Alexander Kyu

BME 355

Project Part 2

Step 1:

See attached work on next page for the full work:

NOTE: Initial Evaluation of the system with the given equations displayed illogical physics. Please see below:

With the given parameters:

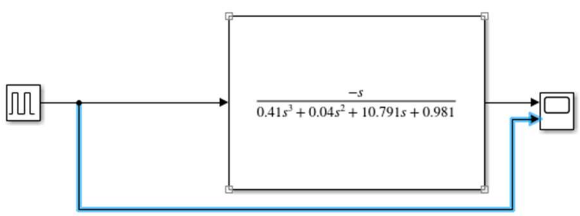
M = 1 m = 0.1

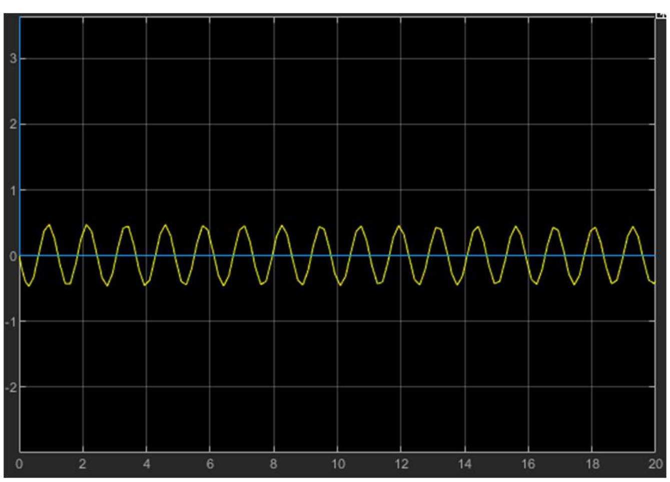
b = 0.1 l = 0.3

I = $\frac{ml^2}{3} = 0.003$ g = 9.81

$$G_p(s) = \frac{-2.439s}{s^3 + 0.09756s^2 + 26.32s + 2.393}$$

In an open loop configuration, we expect an unstable system after an impulse input:

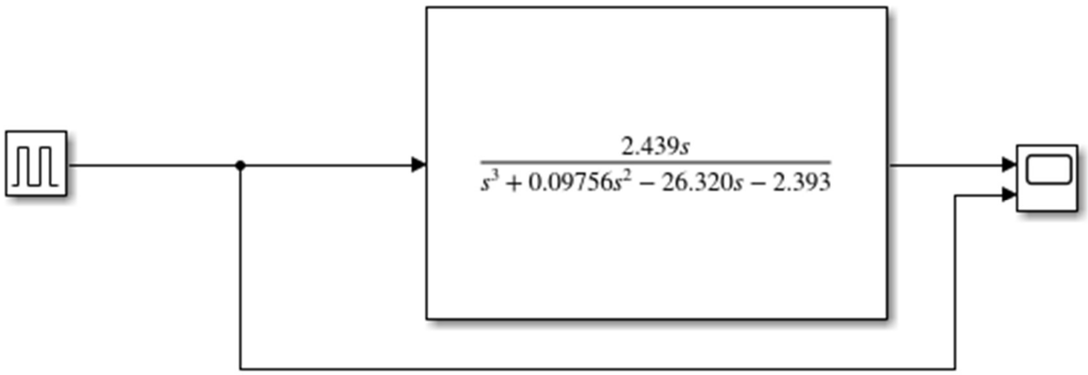


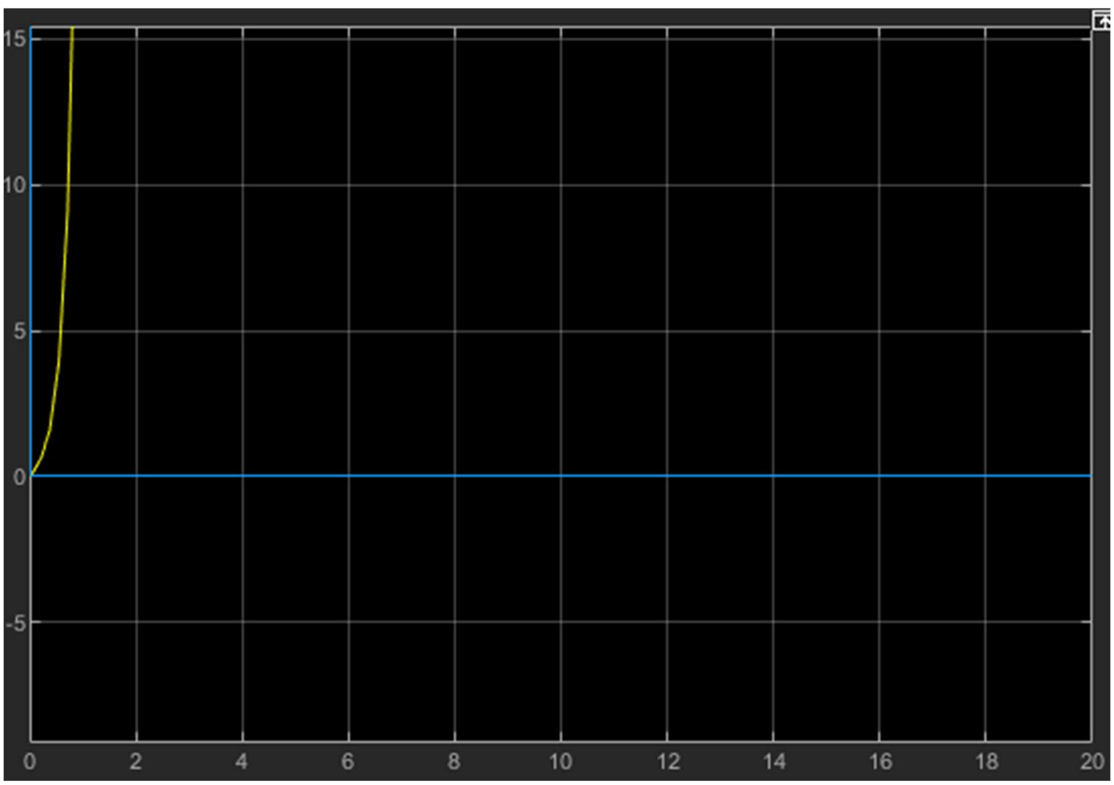


This, however, was not the case. Instead, a marginally system occurred which physically doesn't make sense. A re-evaluation of the initial equations was done, and several sign changes were made, resulting in the following new plant transfer function.

$$G_p(s) = \frac{2.439s}{s^3 + 0.09756s^2 - 26.32s - 2.393}$$

In an open loop configuration, we get the unstable system that makes sense after an impulse response:





BME 355 Equations

Non-linear

1) $F = (M+m)\ddot{x} + b\dot{x} - ml\ddot{\theta}\cos(\theta) + ml\dot{\theta}^2\sin(\theta)$

2) $0 = (I+ml^2)\ddot{\theta} - mgl\sin(\theta) - ml\ddot{x}\cos(\theta)$

☆ Note: the initial signs were all positive. Several reviews of the setup suggests incorrect signs.

Linear Forms

1) $F = (M+m)\ddot{x} + b\dot{x} - ml\ddot{\theta} + ml\dot{\theta}^2\theta$

2) $0 = (I+ml^2)\ddot{\theta} - mgl\theta - ml\ddot{x}$

$M = 1$
$m = 0.1$
$b = 0.1$
$l = 0.3$

$I = \frac{ml^2}{3} = 0.003$
$g = 9.81$

Simplifying the Equations:

1) $F = (M+m)\ddot{x} + b\dot{x} - ml\ddot{\theta} + ml\dot{\theta}^2\theta$

$F = (1.1)\ddot{x} + 0.1\dot{x} - 0.03\ddot{\theta} + 0.03\dot{\theta}^2\theta$

$\mathcal{L}$

$F(s) = 1.1s^2X(s) + 0.1sX(s) - 0.03s^2\theta(s) + \cancel{0.03s\dot{\theta}^2(s)\theta(s)} \rightarrow 0$

for linearization purposes & small angle approx., this can be eliminated.

