Control design (F4, IT4)
Computer-controlled systems (W4, STS4)

Homework assignments, fall 2006

Deadlines
Assignment 1  November 17, 13.15
Assignment 2  December 8, 13.15
Assignment 3  November 24, 13.15  concerns STS4, W4 only
1 Homework assignment 1 – Sampling

Contents:
- Sampling a continuous-time state-space system
- Sampling a continuous-time state-space system with time delay
- Aliasing

1.1 Problem 1

a) Consider the system with two inputs and one output:

\[ G(s) = \begin{bmatrix} \frac{K_1}{sT_1 + 1} & \frac{K_2}{sT_2 + 1} \end{bmatrix} \]

The system is of first order for each input. Write the system in state-space form.

*Hint:* It will be handy to use a diagonal form.

b) Sample the continuous-time state-space system obtained from a) and calculate the discrete state-space matrices:

\[ F = e^{Ah} \]
\[ G = \int_0^h e^{At} B dt \]

where \( h \) is the sampling period.

c) Show that the discrete transfer operator for the system is given by:

\[ H(q) = \begin{bmatrix} \frac{b_1}{q - a_1} & \frac{b_2}{q - a_2} \end{bmatrix} \]

and determine \( a_1, a_2, b_1, b_2 \) as functions of \( K_1, K_2, T_1 \) and \( T_2 \) and \( h \).

1.2 Problem 2

Consider a time-delayed continuous system described by

\[ \frac{dx(t)}{dt} = Ax(t) + Bu(t - \tau) \]

Assume that the time delay \( \tau \) is less than or equal to the sampling period \( h \).

Show that the sampled system is given by

\[ x((k + 1)h) = Fx(kh) + G_0u(kh) + G_1u((k - 1)h) \]

where

\[ F = e^{Ah} \]
\[
G_0 = \int_0^{h-\tau} e^{At} B dt \\
G_1 = e^{A(h-\tau)} \int_0^{\tau} e^{At} B dt
\]

Hint: Split the integral
\[
\int_{kh}^{kh+h} e^{A(kh+h-s)} B u(s - \tau) \, ds
\]
into two parts.

1.3 Problem 3

Illustrate the aliasing effect of sampling by making a Matlab code where a continuous sinusoidal signal is sampled with different sampling frequencies. The frequency of the continuous sinusoidal should be 1 Hz. This can be done by using the Matlab command (give first a value to the variable ts)

\[
t = 0:ts:N; \quad \%ts = sampling interval \\
y = \sin(2\pi t); \\
\]

Show only the sampling instants when you plot the sampled signal. You can use the following type of Matlab code:

\[
\text{plot}(0:ts:N,y,'o')
\]

Try different sampling frequencies. For which sampling frequencies is it possible to reconstruct the continuous-time signal?
2 Homework assignment 2 – LQG design

Consider the DC servo model

\[ G(s) = \frac{1}{s(s + 1)} \]

which we represent in state space form as

\[
\begin{align*}
\dot{x} &= \begin{pmatrix} 0 & 1 \\ 0 & -1 \end{pmatrix} x + \begin{pmatrix} 0 \\ 1 \end{pmatrix} u \\
y &= \begin{pmatrix} 1 & 0 \end{pmatrix} x
\end{align*}
\]

Design linear controllers based on LQG theory for this system.

**Design constraints**

- It is required that the step response from \( r(t) \) to \( y(t) \) does not have any stationary error.
- During the design the input for a step response must at all times be constrained so that

\[ |u(t)| \leq 4 \forall t \]

Evaluate the closed loop system with the designed controller, and simulate it with the reference signal

\[ r(t) = \begin{cases} 
0, & 0 \leq t \leq 4 \\
1, & 4 < t \leq 8 
\end{cases} \]

**Problems to solve**

(a) Assume that both state variables are assessible. Use LQ theory in the design. What constraints have to be applied to the penalty matrices \( Q_1 \) and \( Q_2 \) in order to meet the design constraint on \( u(t) \)? Define the risetime of the closed loop system as the time it takes for the step response to go from 10% to 90% of its final value. How small can the risetime be made?

(b) Next assume that only \( y(t) = x_1(t) \) is available for measurements. Design a regulator using LQG theory. Simulate the closed loop system as above. Use the initial values

\[ x(0) = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \quad \hat{x}(0) = \begin{pmatrix} 0 \\ 0.5 \end{pmatrix} \]

How did you choose the matrices \( Q_1, Q_2, R_1, R_2 \)?

For both cases, include with your solution the program files you have used, and plots of the step responses (plot \( y(t) \) and \( u(t) \)).

**Hints**

Below is part of the Matlab code you can use to find the solution. You will have to input and test various matrices \( Q_1, Q_2, R_1, R_2 \).
% input matrices A,B,C,D; define open loop system
sysservo = ss(A,B,C,D)
% define reference signal
t1 = 0:0.01:8; t1 = t1(:);
r = ones(size(t1));
r(1:400) = zeros(400,1);

% LQ design
% input Q1 and Q2

% Compute L and S
[L,S] = lqr(A,B,Q1,Q2)
% Compute m
m =
% Define closed loop system with y and u as outputs
sysservo2 = ss(A-B*L,B*m,[C;-L],[D;m]);
% simulate the closed loop system, plot y and u
[z2, t2] = lsim(sysservo2,r,t1);
subplot(2,1,1)
plot(t2,z2(:,1))
ylabel('output'), xlabel('time')
subplot(2,1,2)
plot(t2,z2(:,2))
ylabel('input'), xlabel('time')
% compute max value of u
umax = max(abs(z2(:,2)))

