \bar{x}_T \end{bmatrix} \qquad \tilde{x}_T = \underline{\tilde{x}}_T \tag{3.50}$$

The rest of the state variabel, $\bar{x}_T$, might influence the terminal contribution, $\phi_T(x_T)$. Assume for simplicity that $\tilde{x}_T$ do not influence on ϕ_T, then $\phi_T(x_T) = \phi_T(\bar{x}_T)$. In that case the boundary conditions becomes:

$$x_0 = \underline{x}_0 \qquad \tilde{x}_T = \underline{\tilde{x}}_T \qquad \tilde{\lambda}_T = \nu^T \qquad \bar{\lambda}_T = \frac{\partial}{\partial \bar{x}}\phi_T$$

In the more complicated situation where there is *linear end point constraints* of the type

$$Cx_T = \underline{r}$$

Here the known quantities is C, which is a $p \times n$ matrix and $r \in \mathbb{R}^p$. The system is brought from the initial state $\underline{x}_0$ to the final state x_T such that $Cx_T = \underline{r}$, in a fixed period of time along a trajectory such that the performance index, (3.41), is minimized. The boundary condition in (3.47) becomes here:

$$x_0 = \underline{x}_0 \qquad Cx_T = \underline{r} \qquad \lambda_T = \nu^T C + \frac{\partial}{\partial x}\phi_T(x_T) \tag{3.51}$$

Example: 3.5.1 (Motion control) *Let us consider the continuous time version of example 3.1.1. (Eventually see also the unconstrained continuous version in Example 2.3.1). The problem is to bring the system*

$$\dot{x} = u_t \qquad x_0 = \underline{x}_0$$

in final (known) time T from the initial position, $\underline{x}_0$, to the final position, $\underline{x}_t$, such that the performance index

$$J = \frac{1}{2} p x_T^2 + \int_0^T \frac{1}{2} u^2 dt$$

is minimized. The terminal term, $\frac{1}{2}px_T^2$, could have been omitted since only give a constant contribution to the performance index. It has been included here in order to make the comparison with Example 2.3.1 more obvious.

The Hamiltonian function is (also) in this case

$$H = \frac{1}{2}u^2 + \lambda u$$

and the Euler-Lagrange equations are simply

$$\begin{aligned} \dot{x} &= u_t \\ -\dot{\lambda} &= 0 \\ 0 &= u + \lambda \end{aligned}$$

with the boundary conditions:

$$x_0 = \underline{x}_0 \qquad x_T = \underline{x}_T \qquad \lambda_T = \nu + p x_T$$

As in Example 2.3.1 these equations are easily solved and it is also the costate equation here gives the key to the solution. Firstly, we notice that the costate is constant. Let us denote this constant as c.

$$\lambda = c$$

From the stationarity condition we find the control signal (expressed as function of the terminal state x_T) is given as

$$u = -c$$

If this strategy is introduced in the state equation we have

$$x_t = \underline{x}_0 - ct$$

and

$$\underline{x}_T = \underline{x}_0 - cT \qquad or \qquad c = \frac{\underline{x}_0 - \underline{x}_T}{T}$$

Finally, we have

$$x_t = \underline{x}_0 + \frac{\underline{x}_T - \underline{x}_0}{T} t \qquad u_t = \frac{\underline{x}_T - \underline{x}_0}{T} \qquad \lambda = \frac{\underline{x}_0 - \underline{x}_T}{T}$$

It is also quite simple to see, that the Hamiltonian function is constant and equal

$$H = -\frac{1}{2}\left[\frac{\underline{x}_T - \underline{x}_0}{T}\right]^2$$

□

Example: 3.5.2 *(**Orbit injection** from (Bryson 1999)). Let us return to the continuous time version of the orbit injection problem (see. Example 3.3.1.) The problem is to find the input function, θ_t, such that the terminal horizontal velocity, u_T, is maximized subject to the dynamics*

$$\frac{d}{dt}\begin{bmatrix} u_t \\ v_t \\ y_t \end{bmatrix} = \begin{bmatrix} a\ cos(\theta_t) \\ a\ sin(\theta_t) \\ v_t \end{bmatrix} \qquad \begin{bmatrix} u_0 \\ v_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.52}$$

and the terminal constraints

$$v_T = 0 \qquad y_T = H$$

With our standard notation (in relation to Theorem 7) we have

$$J = \phi_T(x_T) = u_T \qquad L = 0 \qquad C = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \underline{r} = \begin{bmatrix} 0 \\ H \end{bmatrix}$$

and the Hamilton functions is

$$H_t = \lambda_t^u a\ cos(\theta_t) + \lambda_t^v a\ sin(\theta_t) + \lambda_t^y v_t$$

The Euler-Lagrange equations consists of the state equation, (3.52), the costate equation

$$-\frac{d}{dt}\begin{bmatrix} \lambda_t^u & \lambda_t^v & \lambda_t^y \end{bmatrix} = \begin{bmatrix} 0 & \lambda_t^y & 0 \end{bmatrix} \tag{3.53}$$

and the stationarity condition

$$0 = -\lambda^u a \; sin(\theta_t) + \lambda^v a \; cos(\theta_t)$$

or

$$tan(\theta_t) = \frac{\lambda_t^v}{\lambda_t^u} \tag{3.54}$$

The costate equations clearly shown that the costate λ_t^u *and* λ_t^y *are constant and that* λ_t^v *has a linear evolution with* λ^y *as slope. To each of the two terminal constraints*

$$\psi = \begin{bmatrix} v_T \\ y_T - H \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} u_T \\ v_T \\ y_T \end{bmatrix} - \begin{bmatrix} 0 \\ H \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

we associate two (scalar) Lagrange multipliers, ν_v *and* ν_y*, and the boundary condition in (3.47) gives*

$$\begin{bmatrix} \lambda_T^u & \lambda_T^v & \lambda_T^y \end{bmatrix} = \begin{bmatrix} \nu_v & \nu_y \end{bmatrix} \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

or

$$\lambda_T^u = 1 \qquad \lambda_T^v = \nu_v \qquad \lambda_T^y = \nu_y$$

If this is combined with the costate equations we have

$$\lambda_t^u = 1 \qquad \lambda_t^v = \nu_v + \nu_y(T-t) \qquad \lambda_t^y = \nu_y$$

and the stationarity condition gives the optimal decision function

$$\boxed{tan(\theta_t) = \nu_v + \nu_y(T-t)} \tag{3.55}$$

The two constants, ν_u *and* ν_y *has to be determined such that the end point constraints are met. This can be achieved by establish the mapping from the two constant and the state trajectories and the end points. This can be done by integrating the state equations either by means of analytical or numerical methods.*

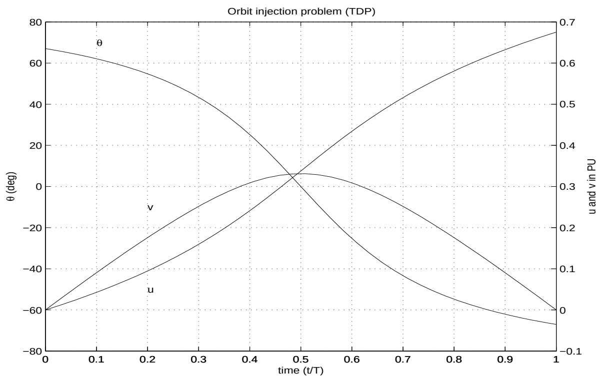

Figure 3.5. TDP for max u_T with $H = 0.2$. Thrust direction angle, vertical and horizontal velocity.

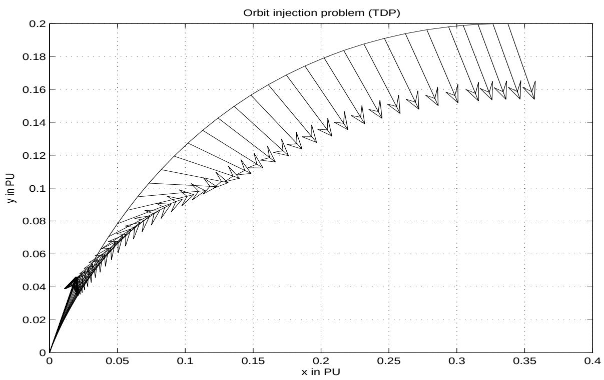

Figure 3.6. TDP for max u_T with $H = 0.2$. Position and thrust direction angle.

□

Chapter 4

The maximum principle

In this chapter we will be dealing with problems where the control actions or the decisions are constrained. One example of constrained control actions is the *Box model* where the control actions are continuous, but limited to certain region

$$|u_i| \leq \underline{u}$$

In the vector case the inequality apply elementwise. Another type of constrained control is where the possible action are finite and discrete e.g. of the type

$$u_i \in \{-1,\ 0,\ 1\}$$

In general we will write

$$u_i \in \mathcal{U}_i$$

where $\mathcal{U}_i$ is feasible set (i.e. the set of allowed decisions).

The necessary conditions are denoted as the maximum principle or Pontryagins maximum principle. In some part of the literature one can only find the name of Pontryagin in connection to the continuous time problem. In other part of the literature the principle is also denoted as the minimum principle if it is a minimization problem. Here we will use the name Pontryagins maximum principle also when we are minimizing.

4.1 Pontryagins maximum principle (D)

Consider the discrete time system ($i = 0,\ \dots\ , N-1$)

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{4.1}$$

and the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{4.2}$$

where the control actions are constrained, i.e.

$$u_i \in \mathcal{U}_i \tag{4.3}$$

The task is to take the system, i.e. to find the sequence of feasible (i.e. satisfying (4.3)) decisions or control actions, u_i $i = 0,\ 1,\ \dots\ N-1$, that takes the system in (4.1) from its initial state $\underline{x}_0$ along a trajectory such that the performance index (4.2) is minimized.

Notice, as in the previous sections we can introduce the Hamiltonian function

$$H_i(x_i, u_i, \lambda_{i+1}) = L_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i)$$

and obtain a much more compact form for necessary conditions, which is stated in the theorem below.

Theorem 8: Consider the dynamic optimization problem of bringing the system (4.1) from the initial state such that the performance index (4.2) is minimized. The necessary condition is given by the following equations (for $i = 0, \ \dots \ , N-1$):

$$x_{i+1} = f_i(x_i, u_i) \qquad \text{State equation} \qquad (4.4)$$

$$\lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad \text{Costate equation} \qquad (4.5)$$

$$u_i = arg \min_{u_i \in \mathcal{U}_i} [H_i] \qquad \text{Optimality condition} \qquad (4.6)$$

The boundary conditions are:

$$x_0 = \underline{x}_0 \qquad \lambda_N^T = \frac{\partial}{\partial x_N} \phi$$

□

Proof: Omitted here. It can be proved by means of dynamic programming which will be treated later (Chapter 6) in these notes. □

If the problem is a maximization problem then then the optimality condition in (4.6) is a maximization rather than a minimization.

Note, if we have end point constraints such as

$$\psi_N(x_N) = 0 \qquad \psi : \mathbb{R}^n \rightarrow \mathbb{R}^p$$

we can introduce a Lagrange multiplier, $\nu \in \mathbb{R}^p$ related to each of the $p \leq n$ end point constraints and the boundary condition are changed into

$$x_0 = \underline{x}_0 \quad \psi(x_N) = 0 \qquad \lambda_N^T = \nu^T \frac{\partial}{\partial x_N} \psi_N + \frac{\partial}{\partial x_N} \phi_N$$

Example: 4.1.1 Investment planning. *Consider the problem from Example 3.1.2, page 26 where we are planning to invest some money during a period of time with N intervals in order to save a specific amount of money $\underline{x}_N = 10000\$$. If the the bank pays interest with rate α in one interval, the account balance will evolve according to*

$$x_{i+1} = (1+\alpha)x_i + u_i \qquad x_0 = 0 \tag{4.7}$$

Here u_i is the deposit per period. As is Example 3.1.2 we will be looking for a minimum effort plan. This could be achieved if the deposits are such that the performance index:

$$J = \sum_{i=0}^{N-1} \frac{1}{2} u_i^2 \tag{4.8}$$

is minimized. In this Example the deposit is however limited to 600 \$.

The Hamiltonian function is

$$H_i = \frac{1}{2} u_i^2 + \lambda_{i+1} \left[(1+\alpha)x_i + u_i \right]$$

and the necessary conditions are:

$$x_{i+1} = (1+\alpha)x_i + u_i \tag{4.9}$$
$$\lambda_i = (1+\alpha)\lambda_{i+1} \tag{4.10}$$
$$u_i = arg \min_{u_i \in \mathcal{U}_i} \left(\frac{1}{2}u_i^2 + \lambda_{i+1}\left[(1+\alpha)x_i + u_i\right] \right) \tag{4.11}$$

As in Example 3.1.2 we can introduce the constants $a = 1+\alpha$ *and* $q = \frac{1}{a}$ *and solve the Costate equation*

$$\lambda_i = c\; q^i$$

where c *is an unknown constant. The optimal deposit is according to (4.11) given by*

$$u_i = min(\underline{u}, -c\; q^{i+1})$$

which inserted in the state equation enable us to find (iterate) the state trajectory for a given value of c*. The correct value of* c *give*

$$x_N = \underline{x}_N = 10000\$ \tag{4.12}$$

The plots in Figure 4.1 has been produced by means of a shooting method where c *has been determined to satisfy the*

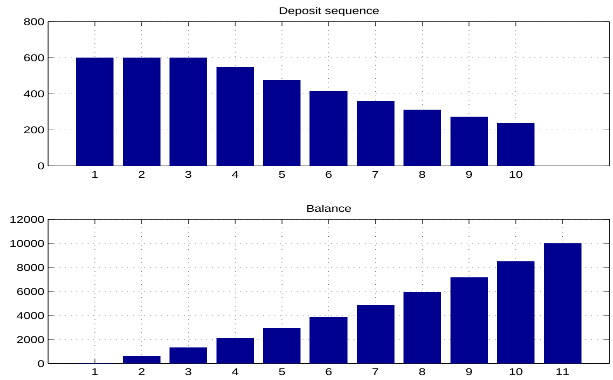

Figure 4.1. Investment planning. Upper panel show the annual deposit and the lower panel shows the account balance.