2) $0 = (I+ml^2)\ddot{\theta} - mgl\theta - ml\ddot{x}$

$0 = (0.012)\ddot{\theta} - 0.2943\theta - 0.03\ddot{x}$

$0 = 0.012s^2\theta(s) - 0.2943\theta(s) - 0.03s^2X(s)$

$$X(s) = \frac{0.012s^2\theta(s) - 0.2943\theta(s)}{0.03s^2} = \frac{0.012s^2 - 0.2943}{0.03s^2}\theta(s)$$

Combined Equation

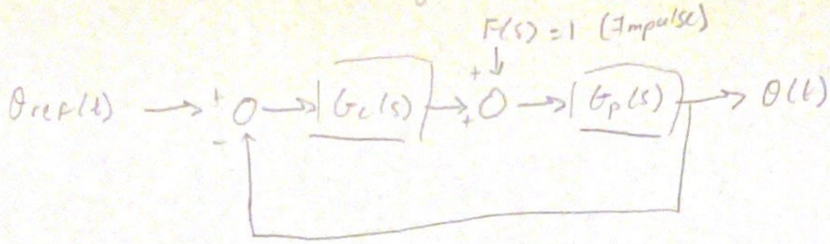
$$F(s) = 1.1s^2\left(\frac{0.012s^2 - 0.2943}{0.03s^2}\right)\theta(s) + 0.1s\left(\frac{0.012s^2 - 0.2943}{0.03s^2}\right)\theta(s) - 0.03s^2\theta(s)$$

$$F(s) = \theta(s)\left(0.44s^2 - 10.791 + \frac{0.04s^2 - 0.981}{s} - 0.03s^2\right)$$

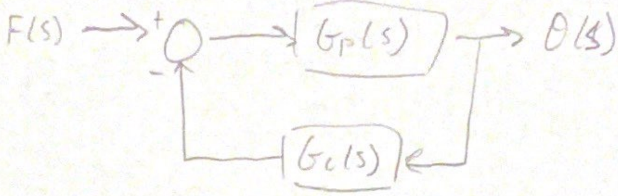
$$F(s) = \theta(s)\left(0.41s^2 + 0.04s - 10.791 - \frac{0.981}{s}\right)$$

$$G_p(s) = \frac{\theta(s)}{F(s)} = \frac{1}{\left(0.41s^2 + 0.04s - 10.791 - \frac{0.981}{s}\right)} = \frac{2.439s}{(s^3 + 0.09756s^2 - 26.32s - 2.39}$$

