KTH Matematik

Homework 1: 5B1873: Optimal Control Spring 2007

Grading: You may use $\min(1, \frac{\text{your credit}}{30})$ extra points on the exam.

The homework is due on Feb 5, at 17.00

1. A corporation has \$10 million to allocate to its five plants for possible expansion. Each plant has submitted a number of proposals on how it intends to spend the money. Each proposal gives the cost of the expansion and the total revenue expected. The following table gives the proposals generated:

	Plant 0		Plant 1		Plant 2		Plant 3		Plant 4	
	cost	revenue	cost	revenue	cost	revenue	cost	revenue	cost	revenue
Proposal 1	0	0	0	0	0	0	0	0	0	0
Proposal 2	2	2.7	1	1.27	2	2.45	1	1.25	2	2.5
Proposal 3	3	3.75	2	2.6	3	3.8	2	2.8	4	5.2
Proposal 4	5	6.5	4	5.6	5	6.75	4	5.4	5	6.3
Proposal 5	—	—	—	—	6	7.6	5	6.6	—	—

The units for cost and revenue are both “million dollar”. Each plant will only be permitted to enact one of its proposals, and the money allocated to each plant will be exactly the amount as given in the proposals. The goal here is to maximize the firm’s expected revenues resulting from the allocation of the \$10 million. Any of the \$10 million unspent can be invested in shares and the expected revenues are given in the following table (again, the units are “million dollar”):

investment	0	1	2	3	4	5	6	7	8	9	10
revenue	0	1.2	2.2	3.5	4.8	5.3	7.7	9.2	9.6	10.2	12.7

This decision problem can be solved stage by stage. At the 0^{th} stage, money is to be allocated to Plant 0. At the 1^{st} stage, money is to be allocated to Plant 1, and so on. At the last stage, unspent money is invested.

Let $u[n]$ be the amount of money allocated to Plant n, $r_n(u[n])$ denote the expected revenue as a result of spending $u[n]$ million dollars, and $x[n]$ be the sum of money allocated out up to stage $n-1$; that is,

$$x[n] = \sum_{i=0}^{n-1} u[i].$$

Finally, let $I(y)$ be the expected revenue as the result of investing y million dollars in shares. Using these notations, the allocation problem can be formulated as an optimization problem

$$\max_{u[n],n=0,\cdots,4} J(x[0],u) := I(10-x[5]) + \sum_{i=0}^{4} r_i(u[i]), \quad \text{subject to}$$
$$x[n+1] = x[n]+u[n], \quad x[n] \in X(n), \quad u[n] \in U(n,x[n]), \quad n=0,\cdots,4. \tag{1}$$

where $X(n)$ and $U(n,x[n])$ denote the sets of admissible values for $x[n]$ and $u[n]$, respectively.

The optimization problem (1) may be solved by using the backward dynamic programming. To this end, let $J_k^*(x[k])$ be the optimal cost-to-go function (of $x[k]$) of stage k. For $k=0,1,\cdots,4$,

$$J_k^*(x[k]) := \max_{u[n],n=k,\cdots,4} I(10-x[5]) + \sum_{i=k}^{4} r_i(u[i]), \quad \text{subject to}$$
$$x[n+1] = x[n]+u[n], \quad x[n] \in X(n), \quad u[n] \in U(n,x[n]), \quad n=k,\cdots,4. \tag{2}$$

Furthermore, let $u_k^*(x[k])$ denote the optimal allocation strategy (as a function of $x[k]$) of stage k. In other words, $u_k^*(x[k])$ is the argument of maximum of (2).

