Static and Dynamic Optimization (42111)

Build. 303b, room 048
Section for Dynamical Systems
Dept. of Applied Mathematics and Computer Science
The Technical University of Denmark

Email: nkpo@dtu.dk
phone: +45 4525 3356
mobile: +45 2890 3797

2019-11-03 13:23

Lecture 9: End Point constraints

Outline of lecture

- Recap F8
- Solution to Free C problem
- Simple EPC
- Simple partial EPC
- Linear EPC
- General EPC
- Continuous time DO with EPC
- Reading guidance (DO chapter 3).

Dynamic Optimization (D, free)

Find a sequence u_i, $i = 0, \ldots, N-1$ which takes the system

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

from its initial state $\underline{x}_0$ along a trajectory such that the performance index

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

is optimized. Define the Hamiltonian function as:

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

Then the Euler-Lagrange equations are:

$$x_{i+1} = f_i(x_i, u_i) \qquad \lambda_i^T = \frac{\partial}{\partial x_i} H_i$$

$$0 = \frac{\partial}{\partial u_i} H_i$$

with boundary conditions:

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

Dynamic Optimization (C, free)

Find a function u_t $t \in [0;\ T]$ which takes the system system

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

from its initial state $\underline{x}_0$ along a trajectory such that the performance index

$$J = \phi_T[x_T] + \int_0^T L_t(x_t, u_t)\, dt$$

is optimized. Define the Hamilton function as:

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

Then the Euler-Lagrange equations are:

$$\dot{x}_t = f_t(x_t, u_t) \qquad -\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t$$

$$0 = \frac{\partial}{\partial u_t} H_t$$

with boundary conditions:

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

Solutions for the C problem

Solutions for the C problem

Type of solutions:

- Analytical solutions (for very simple problems)
- Semi analytical solutions (eg. the LQ problem)
- numerical solutions

Forward sweep method

$$H_t = L_t(x,u) + \lambda_t f_t(x,u)$$

Euler-Lagrange Equations I

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

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad \lambda_T^T = \frac{\partial}{\partial x_T} \phi_T(x_T)$$

$$0 = \frac{\partial}{\partial u_t} H_t$$

Euler-Lagrange Equations II

$$\dot{x}_t = f_t(x_t, u_t)$$

$$-\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} L_t(x,u) + \lambda^T \frac{\partial}{\partial x_t} f_t(x,u)$$

$$0 = \frac{\partial}{\partial u_t} L_t(x_t,u_t) + \lambda^T \frac{\partial}{\partial u_t} f_t(x_t,u_t)$$

Costate equation

$$\dot{\lambda}_t = -[\frac{\partial}{\partial x_t} H_t]^T = g_t(x_t, \lambda_t, u_t)$$

Stationarity equation

$$u_t = h_t(x_t, \lambda_t)$$

Forward sweep method

Guess λ_0 and use the knowledge x_0 and integrate (use e.g. ode45)

$$\frac{d}{dt}\left[\begin{array}{c} x_t \\ \lambda_t \end{array}\right] = \left[\begin{array}{c} f_t(x_t, u_t) \\ g_t(x_t, \lambda_t, u_t) \end{array}\right] \qquad u_t = h_t(x_t, \lambda_t)$$

i.e.

$$\frac{d}{dt}\left[\begin{array}{c} x_t \\ \lambda_t \end{array}\right] = \left[\begin{array}{c} \underline{f}_t(x_t, \lambda_t) \\ \underline{g}_t(x_t, \lambda_t) \end{array}\right]$$

At the end check the condition:

$$\lambda_T^T = \frac{\partial}{\partial x_T}\phi_T(x_T)$$

Use e.g. fsolve to ajust λ_0 such that the condition is satisfied.

End point constraints (EPC)

Find a sequence u_i, $i = 0, \ldots, N-1$ which takes the system

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

from its initial state, $\underline{x}_0$, along a trajectory to

$$x_N = \underline{x}_N \qquad\qquad \textbf{(Simple EPC)}$$

such that the performance index

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

is optimized.