Current System Configuration



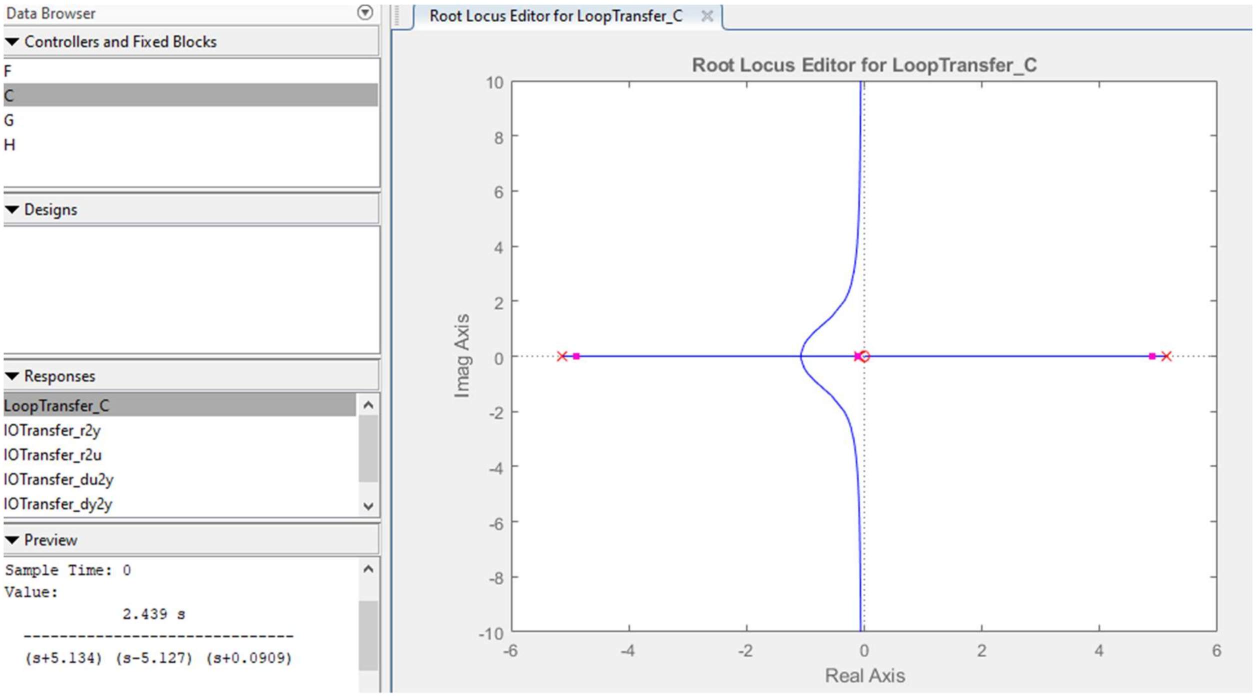
Rearranged

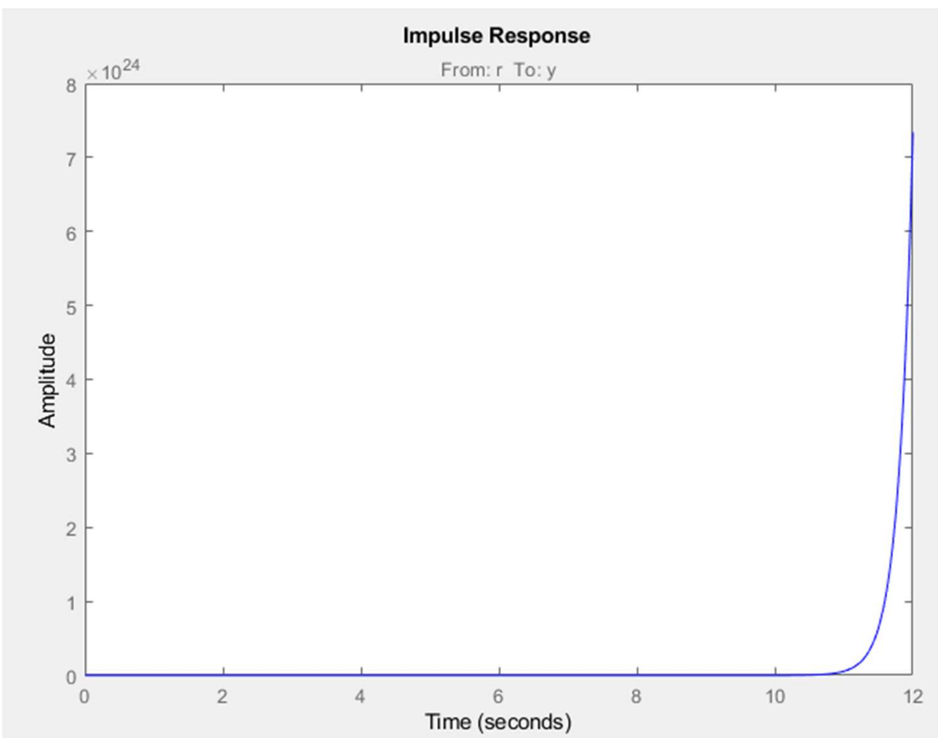


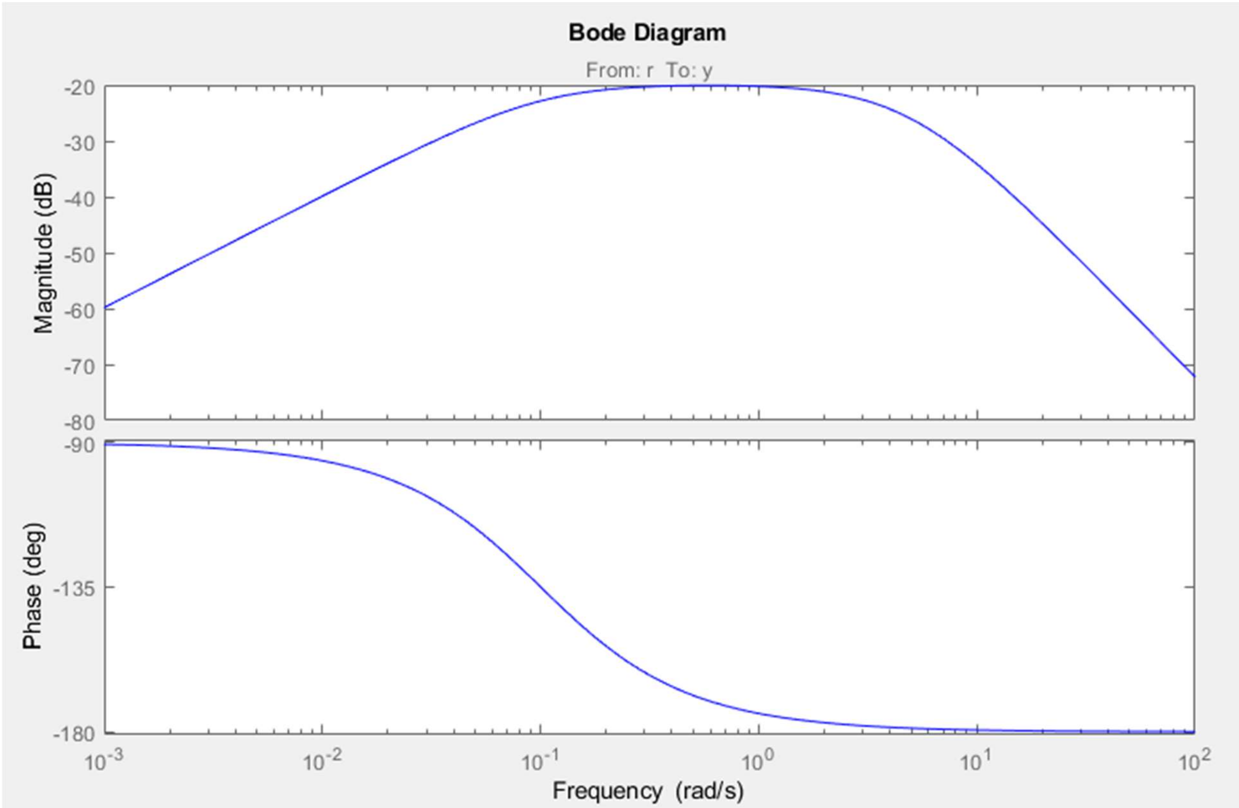
$$T(s) = \frac{G_p(s)}{1 + G_p(s)G_c(s)}$$

Step 2.1:

Root Locus with Gc(s) as a proportional controller K = 1







With this proportional controller of K = 1, the time response makes sense with an unstable system with an input of an impulse. Looking at the frequency response, we get a bandpass-like response with both low and high frequency inputs attenuated quite a bit. There is also a phase shift of -90 that increases as the input frequency increases.

Step 2.2:

With the given plant and nature of the problem, several key components were identified for the compensator:

1. A proportional component is needed to continuously keep track of the error
2. An integrator component is needed to make sure that the steady-state error goes to 0, which ensures that the system stays stable
3. A phase-lead component is needed. This type of compensator helps increase stability which is very much needed for this system. The lowpass sub-component was added to reduce any high frequency noise.

