16-642

PS1

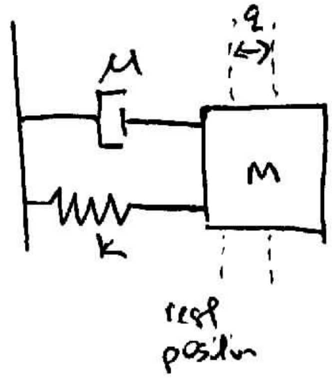
Boxiang Fu

boxiangf

09/18/2024

Q1.

a.



Newton's law gives $F = m\ddot{q}$, the following forces are present:

$F_{forced} = u$

$F_{damper} = -\mu\dot{q}$

$F_{spring} = -kq$

$\Rightarrow F = m\ddot{q}$

$\Rightarrow u - \mu\dot{q} - kq = m\ddot{q}$

$\Rightarrow m\ddot{q} + \mu\dot{q} + kq = u$ (1.1) is the system's ODE.

b. We define the state $x = \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \in \mathbb{R}^2$,

then $\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix}$

Rearranging Eq. 1.1, we have

$$\ddot{q} = \frac{-k}{M}q - \frac{\mu}{M}\dot{q} + \frac{u}{M} \quad (1.2) \quad \text{and}$$

$$\dot{q} = \dot{q} \quad (1.3)$$

In matrix form, we have

$$\dot{x} = \begin{bmatrix} \dot{q} \\ \ddot{q} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{\mu}{M} \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} u \quad (1.4)$$

Assuming $y=q$, we have

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} q \\ \dot{q} \end{bmatrix} \quad (1.5)$$

so $\dot{x} = Ax + Bu$ and $y = Cx$, where

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{\mu}{M} \end{bmatrix}, \quad B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 \end{bmatrix} \quad (1.6)$$

c. We assume $M>0$, $\mu>0$, and $k>0$ for physical plausibility

For an unforced system with $u=0$, we have $\dot{x} = Ax$ where $A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & -\frac{\mu}{M} \end{bmatrix}$

Since $\det A = 0 \cdot \frac{-\mu}{M} - 1 \cdot -\frac{k}{M} = \frac{k}{M} \neq 0$, the matrix A is invertible. Therefore, there is only one unique equilibrium point at $x_e = 0$ (Lec 3 Slide 5)

The stability at $x_e = 0$ is characterized by the eigenvalues of A. The characteristic equation gives:

$$\det(\lambda I - A) = 0$$

$$\Rightarrow 0 = \det\left(\begin{bmatrix} \lambda & -1 \\ \frac{k}{M} & \lambda + \frac{\mu}{M} \end{bmatrix}\right)$$

$$0 = \lambda\left(\lambda + \frac{\mu}{M}\right) - \frac{k}{M} \cdot -1$$

$$0 = \lambda^2 + \lambda \frac{\mu}{M} + \frac{k}{M}$$

The quadratic formula gives:

$$\lambda = \frac{-\frac{\mu}{M} \pm \sqrt{\left(\frac{\mu}{M}\right)^2 - 4(1)\left(\frac{k}{M}\right)}}{2(1)}$$

$$\lambda = \frac{-\frac{\mu}{M} \pm \sqrt{\frac{1}{M^2}\left(\mu^2 - 4Mk\right)}}{2}$$

$$\lambda = \frac{-\mu \pm \sqrt{\mu^2 - 4Mk}}{2m}$$

We have the following cases:

- If $\mu^2 - 4Mk \leq 0$, then $Re(\lambda) = \frac{-\mu}{2M} < 0$

  So $x_e = 0$ is globally asymptotically stable

- If $\mu^2 - 4Mk > 0$, then $Re(\lambda) = \frac{-\mu \pm \sqrt{\mu^2 - 4Mk}}{2M} \leq \frac{-\mu + \sqrt{\mu^2 - 4Mk}}{2m}$

$$< \frac{-\mu + \sqrt{\mu^2}}{2m}$$

$$< 0$$

So $x_e=0$ is also globally asymptotically stable. Therefore, the system is globally asymptotically stable (and thus stable) at equilibrium point $x_e=0$. The system is never unstable for physically plausible variable values.

d. For $x \in \mathbb{R}^2$, $u \in \mathbb{R}$, $\dot{x} = Ax + Bu$ is controllable iff $Q = [B \; AB]$ has $\det Q \neq 0$ (Lec 4 Slide 8)

$$Q = \left[\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \; \begin{bmatrix} 0 & 1 \\ -\frac{k}{m} & -\frac{\mu}{m} \end{bmatrix} \begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \right]$$

$$= \left[\begin{bmatrix} 0 \\ \frac{1}{m} \end{bmatrix} \; \begin{bmatrix} \frac{1}{m} \\ -\frac{\mu}{m^2} \end{bmatrix} \right]$$

$$= \begin{bmatrix} 0 & \frac{1}{m} \\ \frac{1}{m} & \frac{-\mu}{m^2} \end{bmatrix}$$