End point constraints (D)

In general:

$$\psi_N(x_N) = 0 \qquad \psi : \mathbb{R}^{n+1} \to \mathbb{R}^p \qquad p \leq n+1$$

Linear EPC

$$Cx_N = \underline{r} \qquad \text{e.g.} \qquad C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \qquad r = \begin{bmatrix} 1.4 \\ 2.3 \end{bmatrix}$$

Simple partial EPC

$$x_N = \begin{bmatrix} \tilde{x}_N \\ \bar{x}_N \end{bmatrix} \qquad \tilde{x}_N = \underline{\tilde{x}}_N \in \mathbb{R}^p \qquad p \leq n$$

Investment planning

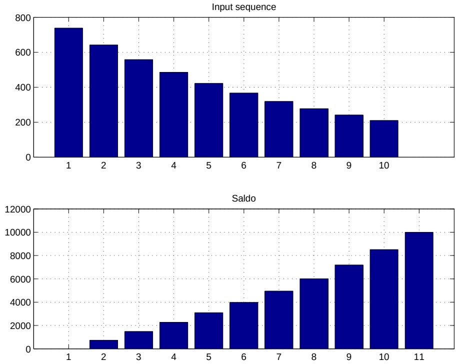
Plan: During a period of time (N intervals) to invest a amount of money u_i to obtain a specified sum ($\underline{x}_N$) at the end of the period.

Dynamics:

$$x_{i+1} = (1+\alpha)x_i + u_i \qquad x_0 = 0 \qquad x_N = 10.000 \ Dkr$$

Objective:

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



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

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

the performance index

$$J = \phi_N(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{2}$$

and the simple terminal constraint

$$x_N = \underline{x}_N \tag{3}$$

where $\underline{x}_N$ (and $\underline{x}_0$) is given. Introduce the **multiplier** (vector with same length as x since EPC are simple) ν and form the **Lagrange** relaxation:

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

New conditions: Stationarity w.r.t. x_N (for $i = N-1$) gives:

$$0^T = \frac{\partial}{\partial x_N}\phi_N + \nu^T - \lambda_N^T \qquad\qquad \lambda_N^T = \nu^T + \frac{\partial}{\partial x_N}\phi_N$$

Stationarity w.r.t. ν gives

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

The rest is as usual (as for the free case).

Simple end point constraints

Defining the Hamiltonian function

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

The Euler-Lagrange equations:

$$x_{i+1} = f_i(x_i, u_i) \qquad \lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad 0^T = \frac{\partial}{\partial u_i} H_i$$

with boundary conditions:

$$x_0 = \underline{x}_0 \qquad x_N = \underline{x}_N \qquad \lambda_N^T = \nu^T + \frac{\partial}{\partial x_N} \phi_N$$

Conditions: $3 \times n$ (of which $2 \times n$ are trivial and n are very simple)
Unknowns: x_0, x_N and ν (results: $3 \times n$)

Conditions on states rather than on costates (for simple EPC). Trade conditions on states for costates.

Partial simple end point constraints

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

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{4}$$

the performance index

$$J = \phi_N(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{5}$$

and the simple but partial simple terminal constraints

$$x_N = \begin{bmatrix} \tilde{x}_N \\ \bar{x}_N \end{bmatrix} \qquad \tilde{x}_N = \underline{\tilde{x}}_N \in \mathbb{R}^p \qquad p < n \qquad \qquad \lambda_N = \begin{bmatrix} \tilde{\lambda}_N \\ \bar{\lambda}_N \end{bmatrix}$$

where $\underline{\tilde{x}}_N$ (and $\underline{x}_0$) are given. Introduce the **multiplier** (vector) $\nu \in \mathbb{R}^p$ and form the **Lagrange** relaxation:

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

New conditions: Stationarity w.r.t. x_N (i.e. $\tilde{x}$ and $\bar{x}$) gives:

$$\tilde{\lambda}_N^T = \nu^T + \frac{\partial}{\partial \tilde{x}} \phi \qquad \qquad \bar{\lambda}_N^T = \frac{\partial}{\partial \bar{x}} \phi$$

Stationarity w.r.t. ν gives

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

The rest is as usual (free dyn. opt.).