(a) Given that $x[4]=2$ and $u[4]=5$, what is $J_5^*(x[5])$? .... (2p)

(b) The optimal cost-to-go functions and the optimal allocation strategies of stages 1 to 4 are (partially) given in Table 1. Find the missing entries $(J_1^*(2), u_1^*(2))$, and $(J_3^*(6), u_3^*(6))$. .... (4p)

(c) Find the optimal allocation scheme and the maximal expected revenue. .... (4p)

2. A decision maker must choose between two activities over a time interval $[0,t_f]$. Each activity earns a reward at rate $g_k(t)$, $k=1,2$. Every switch between the two activities costs $c>0$. As an example, the reward for starting with activity 1, switch to activity 2 at time t_1 and back to 1 at time $t_2>t_1$ earns the total reward

$$\int_0^{t_1} g_1(t)dt + \int_{t_1}^{t_2} g_2(t)dt + \int_{t_2}^{t_f} g_1(t)dt - 2c$$

We want to find a switching sequence that maximize the total reward. Switching can only occur inside the the interval $(0,t_f)$.

Assume the function $g_1(t)-g_2(t)$ changes sign a finite number of times in the interval $(0,t_f)$.

$(a.)$ Formulate the problem as a sequential optimization problem and then formulate the corresponding DP recursion. .... (6p)

$x[4]$	0	1	2	3	4	5	6	7	8	9	10
$J_4^*(x[4])$	12.90	11.70	10.20	9.20	7.70	6.40	5.20	3,7	2.50	1.20	0
$u_4^*(x[4])$	4	2	2	0	0	4	4	2	2	0	0

$x[3]$	0	1	2	3	4	5	6	7	8	9	10
$J_3^*(x[3])$	13.10	12.00	10.60	9.20	8.00	6.60	$J_3^*(6)$	4.00	2.80	1.25	0
$u_3^*(x[3])$	4	2	4	0	2	4	$u_3^*(6)$	2	2	1	0

$x[2]$	0	1	2	3	4	5	6	7	8	9	10
$J_2^*(x[2])$	13.35	12.15	10.75	9.55	8.00	6.75	5.4	4.00	2.80	1.25	0
$u_2^*(x[2])$	5	5	5	5	0	5	0	0	0	0	0

$x[1]$	0	1	2	3	4	5	6	7	8	9	10
$J_1^*(x[1])$	13.60	12.35	$J_1^*(2)$	9.60	8.40	6.85	5.60	4.07	2.80	1.27	0
$u_1^*(x[1])$	4	4	$u_1^*(2)$	4	4	4	4	1	0	1	0

Tabell 1: Optimal cost-to-go functions $J_k^*(x[k])$ and optimal allocation strategies $u_k^*(x[k])$.

(b.) Solve the dynamic programming problem in (a) for the case when $c = 2$, $t_f = 3$ and

$$g_1(t) = \begin{cases} 4, & 0 \le t \le 1 \\ 0, & 1 \le t \le 2 \\ 5, & 2 \le t \le 3 \end{cases}, \qquad g_2(t) = \begin{cases} 1, & 0 \le t \le 1 \\ 6, & 1 \le t \le 2 \\ 2, & 2 \le t \le 3 \end{cases}$$

.... (4p)

3. The purpose of this problem is to balance a ball on a beam using Model Predictive Control (MPC). The ball should be brought to the middle point of the beam (the origin) by steering the angle of the beam using an electric motor. The distance between the ball position on the beam and the origin is denoted y and it satisfies the following differential equation.

$$\ddot{y} = b\sin(\theta) \approx b\theta$$

where b is a constant that depends on various system parameters. We assume the normalized value $b = 1$. If you want to know more about the ball and beam process and its model, please see [1, 2] on the Internet.

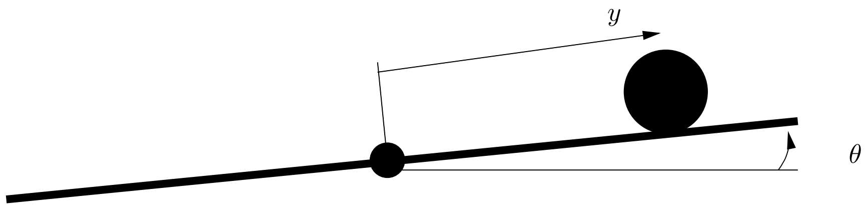
In order to use discrete time MPC we will need a discrete time approximation of the dynamics. We use the following state space realization

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{\Phi}\mathbf{z}_k + \mathbf{\Gamma}u_k \\ y_k &= \mathbf{C}\mathbf{z}_k \end{aligned} \tag{3}$$

where $u_k = \theta(kT)$, T is the sampling time, and

$$\mathbf{z}_k = \begin{bmatrix} y(kT) \\ \dot{y}(kT) \end{bmatrix}, \quad \mathbf{\Phi} = \begin{bmatrix} 1 & T \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Gamma} = \begin{bmatrix} T^2/2 \\ T \end{bmatrix}$$

$$\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$$



Figur 1: The ball is rolling along a track along the beam and can therefore only fall off at either of the two end points.