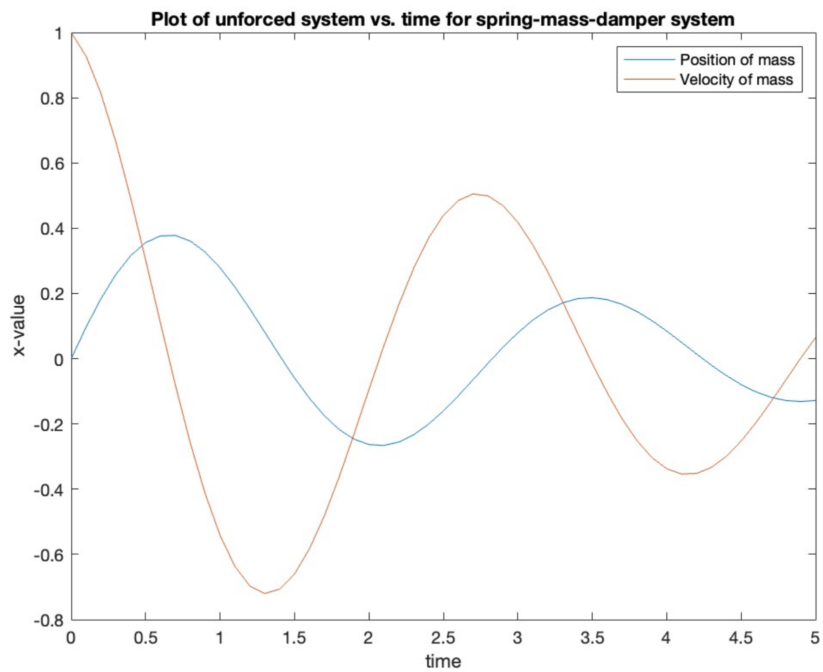
$$\det Q = 0 \cdot -\frac{\mu}{m^2} - \frac{1}{m} \cdot \frac{1}{m}$$

$$= -\frac{1}{m^2}$$

$$\neq 0$$

Therefore, (A, B) is controllable for all physically plausible values of the variables m, μ, and k.

e. For $m=1, \mu=0.5, k=5, t\in[0,5]$, we have:



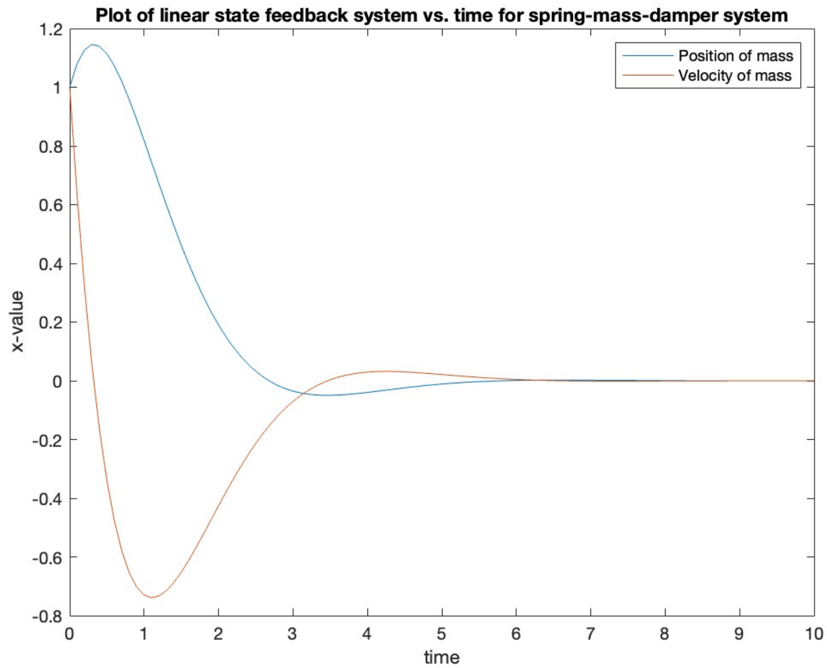
f. With eigenvalues $-1+i$ and $-1-i$, we obtain

$$K = [-3 \quad 1.5]$$

We prefer $A-BK$ to have eigenvalues λ where $Re(\lambda)<0$ because this makes the closed-loop system asymptotically stable about the equilibrium point. By choosing K so that the eigenvalues λ of $A-BK$ are $Re(\lambda)<0$, we can make unstable systems asymptotically stable at $x_e=0$ by implementing a linear control law $u=-Kx$.

9. For $M=1$, $\mu=0.5$, $k=5$, $t\in[0,10]$, $u(t)=-Kx(t)$,

we have:



Q2.

$$\gamma\ddot{x}_c-\beta\ddot{\phi}\cos\phi+\beta\dot{\phi}^2\sin\phi+\mu\dot{x}_c=F \qquad (2.1)$$

$$\alpha\ddot{\phi}-\beta\ddot{x}_c\cos\phi-D\sin\phi=0 \qquad (2.2)$$

a. Define $x=\begin{bmatrix}x_1\\x_2\\x_3\\x_4\end{bmatrix}=\begin{bmatrix}x_c\\\phi\\\dot{x}_c\\\dot{\phi}\end{bmatrix}$ and $u=F$

Let $M=\begin{bmatrix}\gamma & -\beta\cos\phi\\-\beta\cos\phi & \alpha\end{bmatrix}$

So $M^{-1}=\dfrac{1}{\gamma\alpha-\beta^2\cos^2\phi}\begin{bmatrix}\alpha & \beta\cos\phi\\\beta\cos\phi & \gamma\end{bmatrix}$

For the form $M\begin{bmatrix}\ddot{x}_c\\\ddot{\phi}\end{bmatrix}=N$, we get

$$N=\begin{bmatrix}\gamma & -\beta\cos\phi\\-\beta\cos\phi & \alpha\end{bmatrix}\begin{bmatrix}\ddot{x}_c\\\ddot{\phi}\end{bmatrix}$$

$$=\begin{bmatrix}\gamma\ddot{x}_c-\beta\ddot{\phi}\cos\phi\\-\beta\ddot{x}_c\cos\phi+\alpha\ddot{\phi}\end{bmatrix}$$