□

Example: 4.1.2 *(***Orbit injection problem** *from (Bryson 1999)).*

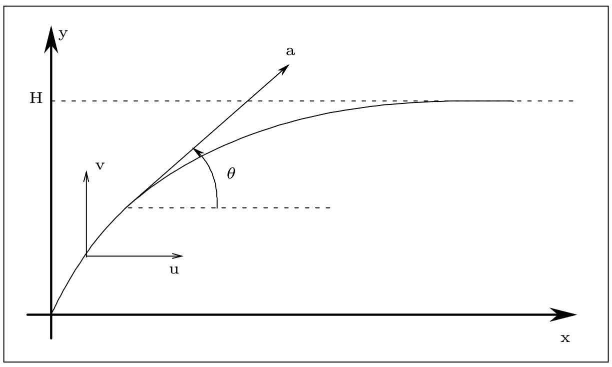

Figure 4.2. Nomenclature for Thrust Direction Programming

Let us return the Orbit injection problem (or Thrust Direction Programming) from Example 3.3.1 on page 30 where a body is accelerated and put in orbit, which in this setup means reach a specific height H*. The problem is to find a* *sequence of thrusts directions such that the end (i.e. for $i = N$) horizontal velocity is maximized while the vertical velocity is zero.*

The specific thrust has a (time varying) horizontal component a_x and a (time varying) vertical component a_y, but has a constant size a. This problem was in Example 3.3.1 solved by introducing the angle θ between the thrust force and the x-axis such that

$$\begin{bmatrix} a^x \\ a^y \end{bmatrix} = a \begin{bmatrix} cos(\theta) \\ sin(\theta) \end{bmatrix}$$

This ensure that the size of the specific trust force is constant and equal a. In this example we will follow another approach and use both a^x and a^y as decision variable. They are constrained through

$$(a^x)^2 + (a^y)^2 = a^2 \tag{4.13}$$

Let (again) u and v be the velocity in the x and y direction, respectively. The equation of motion (EOM) is (apply Newton 2 law):

$$\frac{d}{dt}\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} a^x \\ a^y \end{bmatrix} \qquad \frac{d}{dt}y = v \qquad \begin{bmatrix} u \\ v \\ y \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.14}$$

We have for sake of simplicity omitted the x-coordinate. If the specific thrust is kept constant in intervals (with length h) then the discrete time state equation is

$$\begin{bmatrix} u \\ v \\ y \end{bmatrix}_{i+1} = \begin{bmatrix} u_i + a_i^x h \\ v_i + a_i^y h \\ y_i + v_i h + \frac{1}{2}a_i^y h^2 \end{bmatrix} \qquad \begin{bmatrix} u \\ v \\ y \end{bmatrix}_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.15}$$

where the decision variable or control actions are constrained through (4.13). The performance index we are going to maximize is

$$J = u_N \tag{4.16}$$

and the end point constraints can be written as

$$v_N = 0 \qquad y_N = H \qquad \textit{or as} \qquad \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} u \\ v \\ y \end{bmatrix}_N = \begin{bmatrix} 0 \\ H \end{bmatrix} \tag{4.17}$$

If we (as in Example 3.3.1 assign one (scalar) Lagrange multiplier (or costate) to each of the dynamic elements of the dynamic function

$$\lambda_i = \begin{bmatrix} \lambda^u & \lambda^v & \lambda^y \end{bmatrix}_i^T$$

the Hamiltonian function becomes

$$H_i = \lambda_{i+1}^u(u_i + a_i^x h) + \lambda_{i+1}^v(v_i + a_i^y h) + \lambda_{i+1}^y(y_i + v_i h + \frac{1}{2}a_i^y h^2) \tag{4.18}$$

For the costate we have the same situation as in Example 3.3.1 and

$$\begin{bmatrix} \lambda^u, & \lambda^v, & \lambda^y \end{bmatrix}_i = \begin{bmatrix} \lambda_{i+1}^u, & \lambda_{i+1}^v + \lambda_{i+1}^y h, & \lambda_{i+1}^y \end{bmatrix} \tag{4.19}$$

with the end point constraints

$$v_N = 0 \qquad y_N = H$$

and

$$\lambda_N^u = 1 \qquad \lambda_N^v = \nu_v \qquad \lambda_N^y = \nu_y$$

where ν_v and ν_y are Lagrange multipliers related to the end point constraints. If we combines the costate equation and the end point conditions we find

$$\lambda_i^u = 1 \qquad \lambda_i^v = \nu_v + \nu_y h(N-i) \qquad \lambda_i^y = \nu_y \tag{4.20}$$

Now consider the maximization of H_i in (4.18) with respect to a_i^x and a_i^y subject to (4.13). The decision variable form a vector which maximize the Hamiltonian function if it is parallel to the vector

$$\begin{bmatrix} \lambda_{i+1}^u h \\ \lambda_{i+1}^v h + \frac{1}{2}\lambda_{i+1}^y h^2 \end{bmatrix}$$

Since the length of the decision vector is constrained by (4.13) the optimal vector is:

$$\begin{bmatrix} a_i^x \\ a_i^y \end{bmatrix} = \begin{bmatrix} \lambda_{i+1}^u h \\ \lambda_{i+1}^v h + \frac{1}{2}\lambda_{i+1}^y h^2 \end{bmatrix} \frac{a}{\sqrt{(\lambda_{i+1}^u h)^2 + (\lambda_{i+1}^v h + \frac{1}{2}\lambda_{i+1}^y h^2)^2}} \tag{4.21}$$

If the two constants ν_v and ν_y are known, then the input sequence given by (4.21) (and (4.20)) can be used in conjunction with the state equation, (4.15) and the state trajectories can be determined. The two unknown constant can then be found by means of numerical search such that the end point constraints in (4.17) is met. The results are depicted in Figure 4.3 in per unit (PU) as in Example 3.3.1. In Figure 4.3 the accelerations in the x- and y-direction is plotted versus time as a stem plot. The velocities, u_i and v_i, are also plotted and have the same evolution as in 3.3.1.

□

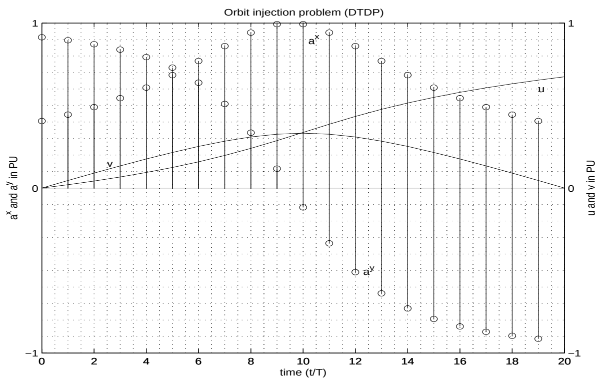

Figure 4.3. The Optimal orbit injection for $H = 0.2$ (in PU). Specific thrust force a^x and a^y and vertical and horizontal velocity.

4.2 Pontryagins maximum principle (C)

Let us now focus on the continuous version of the problem in which $t \in \mathbb{R}$. The problem is to find a feasible input function

$$u_t \in \mathcal{U}_t \tag{4.22}$$

to the system

$$\dot{x} = f_t(x_t, u_t) \qquad\qquad x_0 = \underline{x}_0 \tag{4.23}$$

such that the cost function

$$J = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt \tag{4.24}$$

is minimized. Here the initial state $\underline{x}_0$ and final time T are given (fixed). The problem is specified by the dynamic function, f_t, the scalar value functions ϕ_T and L_t and the constants T and $\underline{x}_0$.

As in section 2.3 we can for the sake of convenience introduce the scalar Hamiltonian function as:

$$H_t(x_t, u_t, \lambda_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) \tag{4.25}$$

Theorem 9: Consider the dynamic optimization problem in continuous time of bringing the system (4.23) from the initial state such that the performance index (4.24) is minimized. The necessary condition is given by the following equations (for $t \in [0,\ T]$):

$$\dot{x}_t = f_t(x_t, u_t) \qquad \text{State equation} \tag{4.26}$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \text{Costate equation} \tag{4.27}$$

$$u_t = arg \min_{u_t \in \mathcal{U}_t} [H_t] \qquad \text{Optimality condition} \tag{4.28}$$

and the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_T = \frac{\partial}{\partial x}\phi_T(x_T) \tag{4.29}$$

which is a split boundary condition. □

Proof: Omitted □

If the problem is a maximization problem, then the minimization in (4.28) is changed into a maximization.

If we have end point constraints, such as

$$\psi_T(x_T) = 0$$

the boundary conditions are changed into:

$$x_0 = \underline{x}_0 \qquad \psi_T(x_T) = 0 \qquad \lambda_T^T = \nu^T \frac{\partial}{\partial x}\psi_T + \frac{\partial}{\partial x}\phi_T$$

Example: 4.2.1 *(**Orbit injection** from (Bryson 1999)). Let us return to the continuous time version of the orbit injection problem (see. Example 3.5.2, page 35). In that example the constraint on the size of the specific thrust was solved by introducing the angle between the thrust force and the x-axis. Here we will solve the problem using Pontryagins maximum principle. The problem is here to find the input function, i.e. the horizontal (a^x) and vertical (a^y) component of the specific thrust force, satisfying*

$$(a_t^x)^2 + (a_t^y)^2 = a^2 \tag{4.30}$$

such that the terminal horizontal velocity, u_T, is maximized subject to the dynamics

$$\frac{d}{dt}\begin{bmatrix} u_t \\ v_t \\ y \end{bmatrix} = \begin{bmatrix} a_t^x \\ a_t^y \\ v_t \end{bmatrix} \qquad \begin{bmatrix} u_0 \\ v_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.31}$$

and the terminal constraints

$$v_T = 0 \qquad y_T = H \tag{4.32}$$

With our standard notation (in relation to Theorem 9 and (3.51)) we have

$$J = \phi_T(x_T) = u_T \qquad L = 0$$

and the Hamilton functions is

$$H_t = \lambda_t^u a_t^x + \lambda_t^v a_t^y + \lambda_t^y v_t$$

The necessary conditions consist of the state equation, (4.31), the costate equation

$$-\frac{d}{dt}\begin{bmatrix} \lambda_t^u & \lambda_t^v & \lambda_t^y \end{bmatrix} = \begin{bmatrix} 0 & \lambda_t^y & 0 \end{bmatrix} \tag{4.33}$$

and the optimality condition

$$\begin{bmatrix} a_t^x \\ a_t^y \end{bmatrix} = arg\max\left(\lambda_t^u a_t^x + \lambda_t^v a_t^y + \lambda_t^y v_t\right)$$

The maximization in the optimality conditions is with respect to the constraint in (4.30). It is easily seen that the solution to this constrained optimization is given by

$$\boxed{\begin{bmatrix} a_t^x \\ a_t^y \end{bmatrix} = \begin{bmatrix} \lambda_t^u \\ \lambda_t^v \end{bmatrix} \frac{a}{\sqrt{(\lambda_t^u)^2 + (\lambda_t^v)^2}}} \tag{4.34}$$

The costate equations clearly shown that the costate λ_t^u and λ_t^y are constant and that λ_t^v has a linear evolution with λ^y as slope. To each of the two terminal constraints in (4.32) we associate a (scalar) Lagrange multipliers, ν_v and ν_y, and the boundary condition is

$$\lambda_T^u = 1 \qquad \lambda_T^v = \nu_v \qquad \lambda_T^y = \nu_y$$

If this is combined with the costate equations we have

$$\lambda_t^u = 1 \qquad \lambda_t^v = \nu_v + \nu_y(T - t) \qquad \lambda_t^y = \nu_y$$

The two constants, ν_u and ν_y has to be determined such that the end point constraints in (4.32) are met. This can be achieved by establish the mapping from the two constant to the state trajectories and the end point values. This can be done by integrating the state equations either by means of analytical or numerical methods. □

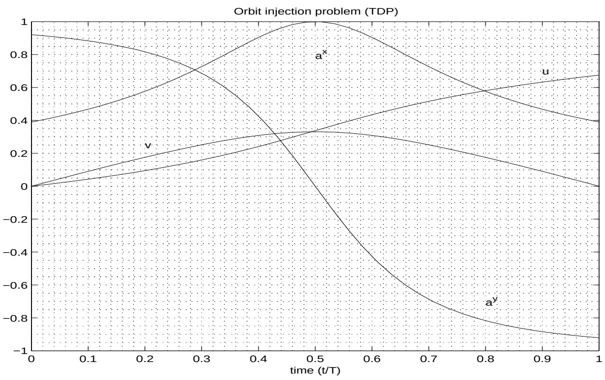

Figure 4.4. TDP for max u_T with $H = 0.2$. Specific thrust force a^x and a^y and vertical and horizontal velocity.

