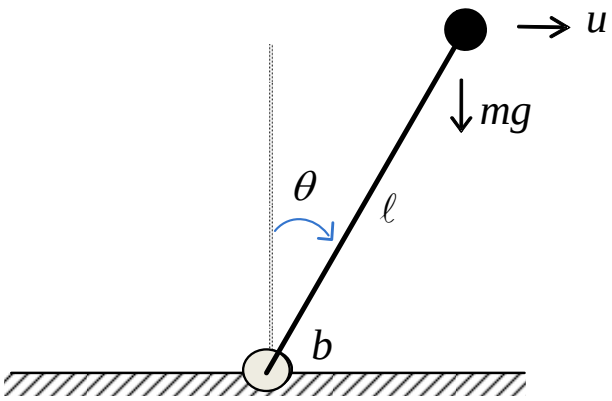
MEM 255 (W2019) Introduction to Controls HW#7

Consider the following simple inverted pendulum system. Note that unlike the cart-inverted pendulum, the base of the simple pendulum is fixed, and the control input force $u(t)$ is perpendicular to the gravity.



$$m\ell^2\ddot{\theta} + b\dot{\theta} - mg\ell\sin\theta = u\ell\cos\theta$$

(a) Use the Lagrange approach to verify the above equation of motion for the simple inverted pendulum system.

(b) Define the state variables $x_1 = \theta, \quad x_2 = \dot{\theta}$, and then find the nonlinear state equation of the system.

(c) Assume $\frac{g}{\ell} = 9, \ \frac{b}{m\ell^2} = 0.6, \ \frac{1}{m\ell} = 1$, and rewrite the nonlinear state equation.

(d) Show that $\bar{x}_U = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and $\bar{x}_D = \begin{bmatrix} \pi & 0 \end{bmatrix}^T$ are the equilibriums of the system associated with the upright and downward pendulum positions, respectively.

(e) Find the linearized state equation $\dot{x}(t) = A_U x(t) + B_U u(t)$ associated with the upright equilibrium $\bar{x}_U = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ and show that the equilibrium is unstable based on the eigenvalues of A_U.

(f) Build a Simulink/MATLAB program based on the nonlinear state equation obtained in (b), and use it to verify that the upright equilibrium $\bar{x}_U = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ is unstable.

(g) Find the linearized state equation $\dot{x}(t) = A_D x(t) + B_D u(t)$ associated with the downward equilibrium $\bar{x}_D = \begin{bmatrix} \pi & 0 \end{bmatrix}^T$ and show that the equilibrium is stable based on the eigenvalues of A_D.

(h) Use the same Simulink/MATLAB program you built in (f) to verify that the downward equilibrium $\bar{x}_D = \begin{bmatrix} \pi & 0 \end{bmatrix}^T$ is stable.

(i) Find a state feedback controller, $u(t) = Fx(t) = F_1 x_1(t) + F_2 x_2(t)$, so that the closed-loop system eigenvalues (the eigenvalues of $A_U + B_U F$) are in the strictly left half of the complex plane.

(j) Modify the Simulink/MATLAB program you used in (f) and (h) to incorporate the state feedback controller you designed in (i). Conduct simulation to verify that the originally unstable upright equilibrium $\bar{x}_U = \begin{bmatrix} 0 & 0 \end{bmatrix}^T$ has become stable.

Hint:

(d) $\dot{x} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} x_2 \\ ? \end{bmatrix} = f(x,u) = \begin{bmatrix} f_1(x_1, x_2, u) \\ f_2(x_1, x_2, u) \end{bmatrix}$

$u = 0, \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = 0 \rightarrow \begin{cases} x_2 = 0 \\ ?? = 0 \end{cases} \rightarrow \begin{bmatrix} \overline{x}_1 \\ \overline{x}_2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ or $\begin{bmatrix} \pi \\ 0 \end{bmatrix}$

(e) $\dot{x}(t) = A_U x(t) + B_U u(t)$

where $A_U = J_x = \left[\frac{\partial f}{\partial x}\right]_{\overline{x}} = \begin{bmatrix} \partial f_1 / \partial x_1 & \partial f_1 / \partial x_2 \\ \partial f_2 / \partial x_1 & \partial f_2 / \partial x_2 \end{bmatrix}_{\overline{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}} = \cdots\cdots$

$B_U = J_u = \left[\frac{\partial f}{\partial u}\right]_{\overline{x}} = \begin{bmatrix} \partial f_1 / \partial u \\ \partial f_2 / \partial u \end{bmatrix}_{\overline{x} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}} = \cdots\cdots$

Note that the eigenvalues of A_U are ? and ?, which verify that the equilibrium is unstable.

(f) Modify from the program in (j)

(g) $\dot{x}(t) = A_D x(t) + B_D u(t)$

where $A_D = J_x = \left[\frac{\partial f}{\partial x}\right]_{\overline{x}} = \begin{bmatrix} \partial f_1 / \partial x_1 & \partial f_1 / \partial x_2 \\ \partial f_2 / \partial x_1 & \partial f_2 / \partial x_2 \end{bmatrix}_{\overline{x} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}} = \cdots\cdots$

$B_D = J_u = \left[\frac{\partial f}{\partial u}\right]_{\overline{x}} = \begin{bmatrix} \partial f_1 / \partial u \\ \partial f_2 / \partial u \end{bmatrix}_{\overline{x} = \begin{bmatrix} \pi \\ 0 \end{bmatrix}} = \cdots\cdots$

Note that the eigenvalues of A_D are ??, which verify that the equilibrium is stable.