Comparing with Eq. 2.1 and 2.2, we get

$$N=\begin{bmatrix}u-\beta\dot{\phi}^2\sin\phi-\mu\dot{x}_c\\D\sin\phi\end{bmatrix} \qquad (2.3)$$

Noting that $M\begin{bmatrix}\ddot{x}_c\\\ddot{\phi}\end{bmatrix}=N$

$$\therefore \begin{bmatrix}\ddot{x}_c\\\ddot{\phi}\end{bmatrix}=M^{-1}N,$$

So $\begin{bmatrix} \ddot{x}_c \\ \ddot{\phi} \end{bmatrix} = \frac{1}{\gamma\alpha - \beta^2\cos^2\phi} \begin{bmatrix} \alpha & \beta\cos\phi \\ \beta\cos\phi & \gamma \end{bmatrix} \begin{bmatrix} u - \beta\dot{\phi}^2\sin\phi - \mu\dot{x}_c \\ D\sin\phi \end{bmatrix}$

$$= \frac{1}{\gamma\alpha - \beta^2\cos^2\phi} \begin{bmatrix} \alpha u - \alpha\beta\dot{\phi}^2\sin\phi - \alpha\mu\dot{x}_c + \beta D\cos\phi\sin\phi \\ \beta u\cos\phi - \beta^2\dot{\phi}^2\cos\phi\sin\phi - \beta\mu\dot{x}_c\cos\phi + \gamma D\sin\phi \end{bmatrix}$$

Noting that $\begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} = \begin{bmatrix} x_c \\ \phi \\ \dot{x}_c \\ \dot{\phi} \end{bmatrix}$ by definition, we get

$$\dot{x} = \begin{bmatrix} \dot{x}_c \\ \dot{\phi} \\ \ddot{x}_c \\ \ddot{\phi} \end{bmatrix} = \begin{bmatrix} x_3 \\ x_4 \\ \frac{1}{\gamma\alpha - \beta^2\cos^2 x_2}\left(\alpha u - \alpha\beta x_4^2\sin x_2 - \alpha\mu x_3 + \beta D\cos x_2\sin x_2\right) \\ \frac{1}{\gamma\alpha - \beta^2\cos^2 x_2}\left(\beta u\cos x_2 - \beta^2 x_4^2\cos x_2\sin x_2 - \beta\mu x_3\cos x_2 + \gamma D\sin x_2\right) \end{bmatrix} \quad (2.4)$$

which is highly non-linear

b. The Matlab solutions are:

```
>> MEC_Q2B
(- alpha*sin(x2)*beta*x4^2 + alpha*u - alpha*mu*x3 + cos(x2)*sin(x2)*D*beta)/(alpha*gamma - beta^2*cos(x2)^2)

(- cos(x2)*sin(x2)*beta^2*x4^2 + u*cos(x2)*beta + sin(x2)*D*gamma - mu*x3*cos(x2)*beta)/(alpha*gamma - beta^2*cos(x2)^2)
```

c. For an unforced state, the equilibrium points satisfy

$u = 0, \ \dot{x} = 0$

From Eq. 2.4.:

$$0 = \begin{bmatrix} x_3 \\ x_4 \\ \frac{1}{\gamma\alpha - \beta^2\cos^2 x_2}\left(\alpha u - \alpha\beta x_4^2 \sin x_2 - \alpha\mu x_3 + \beta D \cos x_2 \sin x_2\right) \\ \frac{1}{\gamma\alpha - \beta^2\cos^2 x_2}\left(\beta u \cos x_2 - \beta^2 x_4^2 \cos x_2 \sin x_2 - \beta\mu x_3 \cos x_2 + \gamma D \sin x_2\right) \end{bmatrix}$$

For row 1:

$\therefore x_3 = \dot{x}_c = 0$

For row 2:

$x_4 = \dot{\phi} = 0$

For row 3:

$$0 = \overset{=0}{\cancel{\alpha u}} - \overset{=0}{\cancel{\alpha\beta x_4^2}} \sin x_2 - \alpha \overset{=0}{\cancel{\mu x_3}} + \cancel{\beta} D \cos x_2 \sin x_2$$

$\therefore 0 = \cos x_2 \sin x_2 \qquad (2.5)$

For row 4:

$$0 = \beta \overset{=0}{\cancel{u \cos}} x_2 - \beta^2 \overset{=0}{\cancel{x_4^2 \cos x_2}} \sin x_2 - \beta\mu \overset{=0}{\cancel{x_3 \cos x_2}} + \cancel{\gamma D} \sin x_2$$

$\therefore 0 = \sin x_2$

So $x_2 = \phi = k\pi, \ k \in \mathbb{Z} \quad (2.6)$

Substituting (2.6) into (2.5), the relationship

$0 = \cos x_2 \sin x_2$ always holds

There are no restrictions on $x_1 = x_c$, so the equilibrium points are:

$$x_e = \begin{bmatrix} r \\ k\pi \\ 0 \\ 0 \end{bmatrix}, \quad r \in \mathbb{R},\ k \in \mathbb{Z}$$