It is easy to see that this is a reasonable representation of the continuous time dynamics in (3) since it is equivalent to the following difference equation

$$\frac{1}{T^2}(y(k+2)T) - 2y((k+1)T) + y(kT)) = \frac{1}{2}(u((k+1)T) + u(kT))$$

The left hand side approximates the second derivative $\ddot{y}(t)$ and the right hand side approximates $u(t)$ on the interval $[t, t+T]$.

The purpose of the optimization problem is to steer the ball to the origin without spending too much energy. A natural optimization criterion is the following linear quadratic problem

$$\begin{array}{ll} \min & q_f|\mathbf{z}_{t+N|t}|^2 + \sum_{k=0}^{N-1}(q|\mathbf{C}\mathbf{z}_{t+k|t}|^2 + ru_{t+k|t}^2) \\ \text{s.t.} & \mathbf{z}_{t+k+1|t} = \mathbf{\Phi}\mathbf{z}_{t+k|t} + \mathbf{\Gamma}u_{t+k|t} \quad k = 0, 1, \ldots, N-1 \end{array} \tag{4}$$

where q_f, q, and r are positive parameters.

In the following problems we ask you to implement the above quadratic optimization problem for MPC of the ball and beam process. You should also experiment with the algorithm. The appendix contains a Matlab skeleton that defines a suggested structure for your code. There are also some hints on how to implement the system matrices.

(a) Rewrite the optimization problem (4) on the form

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ \text{s.t.} & \mathbf{A}\mathbf{x} = \mathbf{b} \end{array}$$

where $\mathbf{x} = \begin{pmatrix}\mathbf{z}_{t+1|t}^T & \cdots & \mathbf{z}_{t+N|t}^T & u_{t|t} & \ldots u_{t+N-1|t}\end{pmatrix}^T$ (note that $\mathbf{z}_{t|t}$ is given and is not a variable in the optimization problem). Note also that the right hand side $\mathbf{b}$ depends on the last measured state $\mathbf{z}_{t|t}$ and must be updated in every iteration of the MPC algorithm (see code in the appendix). Solve the problem using Matlab with the following parameter values $\mathbf{z}_{0|0} = \begin{pmatrix}0.5 & 1\end{pmatrix}^T$, $r = 1$, the sampling time $T = 0.1$ and

(*i*) $N = 5$, $q_f = 5$, $q = 2$,

(*ii*) $N = 10$, $q_f = 5$, $q = 2$,

(*iii*) $N = 10$, $q_f = 10$, $q = 5$.

What conclusions can you make regarding the convergence to the origin. . . . . (7p)

(b) The beam has in reality finite length and it is necessary to introduce constraints in the optimization problem to ensure that the ball does not fall off the beam. There are normally limitations on the magnitude of the control signal. The resulting optimal control problem can be formulated as

$$\begin{array}{ll} \min & q_f|\mathbf{z}_{t+N|t}|^2 + \sum_{k=0}^{N-1}(q|\mathbf{C}\mathbf{z}_{t+k|t}|^2 + ru_{t+k|t}^2) \\ \text{s.t} & \begin{cases} \mathbf{z}_{t+k+1|t} = \mathbf{\Phi}\mathbf{z}_{t+k|t} + \mathbf{\Gamma}u_{t+k|t} & k = 0, \dots, N-1 \\ -1 \le y_{t+k|t} \le 1, \quad k = 1, \dots, N \\ -1 \le u_{t+k|t} \le 1, \quad k = 0, \dots, N-1 \end{cases} \end{array} \tag{5}$$

and your task is to rewrite it on the form

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ \mathbf{A}\mathbf{x} \le \mathbf{b} \end{array} \end{array}$$

and apply MPC with the following parameter values $\mathbf{z}_{0|0} = \begin{pmatrix} 0.5 & 1 \end{pmatrix}^T$, $r = 1$,

(*i*) $N = 5$, $q_f = 5$, $q = 2$

(*ii*) $N = 10$, $q_f = 5$, $q = 2$,

(*iii*) $N = 10$, $q_f = 10$, $q = 5$.