Chapter 5

Time optimal problems

This chapter is devoted to problems in which the length of the period, i.e. T (continuous time) or N (discrete time), is a part of the optimization. Here we will start with the continuous time case.

5.1 Continuous dynamic optimization.

In this section we consider the continuous case in which $t \in [0;\ T] \in \mathbb{R}$. The problem is to find the input function u_t to the system

$$\dot{x} = f_t(x_t, u_t) \qquad\qquad x_0 = \underline{x}_0 \tag{5.1}$$

such that the cost function

$$J = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt \tag{5.2}$$

is minimized. Here the final time T is free and is a part of the optimization and the initial state $\underline{x}_0$ is given (fixed). The problem is specified by the dynamic function, f_t, the scalar value functions ϕ and L and the constant $\underline{x}_0$.

As in section 2.3 we can for the sake of convenience introduce the scalar Hamiltonian function as:

$$H_t(x_t, u_t, \lambda_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) \tag{5.3}$$

Theorem 10: Consider the dynamic optimization problem in continuous time of bringing the system (5.1) from the initial state along a trajectory such that the performance index (5.2) is minimized. The necessary condition is given by the Euler-Lagrange equations (for $t \in [0,\ T]$):

$$\dot{x}_t = f_t(x_t, u_t) \qquad \text{State equation} \tag{5.4}$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \text{Costate equation} \tag{5.5}$$

$$0^T = \frac{\partial}{\partial u_t} H_t \qquad \text{Stationarity condition} \tag{5.6}$$

and the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_T = \frac{\partial}{\partial x} \phi_T(x_T) \tag{5.7}$$

which is a split boundary condition. Due to the free terminal time, T, the solution must satisfy

$$\frac{\partial \phi_T}{\partial T} + H_T = 0 \tag{5.8}$$

which denoted ad the *Transversality condition.* □

Proof: As in section 2.3 we first construct the Lagrange function:

$$J_L = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)dt + \int_0^T \lambda_t^T \left[f_t(x_t, u_t) - \dot{x}_t \right] dt + \nu^T \psi_T(x_T)$$

or in short

$$J_L = \phi_T + \int_0^T \Big(\; H_t - \lambda^T \dot{x} \; \Big) dt$$

where we have introduced the Hamilton function

$$H_t \triangleq H_t(x_t, u_t, \lambda_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t)$$

In the following we going to study the differentials of x_t, i.e. dx_t. The problem is the differentials dx_t and dt are independent. Let us define the variation in x_t, δx_t as the incremental change in x_t when time t is held fixed. Thus, we have the relation

$$dx_t = \delta x_t + \dot{x}_t dt \tag{5.9}$$

Using Leibniz rule (See Lemma 1, 21) we have

$$\begin{aligned} dJ_L = & \frac{\partial \phi_T}{\partial x} dx_T + \frac{\partial \phi_T}{\partial T} dT + \left(H_T - \lambda_T^T \dot{x}_T \right) dT \\ & + \int_0^T \left[\frac{\partial H_t}{\partial x} \delta x - \lambda^T \delta \dot{x} + \frac{\partial H_t}{\partial u} \delta u + \left(\frac{\partial H_t}{\partial \lambda} - \dot{x}^T \right) \delta \lambda \right] \; dt \end{aligned}$$

Then we introduce integration by part

$$\int_0^T \lambda_t^T \delta \dot{x}_t dt + \int_0^T \dot{\lambda}_t^T \delta x_t = \lambda_T^T \delta x_T - \lambda_0^T \delta x_0$$

in the Lagrange function which results in:

$$\begin{aligned} dJ_L = & \frac{\partial \phi_T}{\partial x} dx_T + \frac{\partial \phi_T}{\partial T} dT + \left(H_T - \lambda_T^T \dot{x}_T \right) dT - \lambda_0^T \delta x_0 + \lambda_T^T \delta x_T \\ & + \int_0^T \left[\left(\frac{\partial H_t}{\partial x} + \dot{\lambda}^T \right) \delta x + \frac{\partial H_t}{\partial u} \delta u + \left(\frac{\partial H_t}{\partial \lambda} - \dot{x}^T \right) \delta \lambda \right] \; dt \end{aligned}$$

Notice there are terms depending on dx_T and δx_T. If we apply (5.9) we end up with

$$\begin{aligned} dJ_L = & \left(\frac{\partial \phi_T}{\partial x} - \lambda^T \right) dx_T + \left(\frac{\partial \phi_T}{\partial T} + H_T \right) dT + \\ & + \int_0^T \left[\left(\frac{\partial H_t}{\partial x} + \dot{\lambda}^T \right) \delta x + \frac{\partial H_t}{\partial u} \delta u + \left(\frac{\partial H_t}{\partial \lambda} - \dot{x}^T \right) \delta \lambda \right] \; dt \end{aligned}$$

According to optimization with equality constraints the necessary condition is obtained as a stationary point to the Lagrange function. Setting to zero all the coefficients of the independent increments yields necessary condition as given in Theorem 7. □

Normally, time optimal problems involves some kind of constraints. Firstly, the end point might be constrained in the following manner

$$\psi_T(x_T) = 0 \tag{5.10}$$

as we have seen in Chapter 3. Furthermore the decision might be constrained as well. I Chapter 4 we dealt with problems in which the control action was constrained to

$$u_t \in \mathcal{U}_t \tag{5.11}$$

Theorem 11: Consider the dynamic optimization problem in continuous time of bringing the system (5.1) from the initial state and to a terminal state such that (5.10) is satisfied. The minimization is such that the performance index (5.2) is minimized subject to the constraints in (5.11). The conditions are given by the following equations (for $t \in [0,\ T]$):

$$\dot{x}_t = f_t(x_t, u_t) \qquad \text{State equation} \tag{5.12}$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \text{Costate equation} \tag{5.13}$$

$$u_t = arg \min_{u_t \in \mathcal{U}_t} [H_t] \qquad \text{Optimality condition} \tag{5.14}$$

and the boundary conditions:

$$x_0 = \underline{x}_0 \qquad \lambda_T = \nu^T \frac{\partial}{\partial x} \psi_T(x_T) + \frac{\partial}{\partial x} \phi_T(x_T) \tag{5.15}$$

which is a split boundary condition. Due to the free terminal time, T, the solution must satisfy

$$\frac{\partial \phi_T}{\partial T} + H_T = 0 \tag{5.16}$$

which denoted ad the *Transversality condition*. □

Proof: Omitted □

If the problem is a maximization problem, then the minimization in (5.14) is changed into a maximization. Notice, the special version of the boundary condition for simple, simple partial and linear end points constraints given in (3.49), (3.50) and (3.51), respectively.

Example: 5.1.1 (Motion control) *The purpose of this example is to illustrate the method in a very simple situation, where the solution by intuition is known.*

Let us consider a perturbation of Example 3.5.1. Eventually see also the unconstrained continuous version in Example 2.3.1. The system here is the same, but the objective is changed.

The problem is to bring the system

$$\dot{x} = u_t \qquad x_0 = \underline{x}_0$$

from the initial position, $\underline{x}_0$, to the origin ($\underline{x}_T = 0$), in minimum time, while the control action (or the decision function) is bounded to

$$|u_t| \leq 1$$

The performance index is in this case

$$J = T \qquad = T + \int_0^T 0\ dt = 0 + \int_0^T 1\ dt$$

Notice, we can regards this as $\phi_T = T$, $L = 0$ or $\phi = 0$, $L = 1$ in our general notation. The Hamiltonian function is in this case (if we apply the first interpretation of cost function)

$$H = \lambda_t u_t$$

and the conditions are simply

$$\begin{aligned} \dot{x} &= u_t \\ -\dot{\lambda} &= 0 \\ u_t &= -sign(\lambda_t) \end{aligned}$$

with the boundary conditions:

$$x_0 = \underline{x}_0 \qquad x_T = 0 \qquad \lambda_T = \nu$$

Here we have introduced the Lagrange multiplier, ν, related to the end point constraint, $x_T = 0$. The Transversality condition is

$$1 + \lambda_T u_T = 0$$

As in Example 2.3.1 these equations are easily solved and it is also the costate equation here gives the key to the solution. Firstly, we notice that the costate is constant and equal to ν, i.e.

$$\lambda_t = \nu$$

If the control strategy

$$u_t = -sign(\nu)$$

is introduced in the state equation, we find

$$x_t = \underline{x}_0 - sign(\nu)\ t \qquad \text{and specially} \qquad 0 = \underline{x}_0 - sign(\nu)\ T$$

The last equation gives us

$$T = |\underline{x}_0| \qquad \text{and} \qquad sign(\nu) = sign(\underline{x}_0)$$

Now, we have found the sign of ν and is able to find its absolute value from the Transversality condition

$$1 - \nu\ sign(\nu) = 0$$

That means either is

$$|\nu| = 1$$

The two last equations can be combined into

$$\nu = sign(\underline{x}_0)$$

This results in the control strategy

$$u_t = -sign(\underline{x}_0)$$

and

$$x_t = x_0 - sign(\underline{x}_0)\ t$$

$\square$

Example: 5.1.2 Bang-Bang control *from (Lewis 1986b) p. 260. Consider a mass affected by a force. This is a second order system given by*

$$\frac{d}{dt}\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} v \\ u \end{bmatrix} \qquad\qquad \begin{bmatrix} z \\ v \end{bmatrix}_0 = \begin{bmatrix} \underline{z}_0 \\ \underline{v}_0 \end{bmatrix} \tag{5.17}$$

The state variable are the position, z, and the velocity, v, while the control action is the specific force (force divided by mass), u. This system is denoted as a double integrator (Control Theory) or a Newtonian system (Dynamic Optimization) due to the fact it obeys the second law of Newton. Assume the control action, i.e. the specific force is limited to

$$|u| \leq 1$$

while the objective is to take the system from its original state to the origin

$$\underline{x}_T = \begin{bmatrix} \underline{z}_T \\ \underline{v}_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

in minimum time. The performance index is accordingly

$$J = T$$

and the Hamilton function is

$$H = \lambda^z v + \lambda^v u$$

We can now write the conditions as the state equation, (5.17),

$$\frac{d}{dt}\begin{bmatrix} z \\ v \end{bmatrix} = \begin{bmatrix} v \\ u \end{bmatrix}$$

the costate equations

$$-\frac{d}{dt}\begin{bmatrix} \lambda^z \\ \lambda^v \end{bmatrix} = \begin{bmatrix} 0 \\ \lambda^z \end{bmatrix} \tag{5.18}$$

the optimality condition (Pontryagins maximum principle)

$$u_t = -sign(\lambda^v)$$

and the boundary conditions

$$\begin{bmatrix} z_0 \\ v_0 \end{bmatrix} = \begin{bmatrix} \underline{z}_0 \\ \underline{v}_0 \end{bmatrix} \qquad \begin{bmatrix} \underline{z}_T \\ \underline{v}_T \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad \begin{bmatrix} \lambda^z_T \\ \lambda^v_T \end{bmatrix} = \begin{bmatrix} \nu^z \\ \nu^v \end{bmatrix}$$

Notice, we have introduced the two Lagrange multipliers, ν^z and ν^v, related to the simple end points constraints in the states. The transversality condition is in this case

$$1 + H_T = 1 + \lambda^z_T v_T + \lambda^v_T u_T = 0 \tag{5.19}$$

From the Costate equation, (5.18), we can conclude that λ^z is constant and that λ^v is linear. More precisely we have

$$\lambda^z_t = \nu^z \qquad \lambda^v_t = \nu^v + \nu^z(T-t)$$

Since $v_T = 0$ the transversality conditions gives us

$$\lambda^v_T u_T = -1$$

but since u_t is saturated at ± 1 (for all t including of course the end point T) we only have two possible values for u_T (and λ^v_T), i.e.

- $u_T = \ \ 1$ *and* $\lambda_T^v = -1$
- $u_T = -1$ *and* $\lambda_T^v = \ \ 1$

The linear switching function, λ_t^v*, can only have one zero crossing or none depending on the initial state* z_0, v_0*. That leaves us with 4 possible situations as indicated in Figure 5.1.*

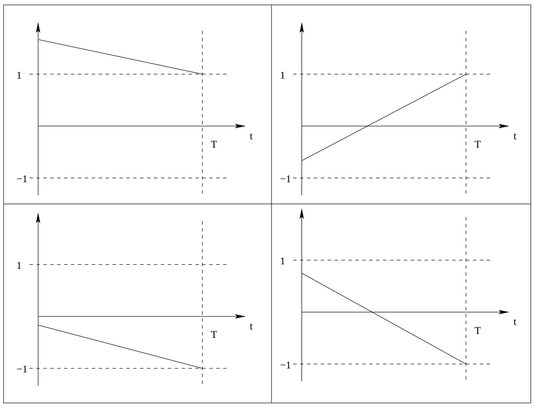

Figure 5.1. The switching function, λ_t^v has 4 different type of evolution.