In English, the equilibrium points are where the pendulum is upright and pointing vertically up or vertically down. The cart can be in any location x_c on the axis, but it's velocity and the pendulum's angular velocity must be zero. In other words, the cart is stationary and the pendulum is upright and stationary.

d. Let $A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 1 & -3 & 0 \\ 0 & 2 & -3 & 0 \end{bmatrix}$, $B = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$

Then, $\text{eig}(A) = \{0, -3.3301, 1.1284, -0.7984\}$

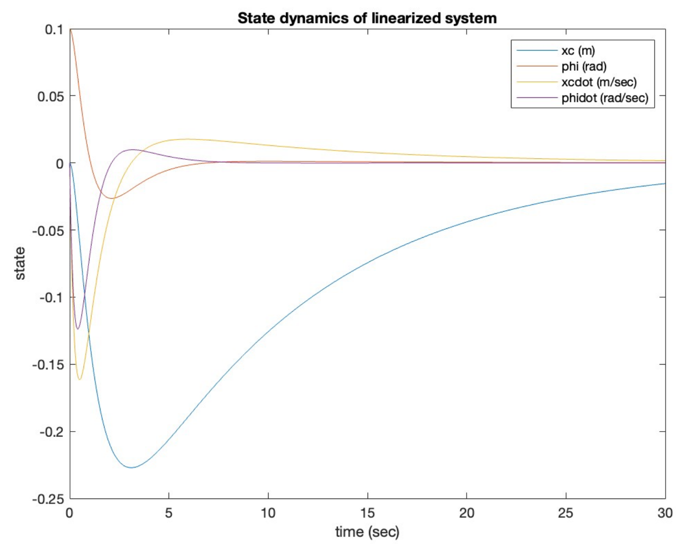
Since there is an eigenvalue (1.1284) with real part greater than 0, the linearized system is unstable at $x_e = 0$. By extension, the theorem in Lec 3 Slide 22 also tells us that the original non-linear system is also unstable locally (i.e. in a neighborhood around $x_e = 0$).

e. Using lqr(A, B, Qx, Qu), we obtain

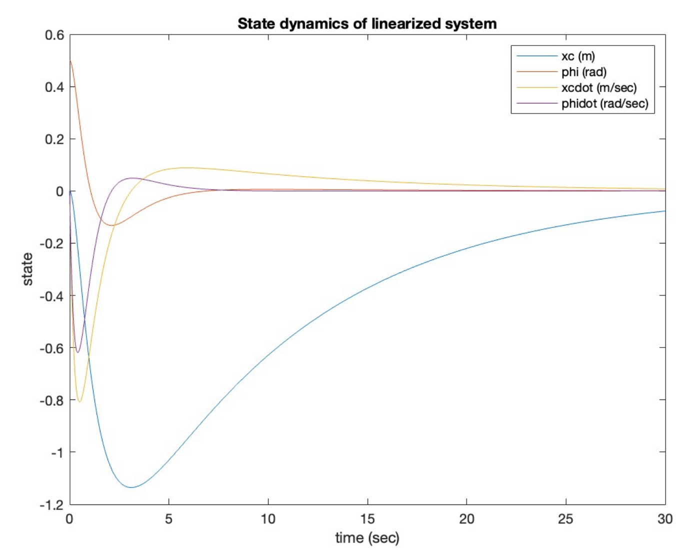
$K_c = [-0.3162 \quad 10.2723 \quad -6.7857 \quad 9.2183]$

The plots of the linearized system are:

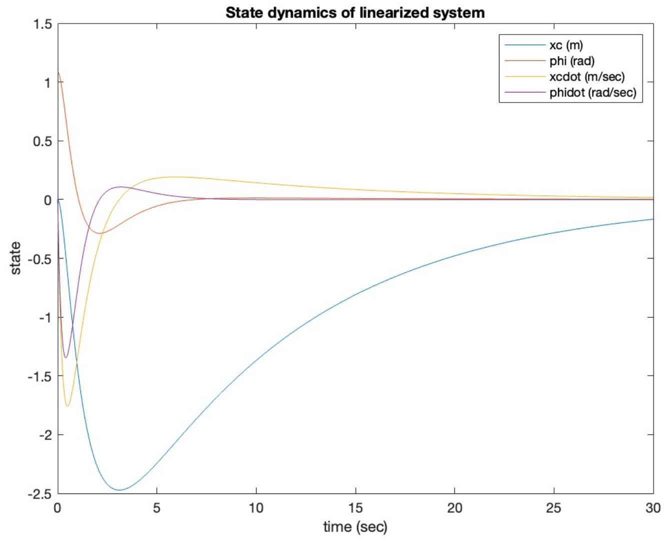
$x_0 = [0, 0.1, 0, 0]^T$



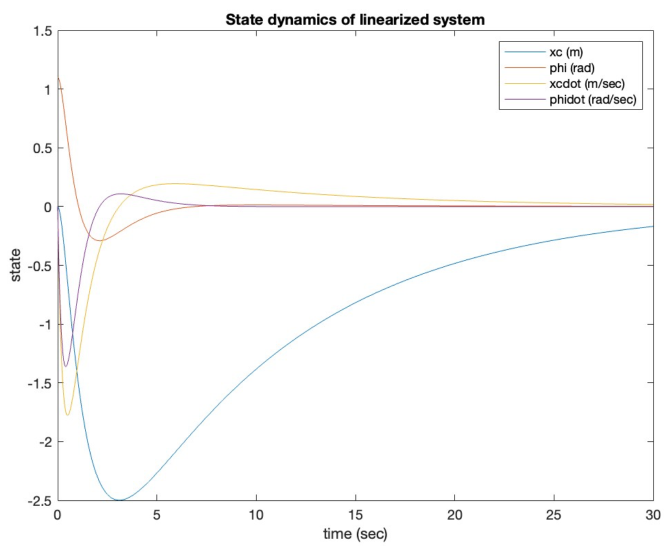
$x_0 = [0, 0.5, 0, 0]^T$



$x_0 = [0, 1.0886, 0, 0]^T$



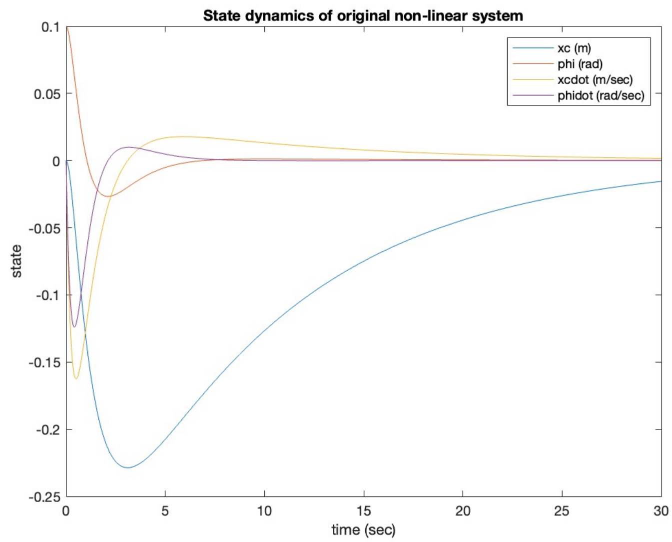
$x_0 = [0, 1.1, 0, 0]^T$



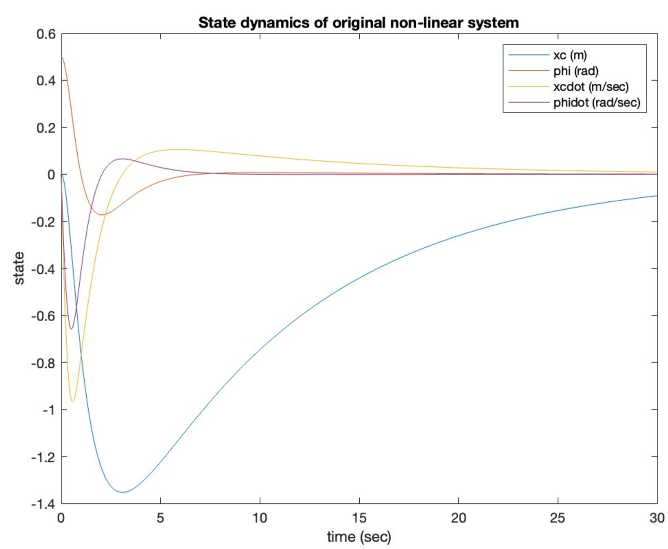
f. Using $K_c = [-0.3162 \quad 10.2723 \quad -6.7857 \quad 9.2183]$

The plots of the original non-linear system are:

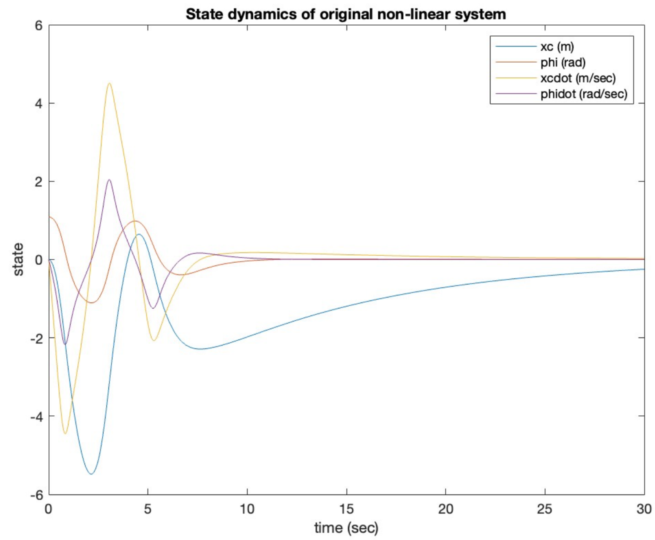
$x_0 = [0, 0.1, 0, 0]^T$



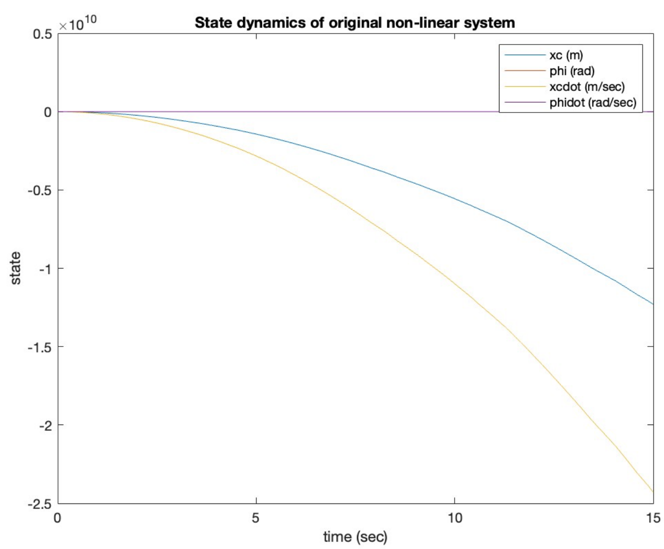
$x_0 = [0, 0.5, 0, 0]^T$



$x_0 = [0, 1.0886, 0, 0]^T$



$x_0 = [0, 1.1, 0, 0]^T$ (only for $t \in [0, 15]$)