(h) Modify from the program in (j)

(i) $\dot{x}(t) = A_U x(t) + B_U u(t)$

Let the state feedback be $u(t) = Fx(t) = \begin{bmatrix} F_1 & F_2 \end{bmatrix} x(t)$, then the closed-loop system will be

$\dot{x}(t) = A_U x(t) + B_U u(t) = A_U x(t) + B_U F x(t) = \left(A_U + B_U F\right) x(t)$

Choose the desired close-loop systems poles to be λ_1 and λ_2, which are in the left half of the complex plane. Then find F_1 and F_2 as follows.

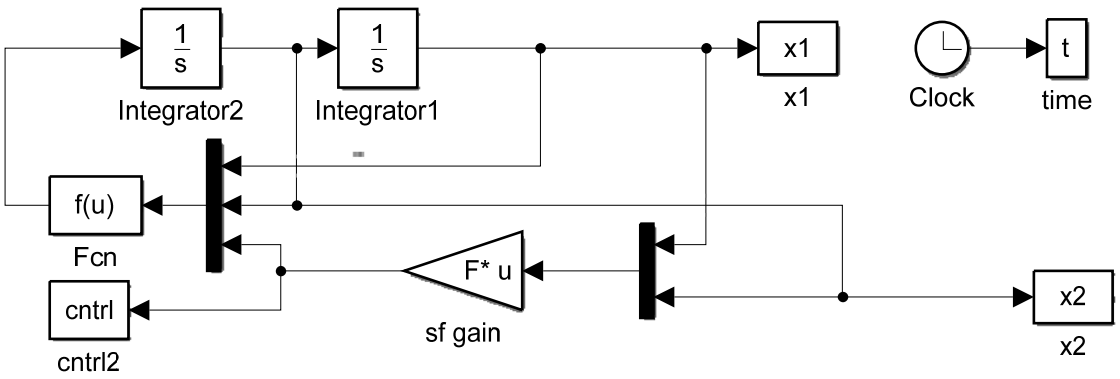
$\det\left[\lambda I - \left(A_U + B_U \begin{bmatrix} F_1 & F_2 \end{bmatrix}\right)\right] = (\lambda - \lambda_1)(\lambda - \lambda_2) \rightarrow F_1$ and F_2

(j) MATLAB programs

```
% filename: HW7_SIP_1.m,  Simple Inverted Pendulum
% BC Chang, Drexel University, on 2/22/2019
% Program running sequence:  MATLAB R2015a or later versions
```

```
% 1. HW7_SIP_1.m
% 2. HW7_SIP_tmrpns_2.mdl
% 3. HW7_plot_3.m

% x1_dot=x2
% x2_dot=9*sinx1-0.6*x2+cosx1*u
% Initial Conditions
x10_deg = 15   %in degrees
x10=x10_deg*3.1416/180  % in radians
x20=0

AU=[0  1; 9  -0.6]
BU=[0; 1]
eig(AU)

% u=F*x
%Choose desired damping ratio ze and natural frequency wn
ze=0.8
wn=5
F=[-9-wn^2  0.6-2*ze*wn]  % Use this to obtain closed-loop response

%Zero Feedback
%F=[0  0];

Acl=A+B*F
eig(Acl)

%Part 3 Simulation
% Using the the following script to run the simulink and plot the
% simulation results
sim_time=5
%sim_time = 20

sim_options = simset('SrcWorkspace ', 'current',...
    'DstWorkspace ', 'current');
%open('CSDInvP_sim_2_wFriction') %
%sim('CSDInvP_sim_2_wFriction', [0, sim_time], sim_options);
open('HW7_SIP_tmrpns_2') %
sim('HW7_SIP_tmrpns_2', [0, sim_time], sim_options);
run('HW7_plot_3')
```

HW7_SIP_tmrpns_2.mdl



In Fcn Expression box: type the following

9*sin(u(1))-0.6*u(2)+cos(u(1))*u(3)

In Integrator1 Initial condition box: type x10

In Integrator1 Initial condition box: type 0

In cntrl2 Variable name box: type cntrl

In x1 Variable name box: type x1

In x2 Variable name box: type x2

In sf gain Gain box: type F

In time Variable name box: type t

```
%Filename: HW7_plot3.m
%Run this program after HW7_SIP-1.m, and HW7_SIP_tmrpns_2.mdl
x1_deg=x1*180/pi;
x2_deg=x2*180/pi;

figure(10)
subplot(1,3,1)
plot(t,cntrl)  % plot cntrl
grid on
grid minor
xlabel('t'), ylabel('cntrl')

subplot(1,3,2)
plot(x1_deg,x2_deg)      % plot phase plane trajectory
grid on
grid minor
xlabel('x1'), ylabel('x2')

subplot(1,3,3)
plot(t,x1_deg,t,x2_deg)  % plot x1, x2 time responses
legend('x1 deg)','x2 (deg/s)')
grid on
grid minor
xlabel('t'), ylabel('x1 and x2')
```