Solution DO.2: Free Dynamic Optimization in Continuous time

Static and Dynamic Optimization

Notice, together with this solution comes (on the course home page) a distribution (dist1.zip) of m-files. On a unix system the distribution can be unpacked by the command: unzip -a dist1.zip.

1 Optimization

Just follow the instructions in the exercise.

2 Dynamic Optimization

We have the state equation:

$$\dot{x} = ax + bu \qquad a = \alpha \quad b = -1$$

$$x_0 = 50000$$

and the objective function (to be minimized).

$$J = \frac{1}{2} p x_T^2 + \int_0^T \frac{1}{2} q x_t^2 + \frac{1}{2} r u_t^2 \quad dt$$

where $r = p = q = \alpha^2$.

Question: 1 We have quite easy:

$$T = 10 \qquad x_0 = 50000$$

$$f = ax_t + bu_t \qquad \phi = \frac{1}{2} p x_T^2 \qquad L = \frac{1}{2} q x_t^2 + \frac{1}{2} r u_t^2$$

□

Question: 2 The Hamiltonian is

$$H = \frac{1}{2} q x_t^2 + \frac{1}{2} r u_t^2 + \lambda_t (a x_t + b u_t)$$

□

Question: 3 Following the instruction in the exercise, we determine the derivates:

$$\frac{\partial}{\partial x} H = q x_t + a \lambda_t$$

$$\frac{\partial}{\partial u} H = r u_t + \lambda_t b$$

□

Question: 4 The solution to this question is stated in the exercise. □

Question: 5 The stationarity condition (last equation) is simply:

$$u_t = -\frac{b}{r} \lambda_t$$

□

Question: 6 If we reverse the costate equation we have

$$\dot{\lambda} = -q x_t - a \lambda_t$$

□

Question: 7 The solution to this question is given in the text. □

Question: 8 The following code (dlq.m) models the ODE. Notice dz is matlab for $\dot{x}$.

```
%-----------------------------------------------------------------------
function dz=dlq(t,z,A,B,P,R,Q,n)
%-----------------------------------------------------------------------
% Determine the derivative of x and la as function of t, x and la
% A and B are system matrices
% Q, R and P are weight matrices in the objective function
% n is number of states.
%-----------------------------------------------------------------------
x=z(1:n); la=z(n+1:end);
u=-inv(R)*B'*la;
dz=[ A*x+B*u;
     -Q*x-A'*la];
```

□

Question: 9 The following code (loss.m) solves the ODE (forward in time) and determine the error (err) in the terminal conditions.

```
%-----------------------------------------------------------------------
function err=loss(la0,A,B,x0,P,R,Q,T,n)
%-----------------------------------------------------------------------
% Determine the error of the terminal condition as function of la0.
% A and B are system matrices
% Q, R and P are weight matrices in the objective function
```

```
% n is number of states.
% x0 is the initial state vector
%-----------------------------------------------------------------------
z0=[x0;la0'];
[time,zt]=ode45(@dlq,[0 T/2 T],z0,[],A,B,P,R,Q,n);
zT=zt(end,:)';  xT=zT(1:n); laT=zT(n+1:end)';
err=laT-xT'*P;             % Terminal condition
```

□

Question: 10

```
%-----------------------------------------------------------------------
% Program for solving the LQ problem
%-----------------------------------------------------------------------

alf=0.05;
b=-1;

A=alf;                          % System matrix
B=b;
n=length(A);
Q=alf^2;                        % Weight matrices in objective function
R=Q;
P=Q;

T=10;                           % Final time
x0=50000;                       % Initial state
la0=131;                        % First guess on lambda
% This is a good guess

% Search for correct initial costates
opt=optimset('fsolve');
opt=optimset(opt,'Display','off');
la0=fsolve(@loss,la0,opt,A,B,x0,P,R,Q,T,n); % Here is the key line

% Simulation with correct initial costate
xp0=[x0;la0'];
[time,xpt]=ode45(@dlq,[0 T],xp0,[],A,B,P,R,Q,n);
xt=xpt(:,1:n); lat=xpt(:,n+1:end);
ut=-inv(R)*B'*lat'; ut=ut';

%-----------------------------------------------------------------------
% The rest (until next function declaraion) is just plotting
subplot(311);
plot(time,xt); grid;
xlabel('Time');
ylabel('State');

subplot(312);
plot(time,lat); grid;
xlabel('Time');
ylabel('Costates ');
```

```
subplot(313);
plot(time,ut); grid;
xlabel('Time'); ylabel('Control input ');
%----------------------------------------------------------------------
```

□

Question: 11 Just change the values in `runex2` and study the effects. □