Partial simple end point constraints (in summary)

Defining the **Hamiltonian** function

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

The Euler-Lagrange equations:

$$x_{i+1} = f_i(x_i, u_i) \qquad \lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad 0^T = \frac{\partial}{\partial u_i} H_i$$

with boundary conditions:

$$x_0 = \underline{x}_0 \qquad \tilde{x}_N = \underline{\tilde{x}}_N \qquad\qquad \tilde{\lambda}_N^T = \nu^T + \frac{\partial}{\partial \tilde{x}_N} \phi(x_N) \qquad \bar{\lambda}_N^T = \frac{\partial}{\partial \bar{x}_N} \phi(x_N)$$

Conditions: $n + p + p + (n - p) = 2 \times n + p$.
Unknowns: x_0, $\tilde{x}_N$, ν and $\bar{\lambda}_N$ (results: $n + p + p + (n - p)$)

General EP

DTU

General end point constraints

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

$$x_{i+1} = f_i(x_i, u_i) \qquad x_0 = \underline{x}_0 \tag{6}$$

the performance index

$$J = \phi_N(x_N) + \sum_{i=0}^{N-1} L_i(x_i, u_i) \tag{7}$$

and the general terminal constraints

$$\psi_N(x_N) = 0 \qquad\qquad \psi : \mathbb{R}^{n+1} \to \mathbb{R}^p \tag{8}$$

where ψ (and $\underline{x}_0$) are given. Introduce the **multiplier** (vector of length p) ν and form the **Lagrange** relaxation:

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

New conditions: Stationarity w.r.t. x_N gives:

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

Stationarity w.r.t. ν gives

$$\psi_N(x_N) = 0$$

The rest is as usual (free dyn. opt.).

DTU

General end point constraints (D)

Defining the **Hamiltonian** function

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

The Euler-Lagrange equations:

$$x_{i+1} = f_i(x_i, u_i) \qquad \lambda_i^T = \frac{\partial}{\partial x_i} H_i \qquad 0^T = \frac{\partial}{\partial u_i} H_i$$

with boundary conditions:

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

Conditions: $n + p + n$.
Unknowns: x_0, x_N and ν (results: $2 \times n + p$)

DTU

End point constraints (C)

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

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

such that the performance index

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

is optimized and the end point constraints in

$$\psi_T(x_T) = 0$$

are met.

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

Stationarity w.r.t. x_T gives:

$$\lambda_T^T = \nu^T \frac{\partial}{\partial x_T} \psi_T + \frac{\partial}{\partial x_T} \phi_T$$

stationarity w.r.t. ν gives

$$\psi_T(x_T) = 0$$

DTU

Euler-Lagrange equations

If we introduce the Hamiltonian function as

$$H_t(x_t, u_t) = L_t(x_t, u_t) + \lambda_t^T f_t(x_t, u_t) \tag{9}$$

we can express the necessary conditions as

$$\dot{x}_t = f_t(x_t, u_t) \qquad -\dot{\lambda}_t^T = \frac{\partial}{\partial x_t} H_t \qquad 0^T = \frac{\partial}{\partial u_t} H_t$$

with the (split) boundary conditions

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

Simple EPC:

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

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

Partial simple EPC:

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

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

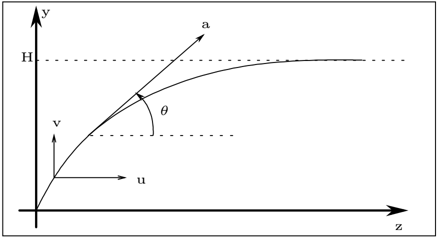
Linear EPC

$$Cx_T = \underline{r}$$

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

Orbit injection problem - Simplified

A body is initially at rest in the origin. A constant specific thrust force, a, is applied to the body in a direction that makes an angle θ_t with the z-axis. Let u and v be the velocity in the z and y direction, respectively.



The task is to find an input function of angles of direction, θ_t such that the body in a finite period, T,

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

$$y_T = H$$

2 has zero vertical speed (y-direction)

$$v_T = 0$$

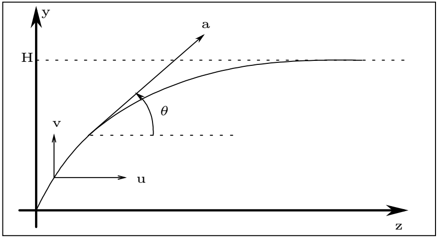
3 has maximum horizontal speed (z-direction)

$$Max\ u_T$$

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

Orbit injection - The dynamic

The problem is to find the input function, θ_t, such that the terminal horizontal velocity, u_T, (at a specific altitude H) is maximized.