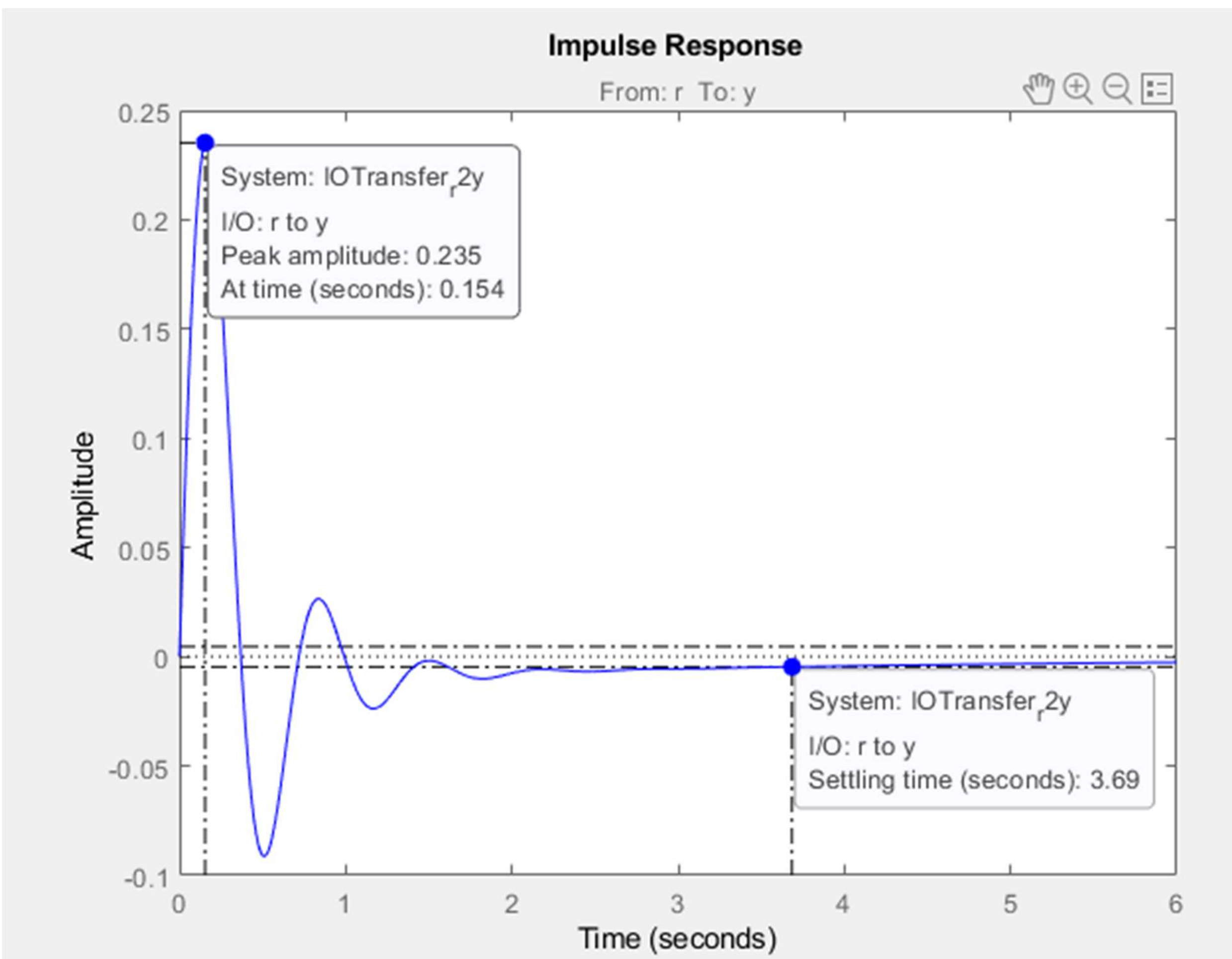
These components were added, and the following compensator was created:

$$G_c(s) = \frac{65.524(s+6)(s+0.1984)}{s(s+12)} = \underbrace{33.837}_{\textbf{P}} + \underbrace{\frac{0.0992}{s}}_{\textbf{I}} + \underbrace{\frac{31.687s}{s+12}}_{\textbf{Phase-Lead}}$$

Let's go a bit more into each component. First, we will start with the proportional component. The gain on this component is crucial to getting the system under control. Any smaller of a gain would prevent the controller from stabilizing the system. Second, we'll talk about the integrator component. Any form of this compensator is required, but the gain doesn't have to be very big. As stated before, this just helps accumulate the error to eliminate steady state error. Lastly, let's talk about the phase-lead component. The gain here is quite crucial too. The relatively high gain for the proportional component creates quite a few oscillations (number of sways), so to compensate for this, a higher gain is required to reduce the number of sways. Both gains (proportional and integrator), also help with the settling time by increasing the rate of decay.