To summarize, we have 3 unknown quantities, ν^z*,* ν^v *and* T *and 3 conditions to met,* $z_T = 0$ $v_T = 0$ *and* $\lambda_T^v = \pm 1$*. The solution can as previous mentioned be found by means of numerical methods. In this simple example we will however pursuit a analytical solution.*

If the control has a constant values $u = \pm 1$*, then the solution is simply*

$$\begin{aligned} v_t &= v_0 + ut \\ z_t &= z_0 + v_0 t + \frac{1}{2}ut^2 \end{aligned}$$

See Figure 5.2 (for $u = 1$*) and 5.3 (for* $u = -1$*).*

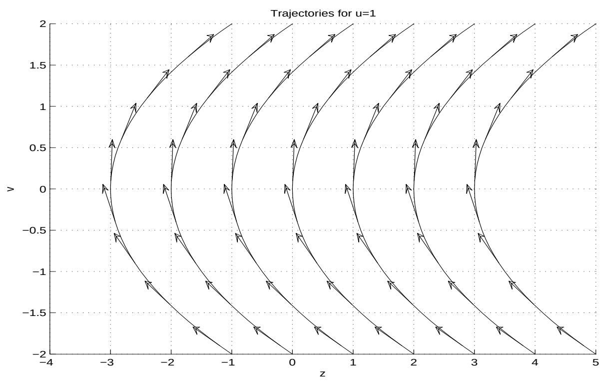

Figure 5.2. Phase plane trajectories for $u = 1$.

This a parabola passing through z_0*,* v_0*. If no switching occurs then the origin and the original point must lie on this parabola, i.e. satisfy the equations*

$$\begin{aligned} 0 &= v_0 + uT_f \\ 0 &= z_0 + v_0 T_f + \frac{1}{2}uT_f^2 \end{aligned} \tag{5.20}$$

where $T_f = T$ *(for this special case). This is the case if*

$$T_f = -\frac{v_0}{u} \quad \geq 0 \qquad\qquad z_0 = \frac{1}{2}\frac{v_0^2}{u} \tag{5.21}$$

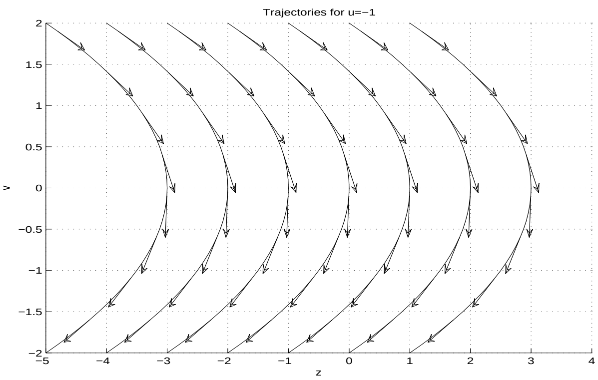

Figure 5.3. Phase plane trajectories for $u = -1$.

for either $u = 1$ or $u = -1$. In order to fulfill the first part and make the time T positive

$$u = -sign(v_0)$$

If $v_0 > 0$ then $u = -1$ and the initial point must lie on $z_0 = -\frac{1}{2}v_0^2$. This is the half (upper part of) the solid curve (for $v_0 > 0$) indicated in Figure 5.4. On the other hand, if $v_0 < 0$ then $u = 1$ and the initial point must lie on $z_0 = \frac{1}{2}v_0^2$. This is the other half (lower part of) the solid curve (for $v_0 < 0$) indicated in Figure 5.4. Notice, that in both cases we have a deacceleration, but in opposite directions.

The two branches on which the origin lies on the trajectory (for $u = \pm 1$) can be described by:

$$z_0 = \begin{cases} -\frac{1}{2}v_0^2 & for \;\; v_0 > 0 \\ \frac{1}{2}v_0^2 & for \;\; v_0 < 0 \end{cases} \quad = \quad -\frac{1}{2}v_0^2 sign(v_0)$$

There will be a switching unless the initial point lies on this curve, which in the literature is denoted as the switching curve. *See Figure 5.4.*

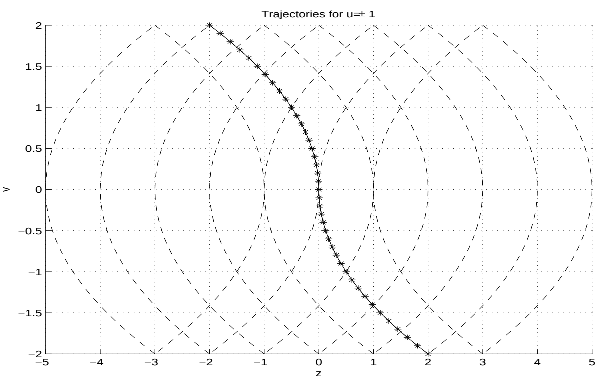

Figure 5.4. The switching curve (solid). The phase plane trajectories for $u = \pm 1$ are shown dashed.

If the initial point is not on the switching curve then there will be precisely one switch either from $u = -1$ to $u = 1$ or the other way around. From the phase plane trajectories in Figure 5.4 we can see that is the initial point is below the switching curve, i.e.

$$z_0 < -\frac{1}{2}v_0^2 sign(v_0)$$

the the input will have a period with $u_t = 1$ until we reach the switching curve where $u_t = -1$ for the rest of the period. The solution is (in this case) to accelerate the mass as much as possible and then, at the right instant of time, deaccelerate the mass as much as possible. Above the switching curve it is the reverse sequence of control input (first $u_t = -1$ and then $u_t = 1$), but it is a acceleration (in the negative direction) succeed by a deacceleration. This can be expressed as a state feedback law

$$u_t = \begin{cases} 1 & for \;\; z_0 < -\frac{1}{2}v_0^2 sign(v_0) \\ -1 & for \;\; z_0 = -\frac{1}{2}v_0^2 sign(v_0) \;\; and \;\; v > 0 \\ -1 & for \;\; z_0 > -\frac{1}{2}v_0^2 sign(v_0) \\ 1 & for \;\; z_0 = -\frac{1}{2}v_0^2 sign(v_0) \;\; and \;\; v < 0 \end{cases}$$

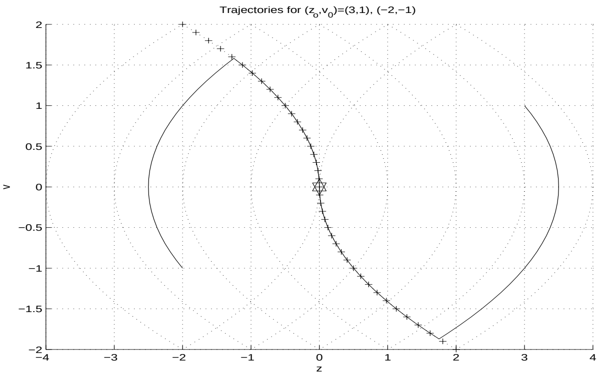

Figure 5.5. Optimal trajectories from two initial points.

Let us now focus on the optimal final time, T, and the switching time, T_s. Let us for the sake of simplicity assume the initial point is above the switching curve. The the initial control is $u_t = -1$ is applied to drive the state along the parabola passing through the initial point, $(z_0,\ v_0)$, to the switching curve, at which time T_s the control is switched to $u_t = 1$ to bring the state to the origin. Above the switching curve the evolution (for $u_t = -1$) of the states is given by

$$\begin{aligned} v_t &= v_0 - t \\ z_t &= z_0 + v_0 t - \frac{1}{2}t^2 \end{aligned}$$

which is valid until the switching curve given (for $v < 0$) by

$$z = \frac{1}{2}v^2$$

is met. This happens at T_s given by

$$z_0 + v_0 T_s - \frac{1}{2}T_s^2 = \frac{1}{2}(v_0 - T_s)^2$$

or

$$T_s = v_0 + \sqrt{z_0 + \frac{1}{2}v_0^2}$$

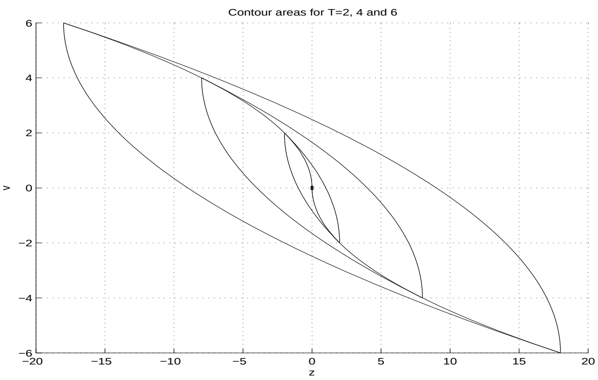

Figure 5.6. Contour of constant time to go.

Since the velocity at the switching point is

$$v_{T_s} = v_0 - T_s$$

the resting time to origin is (according to (5.21)) given by

$$T_f = -v_{T_s}$$

In total the optimal time can be written as

$$T = T_f + T_s = T_s - v_0 + T_s = v_0 + 2\sqrt{z_0 + \frac{1}{2}v_0^2}$$

The contours of constant time to go, T, are given by

$$\begin{aligned}
(v_0 - T)^2 &= 4(\frac{1}{2}v_0^2 + z_0) \qquad z_0 > -\frac{1}{2}v_0^2 sign(v_0) \\
(v_0 + T)^2 &= 4(\frac{1}{2}v_0^2 - z_0) \qquad z_0 < -\frac{1}{2}v_0^2 sign(v_0)
\end{aligned}$$

as indicated in Figure 5.6. □

Chapter 6

Dynamic Programming

Dynamic Programming dates from R.E. Bellman, who wrote the first book on the topic in 1957. It is a quite powerful tools which can be applied to a large variety of problems.

6.1 Discrete Dynamic Programming

Normally, in Dynamic Optimization the independent variable is the time, which (regrettably) is not reversible. In some cases, we apply methods from dynamic optimization on spatial problems, where the independent variable is a measure of the distance. But in these situations we associate the distance with time (i.e. think the distance is a monotonic function of time).

One of the basic properties of dynamic systems is causality, i.e. that a decision do not affects the previous states, but only the present and following states.

Example: 6.1.1 *(***Stagecoach problem** *from (Weber n.d.))*

A traveler wish to go from town A to town J through 4 stages with minimum travel distance. Firstly, from town A the traveler can choose to go to town B, C or D. Secondly the traveler can choose between a journey to E, F or G. After that, the traveler can go to town H or I and then finally to town J. See Figure 6.1 where The arcs are marked with the distances between towns.

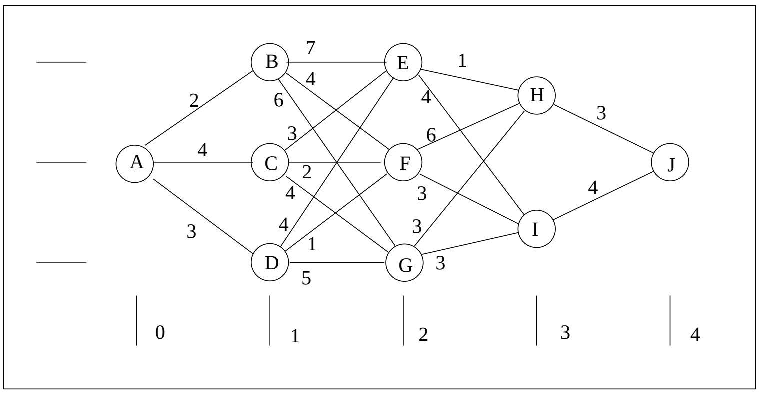

Figure 6.1. Road system for stagecoach problem. The arcs are marked with the distances between towns.

The solution to this simple problem can be found by means of dynamic programming, in which we are solving the problem backwards. Let $V(X)$ be the minimal distance required to reach J from X. Then clearly, $V(H) = 3$ and

$V(I) = 4$, which is related to stage 3. If we move on to stage 2, we find that

$$\begin{array}{lcll} V(E) & = & \min(1 + V(H), 4 + V(I)) = 4 & (E \to H) \\ V(F) & = & \min(6 + V(H), 3 + V(I)) = 7 & (F \to I) \\ V(G) & = & \min(3 + V(H), 3 + V(I)) = 6 & (G \to H) \end{array}$$

For stage 1 we have

$$\begin{array}{lcll} V(B) & = & \min(7 + V(E), 4 + V(F), 6 + V(G)) = 11 & (B \to E, B \to F) \\ V(C) & = & \min(3 + V(E), 2 + V(F), 4 + V(G)) = 7 & (C \to E) \\ V(D) & = & \min(4 + V(E), 1 + V(F), 5 + V(G)) = 8 & (D \to E, D \to F) \end{array}$$

Notice, the minimum is not unique. Finally, we have in stage 0 that

$$V(A) = \min(2 + V(B), 4 + V(C), 3 + V(D)) = 11 \qquad (A \to C, A \to D)$$

where the optimum (again) is not unique. We have not in a recursive manner found the shortest path $A \to C \to E \to H \to J$, which has the length 11. Notice, the solution is not unique. Both $A \to D \to E \to H \to J$ and $A \to D \to F \to I \to J$ are optimal solutions (with a path with length 11). □

6.1.1 Unconstrained Dynamic Programming

Let us now focus on the problem of controlling the system,

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{6.1}$$

i.e. to find a sequence of decisions $u_i \;\; i = 0, \; 1, \ldots \; N$ which takes the system from the initial state x_0 along a trajectory, such that the cost function

$$J = \phi(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{6.2}$$

is minimized. This is the free dynamic optimization problem. We will later on see how constrains easily are included in the setup.

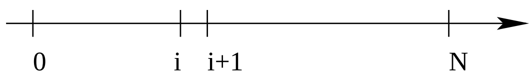

Figure 6.2. The discrete time axis.