What are your conclusions?. . . . . (6p)

(c) The predicted states can be written as a function of $u_{t+k|t}, \dots, u_{t|t}$. Hence, one may define an optimization problem in the reduced variable vector $\mathbf{x} = \begin{pmatrix} u_{t|t} & \dots & u_{t+N-1|t} \end{pmatrix}^T$. Formulate the optimization problem on the form

$$\begin{array}{ll} \min & \frac{1}{2}\mathbf{x}^T\mathbf{H}\mathbf{x} + \mathbf{x}^T\mathbf{f} + \mathbf{g} \\ \text{s.t.} & \begin{array}{l} \mathbf{A}_{eq}\mathbf{x} = \mathbf{b}_{eq} \\ \mathbf{A}\mathbf{x} \le \mathbf{b} \end{array} \end{array}$$

You don't need to implement this in Matlab. . . . . (2p)

Referenser

[1] LTH Department of Automatic Control. The ball and beam lab process. http://www.control.lth.se/education/laboratory/bommen.html.

[2] P. Wellstead. Ball and beam 1: Basics. www.control-systems-principles.co.uk/whitepapers/ball-and-beam1.pdf.

Good luck!

1 Matlabkod

```
%
%------- Basic system model
%
clear;
T=0.1;
Phi=[1 T;0 1];
Gam=[T^2/2;T];
C=[1 0];
n=size(Phi,1);
m=size(Gam,2);
%
%------- Parameters --------
%
q=5;
qf=10;
r=1;
N=10;
z0=[0.5;1];

%
%------- Define matrices for the QP --------
%

For you to do!

%
%----- For problem 2 with inequalities ---------
%
A=[]; %För problem 1 saknas olikhetsbivilkor och då använder ni tomma matriser
b=[];

%
%----- Cost ------------
%

For you to do!

%
%------- MPC algorithm ------
%

M=100;   %time horizon

zt=z0;
yvec=[];
uvec=[];
```

```
options=optimset('largescale','off');
for flcnt1=1:M
    beq=AA*zt; % Matrisen AA definierar hur det senast mätta tillståndet zt
               % bestämmer högerledet i likhetsbivillkoret.
    x=quadprog(H,f,A,b,Aeq,beq,[],[],[],options);
    ut=x(n*N+1);
    zt=Phi*zt+Gam*ut;
    yvec=[yvec;C*zt];
    uvec=[uvec;ut];
end
tvec=T*(1:1:M);
subplot(3,1,3)  %For the other two sets of parameters you should change
                %the third index to 2 and 3, respectively.
plot(tvec,yvec,'-',tvec,uvec,'--')
grid
```

2 Hints for the implementation

Here follows some hints that simplifies the Matlab implementation of block matrices

(a) Block diagonal matrices can be created using the command `blkdiag(A,B)`.

(b) The command `kron(eye(N),[C;-C])` generates the block matrix

$$\begin{bmatrix} C & 0 & \dots & 0 \\ -C & 0 & \dots & 0 \\ 0 & C & \dots & 0 \\ 0 & -C & \dots & 0 \\ & & \ddots & \\ 0 & 0 & \dots & C \\ 0 & 0 & \dots & -C \end{bmatrix}$$

Generally, we have

$$\mathrm{kron}(A,B) = \begin{bmatrix} a_{11}B & a_{12}B & \dots & a_{1n}B \\ \vdots & & & \\ a_{n1}B & a_{n2}B & \dots & a_{nn}B \end{bmatrix}$$