The nonlinear and linear results are similar for the first 2 initial values (i.e. $\phi_0 = 0.1$ and $\phi_0 = 0.5$), but differ wildly for the latter two. For $\phi_0 = 1.0886$, the intermediate dynamics of x are drastically different than the linearized dynamics. However, the good thing is that it still converges to $x_e = 0$ asymptotically. The same cannot be said for $\phi_0 = 1.1$. The linear feedback K_e causes $x_e = 0$ to become unstable and results in x_c and $\dot{x}_c$ blowing up towards $-\infty$. The reason for this is because the linearization of this system is only through a first-order Taylor expansion. For more extreme x_0 values, the first-order approximation does not capture the full dynamics of the system.

As such, the linear feedback control K_e is not accurate for the actual nonlinear system as it is based on an inaccurate linearization. Therefore, at extreme x_0 values, the controller causes unwanted dynamics and may even cause the system to become unstable.

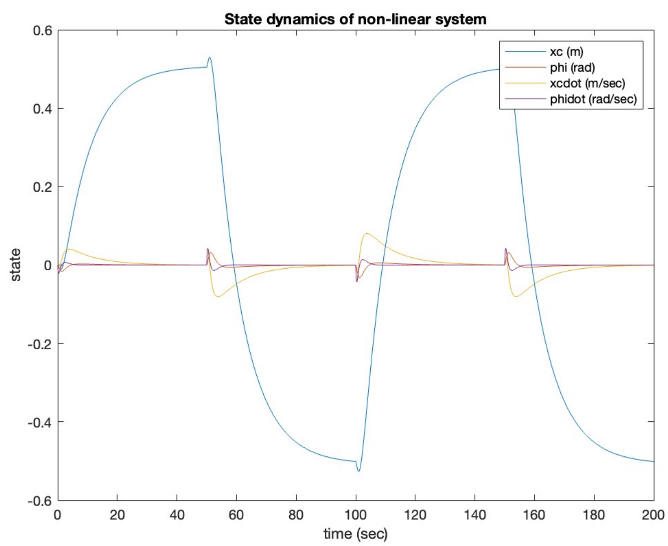
g. We note that a meter is 39.37 inches

The cart position in meters is x_1

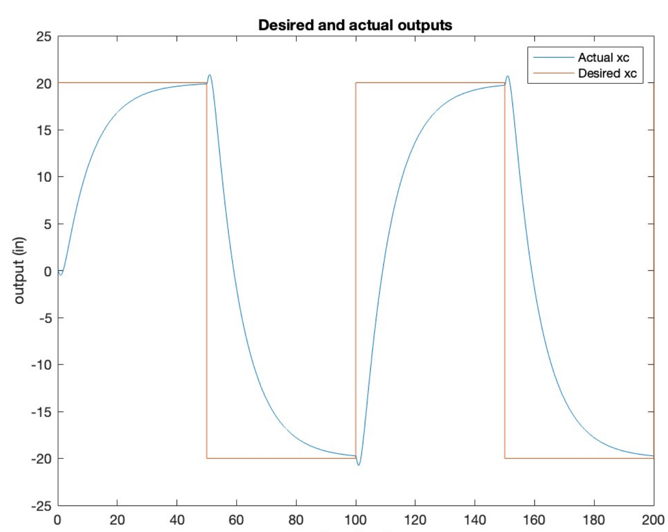
So $y = [39.37 \quad 0 \quad 0 \quad 0] \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$

$\therefore C = [39.37 \quad 0 \quad 0 \quad 0]$

h. State dynamics of nonlinear system for $x_0 = [0, 0, 0, 0]^T$



Desired and actual outputs.



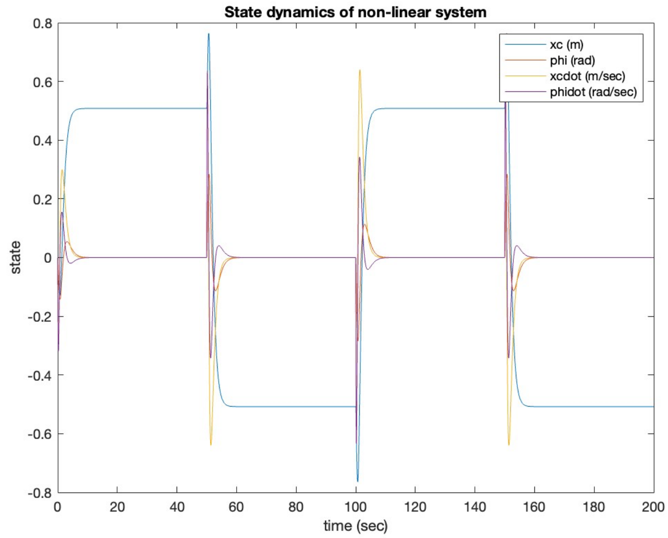
With a feedforward gain K_f, the controller attempts to make the output y converge to the desired output y_d. The controller does this by adding an external input $v = K_f y_d$ to input u. We see that y indeed converges to y_d, but the rate of convergence is slow, presumably due to $Q_x(1,1) = 1$. We tune this in the following section to get faster convergence.