Introduce the notation u_i^k for the sequence of decision from instant i to instant k. Let us consider the truncated performance index

$$J_i(x_i, u_i^{N-1}) = \phi(x_N) + \sum_{k=i}^{N-1} L_k(x_k, u_k)$$

which is a function of x_i and the sequence u_i^{N-1}, due to the fact that given x_i and the sequence u_i^{N-1} we can use the state equation, (6.1), to determine the state sequence $x_k \;\; k = i+1, \; \ldots \; N$. It is quite easy to see that

$$J_i(x_i, u_i^{N-1}) = L_i(x_i, u_i) + J_{i+1}(x_{i+1}, u_{i+1}^{N-1}) \tag{6.3}$$

and that in particular

$$J = J_0(x_0, u_0^{N-1})$$

where J is the performance index from (6.2). The Bellman function is defined as the optimal performance index, i.e.

$$V_i(x_i) = \min_{u_i^{N-1}} J_i(x_i, u_i^{N-1}) \tag{6.4}$$

and is a function of the present state, x_i. Notice, that in particular

$$V_N(x_N) = \phi_N(x_N)$$

We have the following Theorem, which gives a sufficient condition.

Theorem 12: Consider the free dynamic optimization problem specified in (6.1) and (6.2). The optimal performance index, i.e. the Bellman function V_i, is given by the recursion

$$\boxed{V_i(x_i) = \min_{u_i} \Big[\; L_i(x_i, u_i) + V_{i+1}(x_{i+1}) \; \Big]} \tag{6.5}$$

with the boundary condition

$$V_N(x_N) = \phi_N(x_N) \tag{6.6}$$

The functional equation, (6.5), is denoted as the Bellman equation and $J^* = V_0(x_0)$. □

Proof: The definition of the Bellman function in conjunction with the recursion (6.3) gives:

$$\begin{aligned} V_i(x_i) &= \min_{u_i^{N-1}} J_i(x_i, u_i^{N-1}) \\ &= \min_{u_i^{N-1}} \Big[\, L_i(x_i, u_i) + J_{i+1}(x_{i+1}, u_{i+1}^{N-1}) \, \Big] \end{aligned}$$

Since u_{i+1}^{N-1} do not affect L_i we can write

$$V_i(x_i) = \min_{u_i} \left[\; L_i(x_i, u_i) + \min_{u_{i+1}^{N-1}} J_{i+1}(x_{i+1}, u_{i+1}^{N-1}) \; \right]$$

The last term is nothing but V_{i+1}, due to the definition of the Bellman function. The boundary condition, (6.6), is also given by the definition of the Bellman function. □

If the state equation is applied the Bellman recursion can also be stated as

$$V_i(x_i) = \min_{u_i} \Big[\; L_i(x_i, u_i) + V_{i+1}(f_i(x_i, u_i)) \; \Big] \qquad V_N(x_N) = \phi_N(x_N) \tag{6.7}$$

Notice, if we have a maximization problem the minimization in (6.5) (or in (6.7)) is substituted by a maximization.

Example: 6.1.2 (Simple LQ problem) *The purpose of this example is to illustrate the application of dynamic programming in connection to continuous unconstrained dynamic optimization. Compare e.g. with Example 2.1.2 on page 15.*

The problem is bring the system

$$x_{i+1} = a x_i + b u_i \qquad\qquad x_0 = \underline{x}_0$$

from the initial state along a trajectory such the performance index

$$J = p x_N^2 + \sum_{i=0}^{N-1} q x_i^2 + r u_i^2$$

is minimized. (Compared with Example 2.1.2 the performance index is here multiplied with a factor of 2 in order to obtain a simpler notation). In the boundary we have

$$V_N = p x_N^2$$

Inspired of this, we will try the candidate function

$$V_i = s_i x_i^2$$

The Bellman equation gives

$$s_i x_i^2 = \min_{u_i} \Big[q x_i^2 + r u_i^2 + s_{i+1} x_{i+1}^2 \Big]$$

or with the state equation inserted

$$s_i x_i^2 = \min_{u_i} \Big[q x_i^2 + r u_i^2 + s_{i+1} (a x_i + b u_i)^2 \Big] \tag{6.8}$$

The minimum is obtained for

$$u_i = -\frac{a b s_{i+1}}{r + b^2 s_{i+1}} x_i \tag{6.9}$$

which inserted in (6.8) results in:

$$s_i x_i^2 = \Big[q + a^2 s_{i+1} - \frac{a^2 b^2 s_{i+1}^2}{r + b^2 s_{i+1}} \Big] x_i^2$$

The candidate function satisfies the Bellman equation if

$$s_i = q + a^2 s_{i+1} - \frac{a^2 b^2 s_{i+1}^2}{r + b^2 s_{i+1}} \qquad s_N = p \tag{6.10}$$

which is the (scalar version of the) Riccati equation. The solution (as in Example 2.1.2) consists of the backward recursion in (6.10) and the control law in (6.9). □

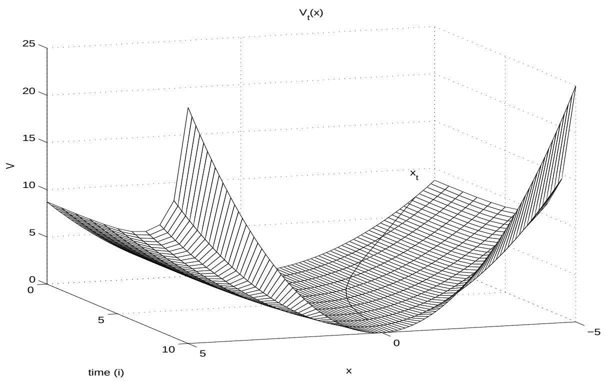

Figure 6.3. Plot of $V_t(x)$ for $a = 0.98$, $b = 1$, $q = 0.1$ and $r = p = 1$. The trajectory for x_t is plotted (on the surface of $V_t(x)$) for $x_0 = -4.5$.

The method applied in Example 6.1.2 can be generalized. In the example we made a qualified guess on the Bellman function. Notice, we made a guess on type of function phrased in a (number of) unknown parameter(s). Then we checked, if the Bellman equation was fulfilled. This check ended up in a (number of) recursion(s) for the parameter(s).

It is possible to establish a close connection between the Bellman equation and the Euler-Lagrange equations. Consider the minimization in (6.5). The necessary condition for minimum is

$$0^T = \frac{\partial L_i}{\partial u_i} + \frac{\partial V_{i+1}}{\partial x_{i+1}} \frac{\partial x_{i+1}}{\partial u_i}$$

Introduce the sensitivity

$$\lambda_i^T = \frac{\partial}{\partial x_i} V_i(x_i)$$

which we later on will recognize as the costate or the adjoint state vector. This has actually motivated the choice of symbol. If the sensitivity is applied the stationarity condition is simply

$$0^T = \frac{\partial L_i}{\partial u_i} + \lambda_{i+1}^T \frac{\partial f_i}{\partial u_i}$$

or

$$0^T = \frac{\partial}{\partial u_i} H_i \tag{6.11}$$

if we use the definition of the Hamiltonian function

$$H_i = L_i(x_i, u_i) + \lambda_{i+1}^T f_i(x_i, u_i) \tag{6.12}$$

On the optimal trajectory (i.e. with the optimal control applied) the Bellman function evolve according to

$$V_i(x_i) = L_i(x_i, u_i) + V_{i+1}(f_i(x_i, u_i))$$

or if we apply the chain rule

$$\lambda_i^T = \frac{\partial L_i}{\partial x} + \frac{\partial V_{i+1}}{\partial x} \frac{\partial}{\partial x} f_i \qquad \lambda_N^T = \frac{\partial}{\partial x} \phi_N(x_N)$$

or

$$\lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad \lambda_N^T = \frac{\partial}{\partial x_N} \phi_N(x_N) \tag{6.13}$$

We notice that the last two equations, (6.11) and (6.13), together with the dynamics in (6.1) precisely is the Euler-Lagrange equation in (2.8).

6.1.2 Constrained Dynamic Programming

In this section we will focus on the problem when the decisions and the state are constrained in the following manner

$$u_i \in \mathcal{U}_i \qquad x_i \in \mathcal{X}_i \tag{6.14}$$

The problem consist in bringing the system

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{6.15}$$

from the initial state along a trajectory satisfying the constraint in (6.14) and such that the performance index

$$J = \phi_N(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{6.16}$$

is minimized.

We have already met such a type of problem. In Example 6.1.1, on page 52, both the decisions and the state was constrained to a discrete and a finite set. It is quite easy to see that in the case of constrained control actions the minimization in (6.16) has to be subject to these constraints. However, if the state also is constrained, then the minimization in (6.16) is further constrained. This is due to the fact that a decision u_i has to ensure the future state trajectories is inside the feasible state area and there exists future feasible decisions. Let us define the feasible state area by the recursion

$$\mathscr{D}_i = \{\ x_i \in \mathcal{X}_i \mid \exists\ u_i \in \mathcal{U}_i : \quad f_i(x_i, u_i) \in \mathscr{D}_{i+1}\ \} \qquad \mathscr{D}_N = \mathcal{X}_N$$

The situation is (as all ways) less complicated in the end of the period (i.e. for $i = N$) where we do not have to take the future into account and then $\mathscr{D}_N = \mathcal{X}_N$. It can noticed, that the recursion for $\mathscr{D}_i$ just states, that the feasible state area is the set for which, there is a decision which bring the system to a feasible state area in the next interval. As a direct consequence of this the decision is constrained to decision which bring the system to a feasible state area in the next interval. Formally, we can define the feasible control area as

$$\mathcal{U}_i^*(x_i) = \{\ u_i \in \mathcal{U}_i : \quad f_i(x_i, u_i) \in \mathscr{D}_{i+1}\ \}$$

Theorem 13: Consider the dynamic optimization problem specified in (6.14) - (6.16). The optimal performance index, i.e. the Bellman function V_i, is for $x_i \in \mathcal{D}_i$ given by the recursion

$$\boxed{V_i(x_i) = \min_{u_i \in \mathcal{U}_i^*} \Big[\ L_i(x_i, u_i) + V_{i+1}(x_{i+1}) \ \Big] \qquad V_N(x_N) = \phi_N(x_N) \qquad (6.17)}$$

The optimization in (6.17) is constrained to

$$\mathcal{U}_i^*(x_i) = \{ \ u_i \in \mathcal{U}_i : \quad f_i(x_i, u_i) \in \mathscr{D}_{i+1} \ \} \tag{6.18}$$

where the feasible state area is given by the recursion

$$\mathscr{D}_i = \{ \ x_i \in \mathcal{X}_i \mid \exists \ u_i \in \mathcal{U}_i : \quad f_i(x_i, u_i) \in \mathscr{D}_{i+1} \ \} \qquad \mathscr{D}_N = \mathcal{X}_N \tag{6.19}$$

□

Example: 6.1.3 (**Stagecoach problem II** *Consider a variation of the Stagecoach problem in Example 6.1.1. See Figure 6.4.*

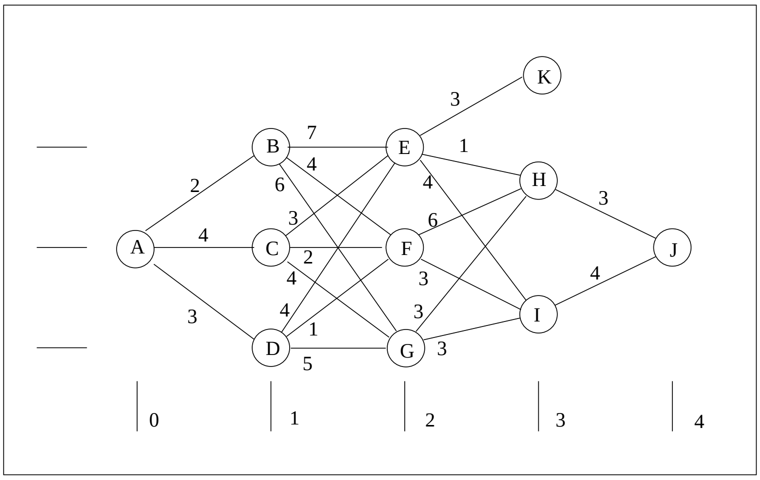

Figure 6.4. Road system for stagecoach problem. The arcs are marked with the distances between towns.

It is easy to realize that

$$\mathcal{X}_4 = \{J\} \qquad \mathcal{X}_3 = \{H, I, K\} \qquad \mathcal{X}_2 = \{E, F, G\} \qquad \mathcal{X}_1 = \{B, C, D\} \qquad \mathcal{X}_0 = \{A\}$$

However, since there is no path from K to J

$$\mathcal{D}_4 = \{J\} \qquad \mathcal{D}_3 = \{H, I\} \qquad \mathcal{D}_2 = \{E, F, G\} \qquad \mathcal{D}_1 = \{B, C, D\} \qquad \mathcal{D}_0 = \{A\}$$

□

Example: 6.1.4 Optimal stepping (DD). *This is a variation of Example 6.1.2 where the sample space is discrete (and the dynamic and performance index are particular simple). Consider the system*

$$x_{i+1} = x_i + u_i \qquad x_0 = 2$$

the performance index

$$J = x_N^2 + \sum_{i=0}^{N-1} x_i^2 + u_i^2 \qquad \textit{with} \qquad N = 4$$

and the constraints

$$u_i \in \{-1, \ 0, \ 1\} \qquad x_i \in \{-2, \ -1, \ 0, \ 1, \ 2\}$$

Firstly, we establish $V_4(x_4)$ as in the following table

x_4	V_4
-2	4
-1	1
0	0
1	1
2	4

We are now going to investigate the different components in the Bellman equation (6.17) for $i = 3$. The combination of x_3 and u_3 determines the following state x_4 and consequently the $V_4(x_4)$ contribution in (6.17).

x_4	u_3		
x_3	-1	0	1
-2	-3	-2	-1
-1	-2	-1	0
0	-1	0	1
1	0	1	2
2	1	2	3

$V_4(x_4)$	u_3		
x_3	-1	0	1
-2	∞	4	1
-1	4	1	0
0	1	0	1
1	0	1	4
2	1	4	∞

Notice, an invalid combination of x_3 and u_3 (resulting in an x_4 outside the range) is indicated with ∞ in the tableau. The combination of x_3 and u_3 also determines the instantaneous loss L_3.