The dynamic is:

$$\frac{d}{dt}\begin{bmatrix} u_t \\ v_t \\ z_t \\ y_t \end{bmatrix} = \begin{bmatrix} a\ cos(\theta_t) \\ a\ sin(\theta_t) \\ u_t \\ v_t \end{bmatrix} \qquad \begin{bmatrix} u_0 \\ v_0 \\ z_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Orbit injection - The terminal conditions

The terminal constraints are

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

The objective is to maximize:

$$J = \phi(x_T) = u_T$$

More condensed:

$$J = \phi(x_T) = u_T \qquad\qquad \begin{bmatrix} v \\ y \end{bmatrix}_T = \begin{bmatrix} 0 \\ H \end{bmatrix}$$

$$x_t = \begin{bmatrix} u \\ v \\ z \\ y \end{bmatrix}_t$$

Orbit injection - Euler-Lagrange equations

The Hamilton functions is (since $L = 0$)

$$H_t = \lambda_t^T f_t = \begin{bmatrix} \lambda_t^u & \lambda_t^v & \lambda_t^z & \lambda_t^y \end{bmatrix} \begin{bmatrix} a\ cos(\theta_t) \\ a\ sin(\theta_t) \\ u_t \\ v_t \end{bmatrix}$$

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

The Euler-Lagrange equations consists of the **state** equation,

$$\frac{d}{dt} \begin{bmatrix} u_t \\ v_t \\ z_t \\ y_t \end{bmatrix} = \begin{bmatrix} a\ cos(\theta_t) \\ a\ sin(\theta_t) \\ u_t \\ v_t \end{bmatrix} \qquad \begin{bmatrix} u_0 \\ v_0 \\ z_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

(just cut and paste)

the **costate** equation

$$-\frac{d}{dt} \begin{bmatrix} \lambda_t^u & \lambda_t^v & \lambda_t^z & \lambda_t^y \end{bmatrix} = \begin{bmatrix} \lambda_t^z & \lambda_t^y & 0 & 0 \end{bmatrix} \qquad = \frac{\partial}{\partial x_t} H_t$$

and the **stationarity** condition

$$0 = -\lambda_t^u a\ sin(\theta_t) + \lambda_t^v a\ cos(\theta_t) \qquad = \frac{\partial}{\partial u_t} H_t$$

DTU

Orbit injection - The boundary conditions

Since

$$\phi_T(x_t) = u_t \qquad \begin{bmatrix} v \\ y \end{bmatrix}_T = \begin{bmatrix} 0 \\ H \end{bmatrix}$$

we have the boundary conditions

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

$$\lambda_T^u = 1 \qquad \lambda_T^z = 0$$

Orbit injection - The stationarity

The **stationarity** condition

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

gives the tangent law:

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

It turns out (later on) to be a linear tangent law.

Orbit injection - The Costates

The Costate equations

$$-\frac{d}{dt}\left[\begin{array}{cccc} \lambda_t^u & \lambda_t^v & \lambda_t^z & \lambda_t^y \end{array}\right] = \left[\begin{array}{cccc} \lambda_t^z & \lambda_t^y & 0 & 0 \end{array}\right]$$

and the boundary conditions

$$\lambda_T^v = \nu_v \qquad \lambda_T^y = \nu_y \qquad \text{(just a copy)}$$

$$\lambda_T^u = 1 \qquad \lambda_T^z = 0$$

gives us:

$$\lambda_t^z = 0 \qquad \lambda_t^y = \nu_y \qquad \textit{constant in time}$$

$$\lambda_t^u = 1 \qquad \textit{constant in time}$$

$$\lambda_t^v = \nu_v + \nu_y(T - t)$$

$$tan(\theta_t) = \nu_v + \nu_y(T - t)$$

DTU

Find ν_v and ν_y such that

$$tan(\theta_t) = \nu_v + \nu_y(T - t)$$

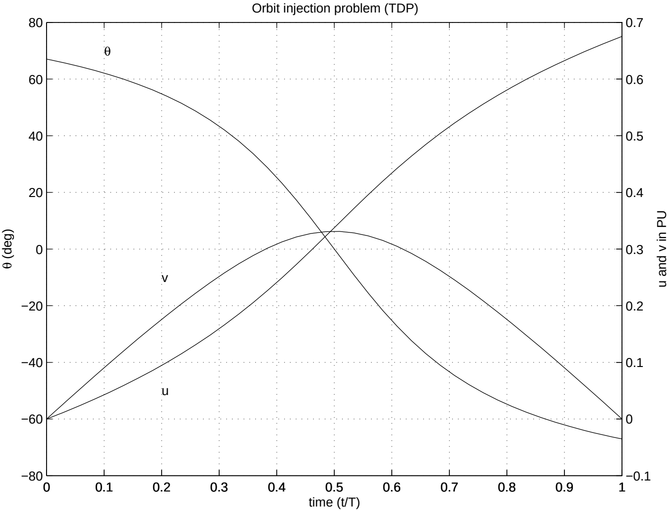
in the dynamics

$$\frac{d}{dt}\begin{bmatrix} u_t \\ v_t \\ z_t \\ y_t \end{bmatrix} = \begin{bmatrix} a\ cos(\theta_t) \\ a\ sin(\theta_t) \\ u_t \\ v_t \end{bmatrix} \qquad \begin{bmatrix} u_0 \\ v_0 \\ z_0 \\ y_0 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

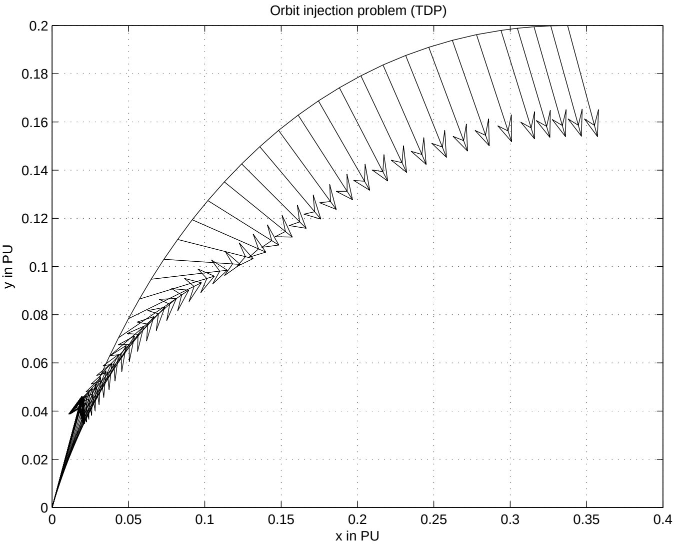
results in

$$\begin{bmatrix} v \\ y \end{bmatrix}_T = \begin{bmatrix} 0 \\ H \end{bmatrix}$$

Orbit injection



Orbit injection



```
% ------------------------------------------------
parms.m
% ------------------------------------------------

T=1; % parameters
a=1;
H=0.2;
x0=zeros(4,1); % Initial state variable

% ------------------------------------------------
function main1
% ------------------------------------------------

parms
parm0=[-2.4 4.7]'; % Initial guess on parametes

opt=optimset; % Options for fsolve
opt=optimset(opt,'Display','iter');
parm=fsolve(@erf,parm0,opt); % Call fsolve for finding parameters

[err,time,xt]=erf(parm); % Call erf ones more for getting the
tht=atan(parm(1)+parm(2)*(T-time)); % optimal input solution

% Here goes the plotting commands. Omitted here.
% file on databar: ~nkpo/02711/dist3/main1.m
```

```
% ----------------------------------------------------
function [err,time,xt]=erf(parm)
% ----------------------------------------------------
% Determine the end point error (err) given the EPC Lagrange multipliers
% in parm (and the constants that specifies the problem).
parms
Tspan=0:T;
[time,xt]=ode45(@tdp,Tspan,x0);
xT=xt(end,:)';
err=[xT(2);
     xT(4)-H];


% ----------------------------------------------------
function dx=tdp(t,x,parm)
% ----------------------------------------------------
% System model. Determine the (time) derivative of the state vector
% given the time, state (x) and the EPC Lagrange multipliers.
parms
u=x(1); v=x(2); z=x(3); y=x(4);
nuu=parm(1); nuy=parm(2);
th=atan(nuu+nuy*(T-t));
dx=[a*cos(th);
    a*sin(th);
    u;
    v];
```

Reading guidance

DO Chapter 3