c. We choose the new parameters to be

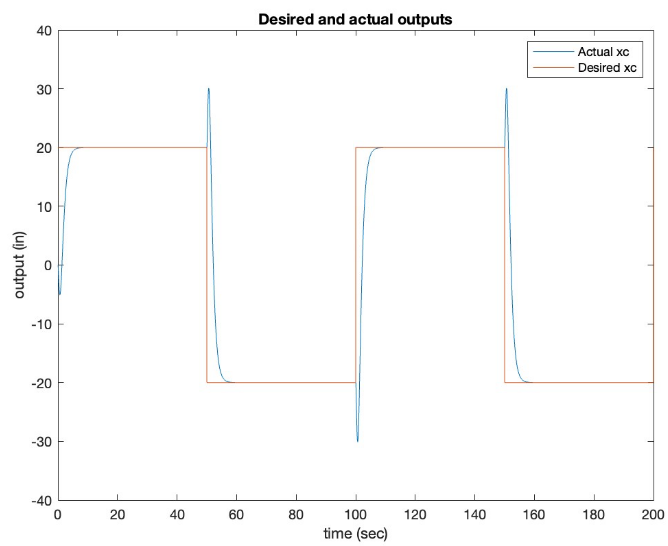
$$Q_x = \begin{bmatrix} 50 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 5 & 0 \\ 0 & 0 & 0 & 5 \end{bmatrix}, \quad Q_u = 1$$

We define "better" as being output y following close to the desired output y_d. We chose $Q_u = 1$ as we are less concerned about the control energy expended. We want to minimize $\|y_d - y\|$ at all costs. Thus, we chose $Q_x(1,1) = 50$ to set a high penalty for x_c not converging to its desired value. Since we do not care much for the convergence of ϕ, we only set $Q_x(2,2)$ a low penalty of one. From physical intuition, the velocity of the cart and pendulum affects the cart's kinematics greatly. Thus, we set the penalties $Q_x(3,3)$ and $Q_x(4,4)$ to 5 to make their velocities converge faster (and by extension making y converge to y_d more stable).
The results are plotted below:

State dynamics of non-linear system for $x_0 = [0,0,0,0]^T$



Desired and actual outputs



Compared with the outputs using the original Q_x and Q_u, we indeed notice a faster convergence to y_d, albeit at the expense of higher control energy expended.

Appendices:

A.1. Code for Q1

```
% MEC
% Q1E
clear;

% Parameters
m = 1;
mu = 0.5;
k = 5;
t = 5;

% Time span
tstep = 0.1;
t_vector = 0:tstep:t;

% Initial conditions
x0 = [0; 5];

% Populate A and B matrix
A = zeros(2,2);
A(1,1) = 0;
A(1,2) = 1;
A(2,1) = -k/m;
A(2,2) = -mu/m;

B = [0 ; 1/m];

% Solve for unforced linear system
x = [];
for time = 0:tstep:t
    curr_x = expm(A*time)*x0;
    x = [x, curr_x];
end

% Plot
figure;

plot(t_vector, x(1,:));
hold on
plot(t_vector, x(2,:));
title("Plot of unforced system vs. time for spring-mass-damper system");
legend("Position of mass","Velocity of mass");
xlabel("time");
ylabel("x-value");
hold off

% Q1F
% Desired eigenvalues
p = [complex(-1,1);
    complex(-1,-1)];

% K matrix
K = place(A,B,p);
eigs = eig(A-B*K);

% Q1G
% Parameters
t = 10;
x0 = [1; 1];

% Time span
tstep = 0.1;
t_vector = 0:tstep:t;

% Solve for linear state feedback system
x = [];
for time = 0:tstep:t
    curr_x = expm((A-B*K)*time)*x0;
```

```
    x = [x, curr_x];
end

% Plot
figure;

plot(t_vector, x(1,:));
hold on
plot(t_vector, x(2,:));
title("Plot of linear state feedback system vs. time for spring-mass-damper system");
legend("Position of mass","Velocity of mass");
xlabel("time");
ylabel("x-value");
hold off
```

A.2. Code for Q2B

```
% MEC
% Q2B
clear;

syms alpha beta gamma D mu u x3 xcdotdot x2 x4 phidotdot
eq1 = gamma * xcdotdot - beta * phidotdot * cos(x2) + beta * x4 * x4 * sin(x2) + mu * x3 == u;
eq2 = alpha * phidotdot - beta * xcdotdot * cos(x2) - D * sin(x2) == 0;
sol = solve(eq1, eq2, xcdotdot, phidotdot);
disp(sol.xcdotdot);
disp(sol.phidotdot);
```

A.3. Code for Q2C-G

```
% MEC
% Q2C
clear;

% Parameters
gamma = 2;
alpha = 1;
beta = 1;
D = 1;
mu = 3;

% Populate A and B matrix
A = zeros(4,4);
A(1,3) = 1;
A(2,4) = 1;
A(3,2) = 1;
A(3,3) = -3;
A(4,2) = 2;
A(4,3) = -3;

B = zeros(4,1);
B(3,1) = 1;
B(4,1) = 1;

% Eigenvalues of A
eigs = eig(A);

% Q matix for LQR
Qu = 10;

Qx = zeros(4,4);
Qx(1,1) = 1;
Qx(2,2) = 5;
Qx(3,3) = 1;
Qx(4,4) = 5;