L_3	u_3		
x_3	-1	0	1
-2	5	4	5
-1	2	1	2
0	1	0	1
1	2	1	2
2	5	4	5

If we add up the instantaneous loss and V_4 we have a tableau in which we for each possible value of x_3 can perform the minimization in (6.17) and determine the optimal value for the decision and the Bellman function (as function of x_3).

$L_3 + V_4$	u_3			V_3	u_3^*
x_3	-1	0	1		
-2	∞	8	6	6	1
-1	6	2	2	2	0
0	2	0	2	0	0
1	2	2	6	2	-1
2	6	8	∞	6	-1

Knowing $V_3(x_3)$ we have one of the components for $i = 2$. In this manner we can iterate backwards and finds:

$L_2 + V_3$	u_2			V_2	u_2^*
x_2	-1	0	1		
-2	∞	10	7	7	1
-1	8	3	2	2	1
0	3	0	3	0	0
1	2	3	8	2	-1
2	7	10	∞	7	-1

$L_1 + V_2$	u_1			V_1	u_1^*
x_1	-1	0	1		
-2	∞	11	7	7	1
-1	9	3	2	2	1
0	3	0	3	0	0
1	2	3	9	2	-1
2	7	11	∞	7	-1

Iterating backwards we end with the following tableau.

$L_0 + V_1$	u_0			V_0	u_0^*
x_0	-1	0	1		
-2	∞	11	7	7	1
-1	9	3	2	2	1
0	3	0	3	0	0
1	2	3	9	2	-1
2	7	11	∞	7	-1

With $x_0 = 2$ we can trace forward and find the input sequence $-1,\ -1,\ 0,\ 0$ which give (an optimal) performance equal 7. Since $x_0 = 2$ we actually only need to determine the row corresponding to $x_0 = 2$. The full tableau gives us, however, information on the sensitivity of the solution with respect to the initial state. $\square$

Example: 6.1.5 Optimal stepping (DD). *Consider the system from Example 6.1.4, but with the constraints that $x_4 = 1$. The task is to bring the system*

$$x_{i+1} = x_i + u_i \qquad x_0 = 2 \qquad x_4 = 1$$

along a trajectory such the performance index

$$J = x_4^2 + \sum_{i=0}^{3} x_i^2 + u_i^2$$

is minimized if

$$u_i \in \{-1,\ 0,\ 1\} \qquad x_i \in \{-2,\ -1,\ 0,\ 1,\ 2\}$$

In this case we assign ∞ with an invalid state

x_4	V_4
-2	∞
-1	∞
0	∞
1	1
2	∞

and further iteration gives

Furthermore:

$L_3 + V_4$	u_3			V_3	u_3^*
x_3	-1	0	1		
-2	∞	∞	∞	∞	
-1	∞	∞	∞	∞	
0	∞	∞	2	2	1
1	∞	2	∞	2	0
2	6	∞	∞	6	-1

$L_2 + V_3$	u_2			V_2	u_2^*
x_2	-1	0	1		
-2	∞	∞	∞	∞	
-1	∞	∞	4	4	1
0	∞	2	3	2	0
1	4	3	8	3	0
2	7	10	∞	7	-1

and

$L_1 + V_2$	u_1			V_1	u_1^*
x_1	-1	0	1		
-2	∞	∞	9	9	1
-1	∞	5	4	4	1
0	5	2	4	2	0
1	4	4	9	4	-1
2	8	11	∞	8	-1

$L_0 + V_1$	u_0			V_0	u_0^*
x_0	-1	0	1		
-2	∞	13	9	9	1
-1	11	5	4	4	1
0	5	2	5	2	0
1	4	5	10	4	-1
2	9	12	∞	9	-1

With $x_0 = 2$ we can iterate forward and find the optimal input sequence $-1,\ -1,\ 0,\ 1$ which is connected to a performance index equal 9.

□

6.1.3 Stochastic Dynamic Programming (D)

In this section we will consider the problem of controlling a dynamic system in which there are involved some stochastics. A *stochastic variable* is a quantity which can not predicted precisely. The description of stochastic variable involve distribution function and (if it exists) density function.

We will focus on control of stochastic dynamic systems described as

$$x_{i+1} = f_i(x_i, u_i, w_i) \qquad x_0 = \underline{x}_0 \tag{6.20}$$

where w_i is a stochastic process (i.e. a stochastic variable indexed by the time, i).

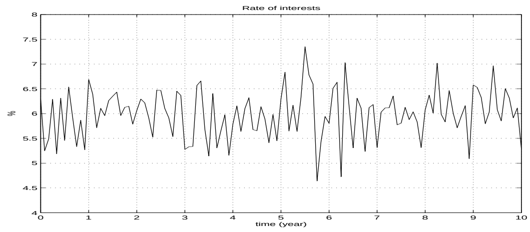

Figure 6.5. The evolution of the rate of interests.

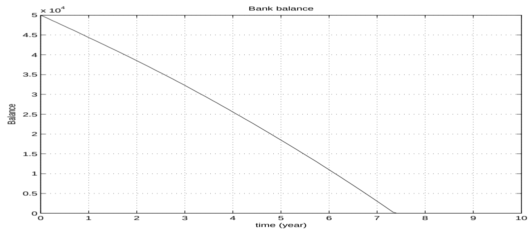

Figure 6.6. The evolution of the bank balance for one particular sequence of rate of interest (and constant down payment $u_i = 700$).

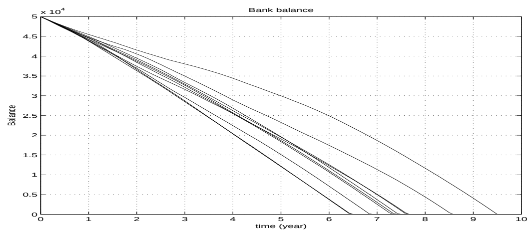

Figure 6.7. The evolution of the bank balance for 10 different sequences of interest rate (and constant down payment $u_i = 700$).

Example: **6.1.6** **The bank loan:** *Consider a bank loan initially of size $\underline{x}_0$. If the rate of interest is r per month then balance will develop according to*

$$x_{i+1} = (1 + r)x_i - u_i$$

Here u_i is the monthly down payment on the loan. If the rate of interest is not a constant quantity and especially if it can not be precisely predicted, then r is a time varying quantity. That means the balance will evolve according to

$$x_{i+1} = (1 + r_i)x_i - u_i$$

This is typical example on a stochastic system.

□

The performance index can also have a stochastic component. We will in this section work with performance index which has a stochastic dependence such as

$$J_s = \phi_N(x_N, w_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i, w_i) \tag{6.21}$$

The task is to find a sequence of decisions or control actions, u_i, $i = 1,\ 0,\ ...\ N - 1$ such that the system is taken along a trajectory such that the performance index is minimized. The problem in this context is however, what we mean by an optimal strategy, when stochastic variables are involved. More directly, how can we rank performance indexes if they are stochastic variable. If we apply an average point of view, then the performance has to be ranked by expected value. That means we have to find a sequence of decisions such that the expected value is minimized, or more precisely such that

$$J = \mathbf{E}\Big\{ \phi_N(x_N, w_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i, w_i) \Big\} \qquad \Big(= \mathbf{E}\Big\{ J_s \Big\} \Big) \tag{6.22}$$

is minimized. The truncated performance index will in this case besides the present state, x_i, and the decision sequence, u_i^{N-1} also depend on the future disturbances, w_k $, k = i,\ i + 1,\ ...\ N$ (or in short on w_i^N). The *stochastic Bellman function* is defined as

$$V_i(x_i) = \min_{u_i} \mathbf{E}\Big\{ J_i(x_i, u_i^{N-1}, w_i^N) \Big\} \tag{6.23}$$

The boundary is interesting in that sense that

$$V_N(x_N) = \mathbf{E}\Big\{ \phi_N(x_N, w_N) \Big\}$$

Theorem 14: Consider the dynamic optimization problem specified in (6.20) and (6.22). The optimal performance index, i.e. the Bellman function V_i, is given by the recursion

$$\boxed{V_i(x_i) = \min_{u_i \in \mathcal{U}_i^*} \mathbf{E}_{w_i} \{ \ L_i(x_i, u_i, w_i) + V_{i+1}(x_{i+1}) \ \}} \tag{6.24}$$

with the boundary condition

$$V_N(x_N) = \mathbf{E}\Big\{ \phi_N(x_N, w_N) \Big\} \tag{6.25}$$

The functional equation, (6.5), is denoted as the Bellman equation and $J^* = V_0(x_0)$. □

Proof: Omitted □

In (6.24) it should be noticed that $x_{i+1} = f_i(x_i, u_i, w_i)$ which a.o. depend on w_i.

Example: 6.1.7 *Consider a situation in which the stochastic vector, w_i can take a finite number, r, of distinct values, i.e.*

$$w_i \in \{\ w_i^1,\ w_i^2,\ \dots\ w_i^r\ \}$$

with certain probabilities

$$p_i^k = P\Big\{w_i = w_i^k\Big\} \qquad k = 1,\ 2,\ \dots\ r$$

(Do not let yourself confuse by the sample space index (k) with anything else). The stochastic Bellman equation can be expressed as

$$V_i(x_i) = \min_{u_i} \sum_{k=1}^{r} p_i^k \Big[L_i(x_i, u_i, w_i^k) + V_{i+1}(f_i(x_i, u_i, w_i^k))\Big]$$

with boundary condition

$$V_N(x_N) = \sum_{k=1}^{r} p_N^k \phi_N(x_N, w_N^k)$$

If the state space is discrete and finite we can produce a table for V_N.

x_N	V_N

The entries will be a weighted sum of the type

$$V_N(x_N) = p_N^1 \phi_N(x_N, w_N^1) + p_N^2 \phi_N(x_N, w_N^2) + \ \dots$$

When this table has been established we can move on to $i = N-1$ and determine a table containing

$$W_{N-1}(x_{N-1}, u_{N-1}) \triangleq \sum_{k=1}^{r} p_{N-1}^k \Big[L_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}^k) + V_N(f_{N-1}(x_{N-1}, u_{N-1}, w_{N-1}^k))\Big]$$

W_{N-1}	u_{N-1}
x_{N-1}	. . .
.	
.	
.	
.	
.	

For each possible values of x_{N-1} we can find the minimal values i.e. $V_{N-1}(x_{N-1})$ and the optimizing decision, u_{N-1}^*.

W_{N-1}	u_{N-1}	V_{N-1}	u_{N-1}^*
x_{N-1}	. . .		
.			
.			
.			
.			
.			