% LQG case
% Input Q1, Q2, R1, R2

% Compute L and S
[L,S] = lqr(A,B,Q1,Q2)
% Determine m
m =
% Compute K and P
[K,P] = lqe(A, eye(2), C, R1, R2)
% Define closed loop system with y and u as outputs
sysservo3 = ss([A,-B*L;K*C,A-B*L-K*C],[B*m;B*m],[C,zeros(1,2);zeros(1,2),-L],[0;0;0;0.5]);
% Define initial values for the closed loop system
xinitial = [0; 0; 0; 0.5];
% Simulate closed loop system
[z3, t3] = lsim(sysservo3,r,t1,xinitial);
subplot(2,1,1)
plot(t3,z3(:,1))
ylabel('output'), xlabel('time')
subplot(2,1,2)
plot(t3,z3(:,2))
ylabel('input'), xlabel('time')
3 Homework assignment 3 – Phase plane analysis

Consider a simple model of two types of animals living in an isolated area. There are prey (such as rabbits) and predators (such as foxes). We assume that both prey and predators exist in large numbers, so we can describe their populations with real-valued numbers. Let \( x \) denote the number of prey individuals, and let \( y \) denote the number of predator individuals. There is unlimited food for the prey, so in absence of predators they would grow in number exponentially. On the other hand, meetings between a prey individual and a predator are assumed to be fatal for the prey. The number of prey individuals is modelled by the differential equation

\[
\frac{dx}{dt} = ax - bxy
\]

where \( a \) and \( b \) are positive parameters. Similarly, the number of predators is modelled as

\[
\frac{dy}{dt} = -cy + dxy
\]

with \( c \) and \( d \) positive. In the absence of prey, the number of predators will just decline. When there are many meetings between preys and predators (the product \( xy \) is large) the number of predators will grow.

Problems to solve

(a) Analyse the above model for \( x \) and \( y \). Determine the stationary points and their character.

(b) What happens if the initial value lies on one of the coordinate axes?

(c) Assume that the system is started strictly inside the first quadrant, that is \([x(0) > 0, \ y(0) > 0]\). Show that the solution will remain there for all future time.

(d) Simulate the system, using the parameter values \( a = 2, \ b = 1, \ c = 3, \ d = 1 \). Show the phase portrait, and plot also \( x(t) \) and \( y(t) \) versus time \( t \).

(e) Show that the orbits are closed curves, using the following idea. Use the original equations to derive an ODE for \( y(x) \) and write it in the form

\[
\frac{dy}{dx} = \frac{f_1(x)}{f_2(y)}
\]

and derive that the orbits fulfil

\[
\frac{y^c}{e^{by}} e^{dx} - K = 0
\]

where \( K \) is a constant.
(f) Let an arbitrary orbit have period \( T \). Determine its mean values

\[
\bar{x} = \frac{1}{T} \int_{t_1}^{t_1+T} x(t)dt \\
\bar{y} = \frac{1}{T} \int_{t_1}^{t_1+T} y(t)dt
\]

*Hint.* Divide the first equation by \( x \), and integrate.

(g) Will different orbits have the same period or not?

(h) Take an arbitrary orbit, with period \( T \), and the parameters listed in (d). Is it possible to choose other parameter values that give the same orbit, but with period \( T/2 \)?

(i) The model of this homework assignment is known as Volterra equations. Vito Volterra was a famous Italian mathematician, who, among other things, helped biologists to explain and model why the percentage of sharks increased significantly in the Mediterranean during World War I. During the war, fishing was reduced for natural reasons. In the model, then \( x \) corresponds to food fish, and \( y \) corresponds to sharks. The model should also be adjusted for the amount of fishing. Hence, consider the model

\[
\begin{align*}
\dot{x} &= ax - bxy - fx \\
\dot{y} &= -cy + dxy - fy
\end{align*}
\]

where \( f \) is proportional to the amount of fishing. Show that the ratio of averaged amount of sharks over the averaged amount of food fish, that is \( \bar{y}/\bar{x} \) indeed increases if \( f \) is decreased.

**Remark.** An old reference by Volterra in this context is


*Hint:* The phase portrait can be plotted in MATLAB by using the function `ode23` or `ode45`. You should first create a function where you define the system. Then call for the function in `ode45` or `ode23` and try different initial values. You may also be forced to use different time spans for different initial values to get a nice plot. The function `hold on` can be useful when you are plotting the portrait. (The current curves in a plot are then kept, and further curves can be added, using new plot commands.) The different directions associated with the curves in the portrait can be drawn by hand. You can use the following MATLAB framework.

```matlab
[t,X]=ode45('example',[0, tf],x0);
plot(X(:,1),X(:,2));
hold on;
```
Here ‘example’ is the name of an m-file, that defines the dynamics of your system, see below. \([0, tf]\) is the time interval over which the simulation is run. \(x0\) is a column vector, giving the initial state vector.

The dynamics is to be defined in a separate file, here named ‘example’ (you can replace it with another name of your choice). It should define the right hand side of the dynamics

\[
\dot{x} = f(t, x)
\]

for any given \(t\) and \(x\), and is used repeatedly by the Matlab routine \texttt{ode45} for simulating the system. The code should have the structure

```matlab
function xdot = system(t,x)
    xdot(1) = f1(t,x) % insert the expression for f1 here
    xdot(1) = f2(t,x) % insert the expression for f2 here
    xdot=xdot(:); % ensure that xdot is a column vector
```