When looking at the overall zeros and poles, the locations are quite important too. The zero at -0.1984 helps offset the pole created by the differentiator at 0. While the corresponding locations of the zero at -6 and pole at -12 create a positive angle, increasing stability. It is also important that the zero at the -6 is close to the root locus of the plant as it helps "pull" the root-locus left, which helps to further increase stability. If it was too far from the plant's root locus, then the effect would be diminished.

The impulse response with a total of 9 sways is shown below:



$$65.524\,\frac{(s+6)(s+0.1981)}{s(s+12)} = \frac{A}{1} + \frac{B}{s} + \frac{Cs}{s+12}$$

$$65.524s^2 + 406.144s + 1.1904 = As^2 + 12As + Bs + 12B + Cs^2$$

$A + C = 65.524$

$12A + B = 406.144$

$12B = 1.1904$

$B = 0.0992$

$A = 33.837$

$C = 31.687$

$C_1 = \frac{C}{12} = 2.641 \quad C_2 = 12$

$$\underbrace{33.837}_{P} + \underbrace{\frac{0.0992}{s}}_{I} + \underbrace{\frac{31.687s}{s+12}}_{\text{Phase-Lead}}$$

Part 3:

For the animation, the code below was used. To calculate the position of the cart, the transfer function X(s)/ θ(s) was used. The code, Simulink, and animation video are attached. The code is also below:

```
sim('model_simulink.slx')

time = out.theta.time;
theta = out.theta.signals.values;
x_total = out.x.signals.values;

% Initialize video
myVideo = VideoWriter('invertedPendulumAnimation'); %open video file
myVideo.FrameRate = 30;  %can adjust this, 5 - 10 works well for me
open(myVideo)

%In each frame, you know the position of cart x and theta th; therefore you
%can put these code into function, e.g. function animation_pedulum(x,th)
%%Define the dimension of the cart; you can change based on the visual
%%needed.
W = 1; % cart width
H = 0.5; % cart height
wr = 0.3; % wheel radius
mr = 0.05; % mass radius; You can change these dimension
L = 0.3;
for i=1:length(time)
    th = theta(i) - pi;
    x = x_total(i);
    % Define positions of cart and two wheels
    y = wr/2+H/2; % cart vertical position
    w1x = x-.9*W/2; % Wheels position
    w1y = 0;
    w2x = x+.9*W/2-wr;
    w2y = 0;
    % pendulum position
    pendx = x + L*sin(th); % L is the length of inverted pendulum. Please define yourself
    pendy = y - L*cos(th);
    plot([-10 10],[0 0],'k','LineWidth',2), hold on
    time_txt = sprintf('Time=%0.2f seconds', time(i));
    text(0.5, 1, time_txt)
    rectangle('Position',[x-W/2,y-H/2,W,H],'Curvature',.1,'FaceColor',[1 0 0],'LineWidth',1.5); %
Draw cart
    rectangle('Position',[x-.9*W/2,0,wr,wr],'Curvature',1,'FaceColor',[0 0 1],'LineWidth',1.5); %
Draw wheel
    rectangle('Position',[x+.9*W/2-wr,0,wr,wr],'Curvature',1,'FaceColor',[0 0
1],'LineWidth',1.5); % Draw wheel
    plot([x pendx],[y pendy],'k','LineWidth',2); % Draw pendulum
    rectangle('Position',[pendx-mr/2,pendy-mr/2,mr,mr],'Curvature',1,'FaceColor',[0 1
0],'LineWidth',1.5);
    axis([-2 2 -1 1.5]);axis equal
    set(gcf,'Position',[100 100 1000 400])
    drawnow, hold off

    frame = getframe(gcf); %get frame
    writeVideo(myVideo, frame);
end
close(myVideo)
```