After establishing the table for $V_{N-1}(x_{N-1})$ we can repeat the procedure for $N-2$, $N-3$ and so forth until $i = 0$. □

Example: 6.1.8 (Optimal stepping (DD)). *Let us consider a stochastic version of Example 6.1.4. Consider the system*

$$x_{i+1} = x_i + u_i + w_i \qquad x_0 = 2,$$

where the stochastic disturbance, w_i has a discrete sample space

$$w_i \in \Big\{-1 \quad 0 \quad 1\Big\}$$

and has a distribution given by:

p_i^k	w_i		
x_i	-1	0	1
-2	0	$\frac{1}{2}$	$\frac{1}{2}$
-1	0	$\frac{1}{2}$	$\frac{1}{2}$
0	$\frac{1}{2}$	0	$\frac{1}{2}$
1	$\frac{1}{2}$	$\frac{1}{2}$	0
2	$\frac{1}{2}$	$\frac{1}{2}$	0

That means for example that w_i takes the value 1 with probability $\frac{1}{2}$ if $x_i = -2$. We have the constraints on the decisions

$$u_i \in \{-1,\ 0,\ 1\}$$

except if $x_i = 2$, then $u_i = 1$ is not allowed. Similarly, $u_i = -1$ is not allowed if $x_i = -2$. The states are constrained to

$$x_i \in \{-2,\ -1,\ 0,\ 1,\ 2\}$$

The cost function (as in Example 6.1.4) given by

$$J = x_N^2 + \sum_{i=0}^{N-1} x_i^2 + u_i^2 \qquad \text{with} \qquad N = 4$$

Firstly, we establish $V_4(x_4)$ as in the following table

x_4	V_4
-2	4
-1	1
0	0
1	1
2	4

Next we can establish the $W_3(x_3, u_3)$ function (see Example 6.1.7) for each possible combination of x_3 and u_3. In this case w_3 can take 3 possible values with a certain probability as given in the table above. Let us denote these values as w_3^1, w_3^2 and w_3^3 and the respective probabilities as p_3^1, p_3^2 and p_3^3. Then

$$W_3(x_3, u_3) = p_3^1\Big(x_3^2 + u_3^2 + V_4(x_3 + u_3 + w_3^1)\Big) + p_3^2\Big(x_3^2 + u_3^2 + V_4(x_3 + u_3 + w_3^2)\Big) + p_3^3\Big(x_3^2 + u_3^2 + V_4(x_3 + u_3 + w_3^3)\Big)$$

or

$$W_3(x_3, u_3) = x_3^2 + u_3^2 + p_3^1 V_4(x_3 + u_3 + w_3^1) + p_3^2 V_4(x_3 + u_3 + w_3^2) + p_3^3 V_4(x_3 + u_3 + w_3^3)$$

The numerical values are given in the table below

W_3	u_3		
x_3	-1	0	1
-2	∞	6.5	5.5
-1	4.5	1.5	2.5
0	3	1	3
1	2.5	1.5	4.5
2	5.5	3.5	∞

For example the cell corresponding to $x_3 = 0$, $u_3 = -1$ is determined by

$$W_3(0, -1) = 0^2 + (-1)^2 + \frac{1}{2}4 + \frac{1}{2}0 = 3$$

Due to the fact that x_4 takes the values $-1 - 1 = -2$ and $-1 + 1 = 0$ with same probability ($\frac{1}{2}$). Further examples are

$$W_3(-1, -1) = (-1)^2 + (-1)^2 + \frac{1}{2}4 + \frac{1}{2}1 = 4.5$$

$$W_3(-1, 0) = (-1)^2 + (0)^2 + \frac{1}{2}1 + \frac{1}{2}0 = 1.5$$

$$W_3(-1, 1) = (-1)^2 + (1)^2 + \frac{1}{2}0 + \frac{1}{2}1 = 2.5$$

With the table for W_3 it is quite easy to perform the minimization in (6.24). The results are listed in the table below.

W_3	u_3			$V_3(x_3)$	$u_3^*(x_3)$
x_3	-1	0	1		
-2	∞	6.5	5.5	5.5	1
-1	4.5	1.5	2.5	1.5	0
0	3	1	3	1	0
1	2.5	1.5	4.5	1.5	0
2	5.5	3.5	∞	3.5	0

By applying this method we can iterate the solution backwards and find.

W_2	u_2			$V_2(x_2)$	$u_2^*(x_2)$
x_2	-1	0	1		
-2	∞	7.5	6.25	6.25	1
-1	5.5	2.25	3.25	2.25	0
0	4.25	1.5	3.25	1.5	0
1	3.25	2.25	4.5	2.25	0
2	6.25	6.5	∞	6.25	-1

W_1	u_1			$V_1(x_1)$	$u_1^*(x_1)$
x_1	-1	0	1		
-2	∞	8.25	6.88	6.88	1
-1	6.25	2.88	3.88	2.88	0
0	4.88	2.25	4.88	2.25	0
1	3.88	2.88	6.25	2.88	0
2	6.88	8.25	∞	6.88	-1

In the last iteration we only need one row (for $x_0 = 2$) and can from this state the optimal decision.

W_0	u_0			$V_0(x_0)$	$u_0^*(x_0)$
x_0	-1	0	1		
2	7.56	8.88	∞	7.56	-1

$\square$

It should be noticed, that state space and decision space in the previous examples are discrete. If the state space is continuous, then the method applied in the examples be used as an approximation if we use a discrete grid covering the relevant part of the state space.

6.2 Continuous Dynamic Programming

Consider the problem of finding the input function u_t, $t \in \mathbb{R}$, that takes the system

$$\dot{x} = f_t(x_t, u_t) \qquad x_0 = \underline{x}_0 \qquad t \in [0,\ T] \tag{6.26}$$

from its initial state along a trajectory such that the cost function

$$J = \phi_T(x_T) + \int_0^T L_t(x_t, u_t)\ dt \tag{6.27}$$

is minimized. As in the discrete time case we can define the truncated performance index

$$J_t(x_t, u_t^T) = \phi_T(x_T) + \int_t^T L_s(x_s, u_s)\ ds$$

which depend on the point of truncation, on the state, x_t, and on the whole input function, u_t^T, in the interval from t to the end point, T. The optimal performance index, i.e. the Bellman function is defined by

$$V_t(x_t) = \min_{u_t^T} \left[\ J_t(x_t, u_t^T)\ \right]$$

We have the following theorem, which states a sufficient condition.

Theorem 15: Consider the free dynamic optimization problem specified by (6.26) and (6.27). The optimal performance index, i.e. the Bellman function $V_t(x_t)$, satisfy the equation

$$-\frac{\partial V_t(x_t)}{\partial t} = \min_u \left[L_t(x_t, u_t) + \frac{\partial V_t(x_t)}{\partial x} f_t(x_t, u_t) \right] \tag{6.28}$$

with boundary conditions

$$V_T(x_T) = \phi_T(x_T) \tag{6.29}$$

□

Equation (6.28) is often denoted as the HJB (Hamilton, Jacobi, Bellman) equation.

Proof: In describe time we have the Bellman equation

$$V_i(x_i) = \min_{u_i} \left[\; L_i(x_i, u_i) + V_{i+1}(x_{i+1}) \; \right]$$

with the boundary condition

$$V_N(x_N) = \phi_N(x_N) \tag{6.30}$$

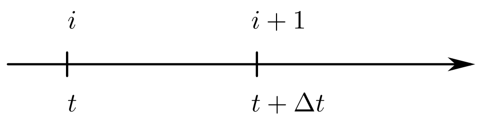

Figure 6.8. The continuous time axis

In continuous time i corresponds to t and $i+1$ to $t+\Delta t$, respectively. Then

$$V_t(x_t) = \min_u \left[\int_t^{t+\Delta t} L_t(x_t, u_t) \; dt + V_{t+\Delta t}(x_{t+\Delta t}) \right]$$

If we apply a Taylor expansion on $V_{t+\Delta t}(x_{t+\Delta t})$ and on the integral we have

$$V_t(x_t)i = \min_u \left[L_t(x_t, u_t)\Delta t + V_t(x_t) + \frac{\partial V_t(x_t)}{\partial x} f_t \Delta t + \frac{\partial V_t(x_t)}{\partial t} \Delta t \right]$$

Finally, we can collect the terms which do not depend on the decision

$$V_t(x_t) = V_t(x_t) + \frac{\partial V_t(x_t)}{\partial t} \Delta t + \min_u \left[Lt(x_t, u_t)\Delta t + \frac{\partial V_t(x_t)}{\partial x} f_t \Delta t \right]$$

In the limit $\Delta t \to 0$ we will have (6.28). The boundary condition, (6.29), comes directly from (6.30). □

Notice, if we have a maximization problem, then the minimization in (6.28) is substituted with a maximization.

If the definition of the Hamiltonian function

$$H_t = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t)$$

is used, then the HJB equation can also be formulated as

$$-\frac{\partial V_t(x_t)}{\partial t} = \min_{u_t} H_t(x_t, u_t, \frac{\partial V_t(x_t)}{\partial x})$$

Example: 6.2.1 (Motion control). *The purpose of this simple example is to illustrate the application of the HJB equation. Consider the system*

$$\dot{x}_t = u_t \qquad x_0 = \underline{x}_0$$

and the performance index

$$J = \frac{1}{2} p x_T^2 + \int_0^T \frac{1}{2} u_t^2 \; dt$$

The HJB equation, (6.28), gives:

$$-\frac{\partial V_t(x_t)}{\partial t} = \min_{u_t} \left[\frac{1}{2} u_t^2 + \frac{\partial V_t(x_t)}{\partial x} u_t \right]$$

The minimization can be carried out and gives a solution w.r.t. u_t which is

$$u_t = -\frac{\partial V_t(x_t)}{\partial x} \tag{6.31}$$

So if the Bellman function is known the control action or the decision can be determined from this. If the result above is inserted in the HJB equation we get

$$-\frac{\partial V_t(x_t)}{\partial t} = \frac{1}{2}\left[\frac{\partial V_t(x_t)}{\partial x}\right]^2 - \left[\frac{\partial V_t(x_t)}{\partial x}\right]^2 = -\frac{1}{2}\left[\frac{\partial V_t(x_t)}{\partial x}\right]^2$$

which is a partial differential equation with the boundary condition

$$V_T(x_T) = \frac{1}{2} p x_T^2$$

Inspired of the boundary condition we guess on a candidate function of the type

$$V_t(x) = \frac{1}{2} s_t x^2$$

where the explicit time dependence is in the time function, s_t. Since

$$\frac{\partial V}{\partial x} = s_t x \qquad \qquad \frac{\partial V}{\partial t} = \frac{1}{2} \dot{s} x^2$$

the following equation

$$-\frac{1}{2}\dot{s}_t x^2 = -\frac{1}{2}(s_t x)^2$$

must be valid for any x, i.e. we can find s_t by solving the ODE

$$\dot{s}_t = s_t^2 \qquad s_T = p$$

backwards. This is actually (a simple version of) the continuous time Riccati equation. The solution can be found analytically or by means of numerical methods. Knowing the function, s_t, we can find the control input from (6.31). □

It is possible to derived the (continuous time) Euler-Lagrange equations from the HJB equation.

Appendix **A**

A.1 Quadratic forms

In this section we will give a short resume of the concepts and the results related to positive definite matrices. If $x \in \mathbb{R}^n$ is a vector, then the squared Euclidian norm is obeying:

$$J = \|x\|^2 = x^T x \geq 0 \tag{A.1}$$

If A is a nonsignular matrix, then the vector Ax has a quadratic norm $x^\top A^\top Ax \geq 0$. Let $Q = A^\top A$ then

$$\|x\|_Q^2 = x^T Q x \geq 0 \tag{A.2}$$

and denote $\parallel x \parallel_Q$ as the square norm of x w.r.t.. Q.

Now, let S be a square matrix. We are now interested in the sign of the quadratic form:

$$J = x^\top S x \tag{A.3}$$

where J is a scalar. Any quadratic matrix can be decomposed in a symmetric part, S_s and a nonsymmetric part, S_a, i.e.

$$S = S_s + S_a \qquad S_s = \frac{1}{2}(S + S^\top) \qquad S_a = \frac{1}{2}(S - S^\top) \tag{A.4}$$

Notice:

$$S_s^\top = S_s \qquad S_a^\top = -S_a \tag{A.5}$$

Since the scalar, $x^\top S_a x$ fulfills:

$$x^\top S_a x = (x^\top S_a x)^\top = x^\top S_a^\top x = -x^\top S_a x \tag{A.6}$$

it is true that $x^\top S_a x = 0$ for any $x \in \mathbb{R}^n$, or that:

$$J = x^\top S x = x^\top S_s x \tag{A.7}$$

An analysis of the sign variation of J can then be carried out as an analysis of S_s, which (as a symmetric matrix) can be diagonalized.

A matrix S is said to be positive definite if (and only if) $x^\top S x > 0$ for all $x \in \mathbb{R}^n$ $x^x > 0$. Consequently, S is positive definite if (and only if) all eigen values are positive. A matrix, S, is positive semidefinite if $x^\top S x \geq 0$ for all $x \in \mathbb{R}^n$ $x^x > 0$. This can be checked by investigating if all eigen values are non negative. A similar definition exist for negative definite matrices. If J can take both negative and positive values, then S is denoted as indefinite. In that case the symmetric part of the matrix has both negative and positive eigenvalues.

We will now examine the situation

Example: A.1.1 *In the following we will consider some two dimensional problems. Let us start with a simple problem in which:*

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \qquad g = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad b = 0 \tag{A.8}$$

In this case we have

$$J = x_1^2 + x_2^2 \tag{A.9}$$

and the levels (domain in which the loss function J is equal c^2) are easily recognized as circles with center in origin and radius equal c. See Figure A.1, area a *and* surface a. □

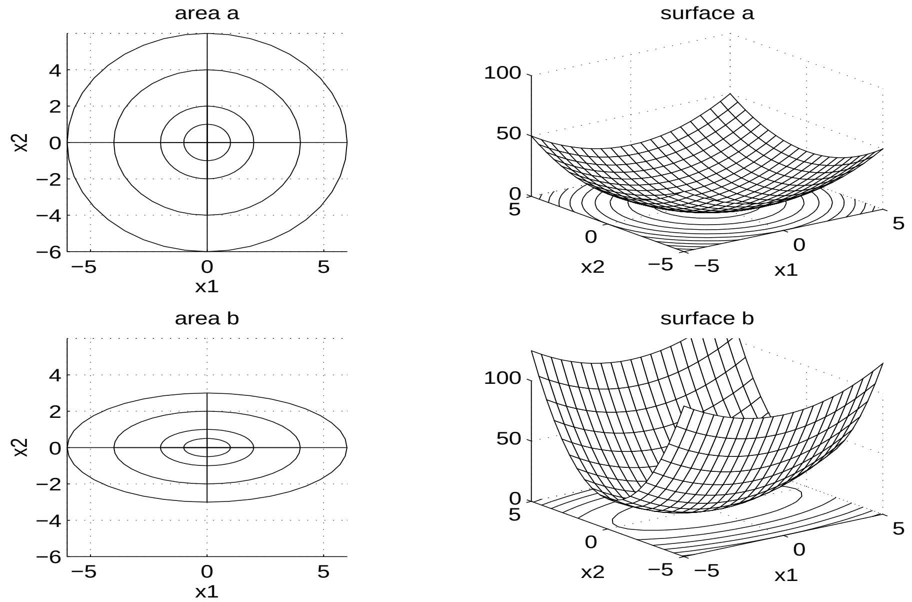

Figure A.1.