```

```
% LQR
[K,S,P] = lqr(A,B,Qx,Qu);

% Timespan
T = 0.01;
tspan = [0 30];
t_vector = 0:T:30;

% Initial conditions
% x0 = transpose([0, 0.1, 0, 0]);
% x0 = transpose([0, 0.5, 0, 0]);
% x0 = transpose([0, 1.0886, 0, 0]);
% x0 = transpose([0, 1.1, 0, 0]);

% Run ode45 for linearized system
% [t, x] = ode45(@(t, x) odefun(t, x, A, B, K), t_vector, x0);

% Run ode45 for original non-linear system
[t, x] = ode45(@(t, x) odefunnl(t, x, gamma, alpha, beta, D, mu, K), t_vector, x0);

% For the [0, 1.1, 0, 0]^T non-linear system initial state, ode45 is not
% able to plot beyond 9.6 secs. Plotting using ode23t instead to 15 secs

% tspan = [0 15];
% [t, x] = ode15s(@(t, x) odefunnl(t, x, gamma, alpha, beta, D, mu, K), tspan, x0);
% t_vector = 0:(30/(length(x)-1)):30;

% Plotting
figure();
plot(t_vector,x(:,1));
hold on
plot(t_vector,x(:,2));
plot(t_vector,x(:,3));
plot(t_vector,x(:,4));
% title("State dynamics of linearized system");
title("State dynamics of original non-linear system");
legend("xc (m)", "phi (rad)", "xcdot (m/sec)", "phidot (rad/sec)");
xlabel("time (sec)");
ylabel("state");
hold off

% Create function for linearized ODE
function dxdt = odefun(t, x, A, B, K)
    dxdt = (A - B * K) * x;
end

% Create function for original non-linear ODE
function dxdt = odefunnl(t, x, gamma, alpha, beta, D, mu, K)
    u = -(K * x);
    x1 = x(1);
    x2 = x(2);
    x3 = x(3);
    x4 = x(4);

    dxdt = zeros(4,1);
    dxdt(1) = x3;
    dxdt(2) = x4;
    dxdt(3) = (-alpha*sin(x2)*beta*x4^2 + alpha*u - alpha*mu*x3 + cos(x2)*sin(x2)*D*beta)/(alpha*gamma - beta^2*cos(x2)^2);
    dxdt(4) = (- cos(x2)*sin(x2)*beta^2*x4^2 + u*cos(x2)*beta + sin(x2)*D*gamma - mu*x3*cos(x2)*beta)/(alpha*gamma - beta^2*cos(x2)^2);
end
```

A.4. Code for Q2H-I

```
% MEC
% Q2H
clear;

% Parameters
```

```
gamma = 2;
alpha = 1;
beta = 1;
D = 1;
mu = 3;

% Populate A, B, and C matrix
A = zeros(4,4);
A(1,3) = 1;
A(2,4) = 1;
A(3,2) = 1;
A(3,3) = -3;
A(4,2) = 2;
A(4,3) = -3;

B = zeros(4,1);
B(3,1) = 1;
B(4,1) = 1;

C = [39.37 0 0 0];

% Eigenvalues of A
eigs = eig(A);

% Q matix for LQR
Qu = 1;

Qx = zeros(4,4);
Qx(1,1) = 50;
Qx(2,2) = 1;
Qx(3,3) = 5;
Qx(4,4) = 5;

% LQR
[K,S,P] = lqr(A,B,Qx,Qu);

% Timespan
T = 0.01;
tspan = [0 200];
t_vector = 0:T:200;

% Initial conditions
x0 = transpose([0, 0, 0, 0]);

% Run ode45 for original non-linear system
[t, x] = ode45(@(t, x) odefunnl(t, x, gamma, alpha, beta, D, mu, K, A, B, C), t_vector, x0);

% Plotting
figure();
plot(t_vector,x(:,1));
hold on
plot(t_vector,x(:,2));
plot(t_vector,x(:,3));
plot(t_vector,x(:,4));
title("State dynamics of non-linear system");
legend("xc (m)", "phi (rad)", "xcdot (m/sec)", "phidot (rad/sec)");
xlabel("time (sec)");
ylabel("state");
hold off

yd = 20 * square(0.02 * pi * t_vector);
y = C * transpose(x);
figure();
plot(t_vector,y);
hold on
plot(t_vector,yd);
title("Desired and actual outputs");
legend("Actual xc", "Desired xc");
xlabel("time (sec)");
ylabel("output (in)");
hold off

% Create function for original non-linear ODE
```

```
function dxdt = odefunnl(t, x, gamma, alpha, beta, D, mu, K, A, B, C)
    % Find feedback control from linearized ODE
    yd = 20 * square(0.02 * pi * t);
    v = -inv((C * inv(A - B * K) * B)) * yd;
    u = -(K * x) + v;

    x1 = x(1);
    x2 = x(2);
    x3 = x(3);
    x4 = x(4);

    dxdt = zeros(4,1);
    dxdt(1) = x3;
    dxdt(2) = x4;
    dxdt(3) = (-alpha*sin(x2)*beta*x4^2 + alpha*u - alpha*mu*x3 + cos(x2)*sin(x2)*D*beta)/(alpha*gamma - beta^2*cos(x2)^2);
    dxdt(4) = (- cos(x2)*sin(x2)*beta^2*x4^2 + u*cos(x2)*beta + sin(x2)*D*gamma - mu*x3*cos(x2)*beta)/(alpha*gamma - beta^2*cos(x2)^2);
end
```