Example: A.1.2 *Let us continue the sequence of two dimensional problems in which:*

$$H = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix} \qquad g = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad b = 0 \tag{A.10}$$

The explicit form of the loss function J is:

$$J = x_1^2 + 4x_2^2 \tag{A.11}$$

and the levels (with $J = c^2$) is ellipsis with main directions parallel to the axis and length equal c and $\frac{c}{2}$, respectively. See Figure A.1, area b *and* surface b. □

Example: A.1.3 *Let us now continue with a little more advanced problem with:*

$$H = \begin{bmatrix} 1 & 1 \\ 1 & 4 \end{bmatrix} \qquad g = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \qquad b = 0 \tag{A.12}$$

In this case the situation is a little more difficult. If we perform an eigenvalue analysis of the symmetric part of H (which is H itself due to the fact H is symmetric), then we will find that:

$$H = VDV^\top \qquad V = \begin{bmatrix} 0.96 & 0.29 \\ -0.29 & 0.96 \end{bmatrix} \qquad D = \begin{bmatrix} 0.70 & 0 \\ 0 & 4.30 \end{bmatrix} \tag{A.13}$$

which means the column in V are the eigenvectors and the diagonal elements of D are the eigenvalues. Since the eigenvectors conform a orthogonal basis, $V^\top V = I$, we can choose to represent x in this coordinate system. Let ξ be the coordinates in relation to the column of V, then

$$x = V\xi \tag{A.14}$$

and

$$J = x^\top Hx = \xi^\top V^\top HV\xi = \xi^\top D\xi \tag{A.15}$$

We are hereby able to write the loss function as:

$$J = 0.7\xi_1^2 + 4.3\xi_2^2 \tag{A.16}$$

Notice the eigenvalues 0.7 *and* 4.3. *The levels (*$J = c^2$*) are recognized ad ellipsis with center in origin and main directions parallel to the eigenvectors. The length of the principal axis are* $\frac{c}{\sqrt{0.7}}$ *and* $\frac{c}{\sqrt{4.3}}$*, respectively. See Figure A.2* area c *and* surface c. □

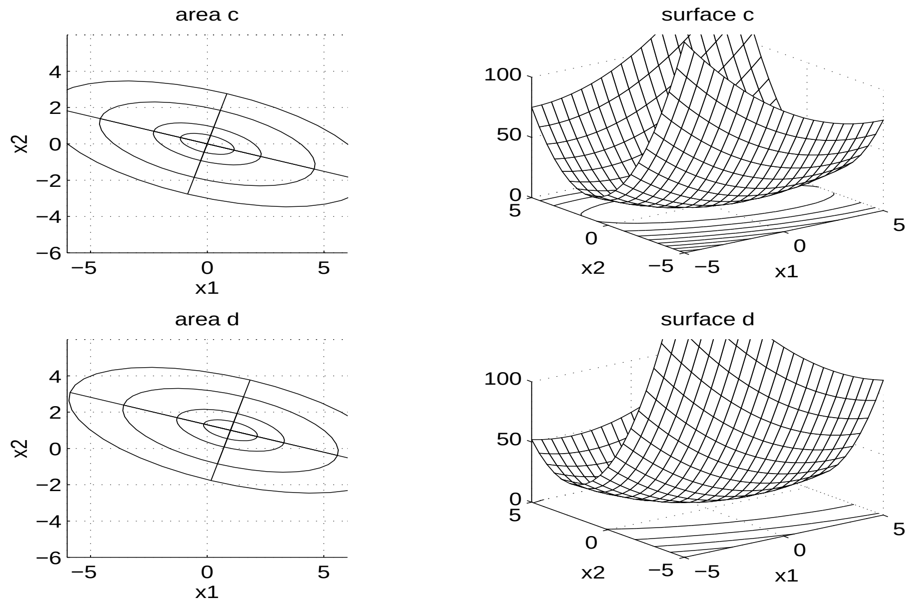

Figure A.2.

For the sake of simplify the calculus we are going to consider special versions of quadratic forms.

Lemma 3: The quadratic form

$$J = [Ax + Bu]^T S [Ax + Bu]$$

can be expressed as

$$J = \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{pmatrix} A^TSA & A^TSB \\ B^TSA & B^TSB \end{pmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

□

Proof: The proof is simply to express the loss function J as

$$J = z^T S z \qquad z = Ax + Bu = \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

or

$$J = \begin{bmatrix} x^T & u^T \end{bmatrix} \begin{bmatrix} A^T \\ B^T \end{bmatrix} S \begin{bmatrix} A & B \end{bmatrix} \begin{bmatrix} x \\ u \end{bmatrix}$$

or as stated in lemma 3 □

We have now studied properties of quadratic forms and a single lemma (3). Let us now focus on the problem of finding a minimum (or similar a maximum) in a quadratic form.

Lemma 4: Assume, u is a vector of decisions (or control actions) and x is a vector containing known state variables. Consider the quadratic form:

$$J = \begin{pmatrix} x^\top u^\top \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} \\ h_{12}^\top & h_{22} \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} \tag{A.17}$$

There exists not a minimum if h_{22} is **not** a positive semidefinite matrix. If h_{22} is **positive definite** then the quadratic form has a minimum for

$$u^* = -h_{22}^{-1} h_{12}^\top x \tag{A.18}$$

and the minimum is

$$J^* = x^\top S x \tag{A.19}$$

where

$$S = h_{11} - h_{12} h_{22}^{-1} h_{12}^\top \tag{A.20}$$

If h_{22} is only positive semidefinite then we have infinite many solutions with the same value of J. □

Proof: Firstly we have

$$\begin{aligned} J &= \begin{pmatrix} x^\top u^\top \end{pmatrix} \begin{pmatrix} h_{11} & h_{12} \\ h_{12}^\top & h_{22} \end{pmatrix} \begin{pmatrix} x \\ u \end{pmatrix} && \text{(A.21)} \\ &= x^\top h_{11} x + 2x^\top h_{12} u + u^\top h_{22} u && \text{(A.22)} \end{aligned}$$

and then

$$\frac{\partial}{\partial u} J = 2h_{22} u + 2h_{12}^\top x \tag{A.23}$$

$$\frac{\partial^2}{\partial u^2} J = 2h_{22} \tag{A.24}$$

If h_{22} is positive definite then J has a minimum for:

$$u^* = -h_{22}^{-1} h_{12}^\top x \tag{A.25}$$

which introduced in the expression for the performance index give that:

$$\begin{aligned} J^* &= x^\top h_{11} x + 2x^\top h_{12} u^* + (u^*)^\top h_{22} u^* \\ &= x^\top h_{11} x - 2(x^\top h_{12} h_{22}^{-1}) h_{12}^\top x \\ &\quad + (x^\top h_{12} h_{22}^{-1}) h_{22} (h_{22}^{-1} h_{12}^\top x) \\ &= x^\top h_{11} x - x^\top h_{12} h_{22}^{-1} h_{12}^\top x \\ &= x^\top (h_{11} - h_{12} h_{22}^{-1} h_{12}^\top) x \end{aligned}$$

□

A.2 Matrix Calculus

Let x be a vector

$$x = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \tag{A.26}$$

and let s be a scalar. The derivative of x w.r.t. s is defined as:

$$\frac{dx}{ds} = \begin{bmatrix} \frac{dx_1}{ds} \\ \frac{dx_2}{ds} \\ \vdots \\ \frac{dx_n}{ds} \end{bmatrix} \tag{A.27}$$

If s is a function of x, then the derivative of s w.r.t. x is:

$$\frac{ds}{dx} = \left[\frac{ds}{dx_1}, \ \frac{ds}{dx_2}, \ \cdots \ \frac{ds}{dx_n} \right] \tag{A.28}$$

If x depend on a scalar variable, t, then the derivative of s with respect to t is given by:

$$\frac{ds}{dt} = \frac{\partial s}{\partial x}\frac{dx}{dt} \tag{A.29}$$

The second derivative or the **Hessian** matrix for s with respect to x is denoted as:

$$H = \frac{d^2 s}{dx^2} = \left[\frac{d^2 s}{dx_r dx_s}\right] \tag{A.30}$$

which is a symmetric $n \times n$ matrix. It is possible to use a Taylor expantion of s from x_0, i.e.

$$s(x) = s(x_0) + \left(\frac{ds}{dx}\right)^\top (x - x_0) + \frac{1}{2}((x - x_0)^\top \left[\frac{d^2 s}{dx^2}\right](x - x_0) + \cdots \tag{A.31}$$

Let $f: \;\; \mathbb{R}^n \to \mathbb{R}^M$ be a vector function, i.e.:

$$f = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_m \end{bmatrix} \tag{A.32}$$

The **Jacobian** matrix of f with respect to x is a $m \times n$ matrix:

$$\frac{df}{dx} = \begin{bmatrix} \frac{df_1}{dx_1} & \frac{df_1}{dx_2} & \cdots & \frac{df_1}{dx_n} \\ \frac{df_2}{dx_1} & \frac{df_2}{dx_2} & \cdots & \frac{df_2}{dx_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{df_m}{dx_1} & \frac{df_m}{dx_2} & \cdots & \frac{df_m}{dx_n} \end{bmatrix} = \left[\frac{df}{dx_1}\frac{df}{dx_2}\cdots\frac{df}{dx_n}\right] = \begin{bmatrix} \frac{df_1}{dx} \\ \frac{df_2}{dx} \\ \vdots \\ \frac{df_m}{dx} \end{bmatrix} \tag{A.33}$$

The derivative of f with respect to t is a $m \times 1$ vector

$$\frac{df}{dt} = \frac{\partial f}{\partial x}\frac{dx}{dt} \tag{A.34}$$

Let y be a vector, A, B, D and Q matrices of apropriate dimensions (such the given expression is well defined), then:

$$\frac{d}{dt}\left(A^{-1}\right) = -A^{-1}\left(\frac{d}{dt}A\right)A^{-1} \tag{A.35}$$

and

$$\frac{\partial}{\partial x}(y^\top x) = \frac{\partial}{\partial x}(x^\top y) = y \tag{A.36}$$

$$\frac{\partial}{\partial x}(y^\top A x) = \frac{\partial}{\partial x}(x^\top A y) = Ay \tag{A.37}$$

$$\frac{\partial}{\partial x}(y^\top f(x)) = \frac{\partial}{\partial x}(f^\top(x) y) = \nabla_x f^\top y \tag{A.38}$$

$$\frac{\partial}{\partial x}(x^\top A x) = Ax + A^\top x \tag{A.39}$$

If Q is symmetric then:

$$\frac{\partial}{\partial x}(x^\top Q x) = 2Qx \tag{A.40}$$

A very important Hessian matrix is:

$$\frac{\partial^2}{\partial x^2}(x^\top A x) = A + A^\top \tag{A.41}$$

and if Q is symmetric:

$$\frac{\partial^2}{\partial x^2}(x^\top Q x) = 2Q \tag{A.42}$$

We also have the Jacobian matrix

$$\frac{\partial}{\partial x}(Ax) = A \tag{A.43}$$

and furthermore:

$$\begin{aligned}
\frac{\partial}{\partial A} tr\{A\} &= I && \text{(A.44)}\\
\frac{\partial}{\partial A} tr\{BAD\} &= B^\top D^\top && \text{(A.45)}\\
\frac{\partial}{\partial A} tr\{ABA^\top\} &= 2AB && \text{(A.46)}\\
\frac{\partial}{\partial A} det\{BAD\} &= det\{BAD\}A^{-\top} && \text{(A.47)}\\
\frac{\partial}{\partial A} log(detA) &= A^{-\top} && \text{(A.48)}\\
\frac{\partial}{\partial A}(tr(WA^{-1})) &= -(A^{-1}WA^{-1})^\top && \text{(A.49)}
\end{aligned}$$

A.3 Matrix Algebra

Consider matrices, P, Q, R and S of appropriate dimensions (such the products exists). Assume that the inverse of P, R and $(SPQ + R)$ exists. Then the follow indentity is valid.

$$(P^{-1} + QR^{-1}S)^{-1} = P - PQ(SPQ + R)^{-1}SP \tag{A.50}$$

This identity is known as the *inversion lemma*.

Bibliography

Bertsekas, D. P. (1995), *Dynamic Programming and Optimal Control*, Athena.

Bryson, A. E. (1999), *Dynamic Optimization*, Addison-Wesley.

Bryson & Ho (1975), *Applied Optimal Control*, Hemisphere.

Lewis, F. (1986*a*), *Optimal Control*, Wiley.

Lewis, F. (1986*b*), *Optimal Estimation*, Wiley.

Lewis, F. L. (1992), *Applied Optimal Control and Estimation*, Prentice Hall. ISBN 0-13-040361-X.

Ravn, H. (1994), *Statisk og Dynamisk Optimering*, Institute of Mathematical Statistics and Operations Research (IMSOR), The Technical University of Denmark.

Vidal, R. V. V. (1981), *Notes on Static and Dynamic Optimization*, Institute of Mathematical Statistics and Operations Research (IMSOR), The Technical University of Denmark.

Weber, R. (n.d.), 'Optimization and control'.

Index