# Project 1

Ross Smyth

Cart and rod system

**Table of Contents**

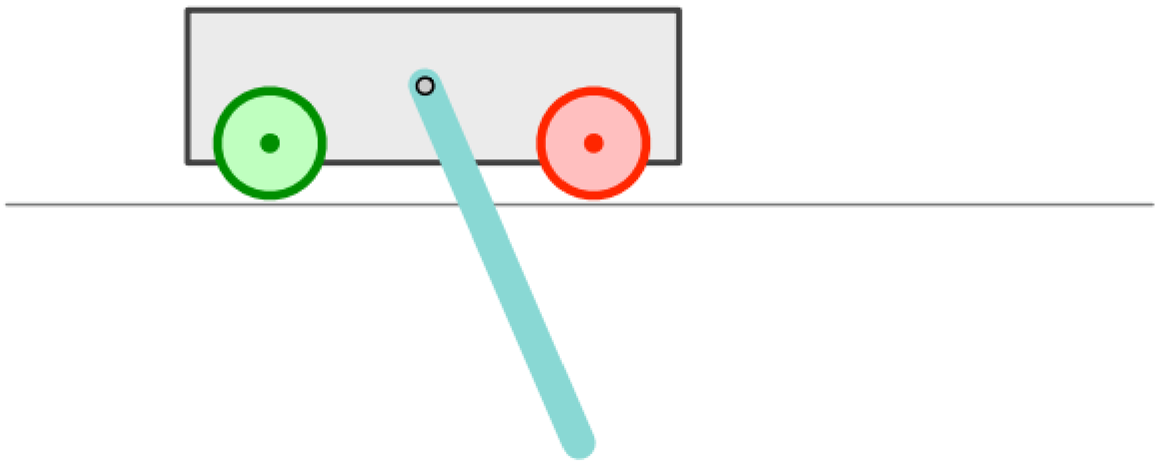


```
clear, clc, close all
addpath 'scripts'
addpath 'models'
```

## Summary


This report will go over the derivations, work, outputs, and discussion of result of Project 1 in MEEM 4775. This project analyzes a system of a cart that has a motor, gearbox, and rod on it. One of the wheels is an idler wheel that is not connected to the motor or gearbox. The other wheel is connected to the gearbox, which is connected to the motor. Also on the cart is a rod that can freely swing on an axle as the cart moves.

This project will demonstrate the derivation of a differential equations model of the cart. This will then be linearized and used to model the cart in Simulink, which where it will be fed an input voltage and the response verified with another model. After that the differential equation model will be further simplified, and the rod and cart equations will be decoupled from each other ("decoupled model"). This model will then again be simulated in Simulink, and then compared with the first model ("coupled model") in its reponse to a step function. After this is done a proportional control system will be designed to not overshoot the demanded position by 20%. This will be done analytically. Once designed the proportional control will be applied to both of the models and the response compared against each other. Finally two model models will be ran. One with a sine wave demanded position with a frequency of 1 rad/s and another with a sine wave input of the natrual frequency of the rod.

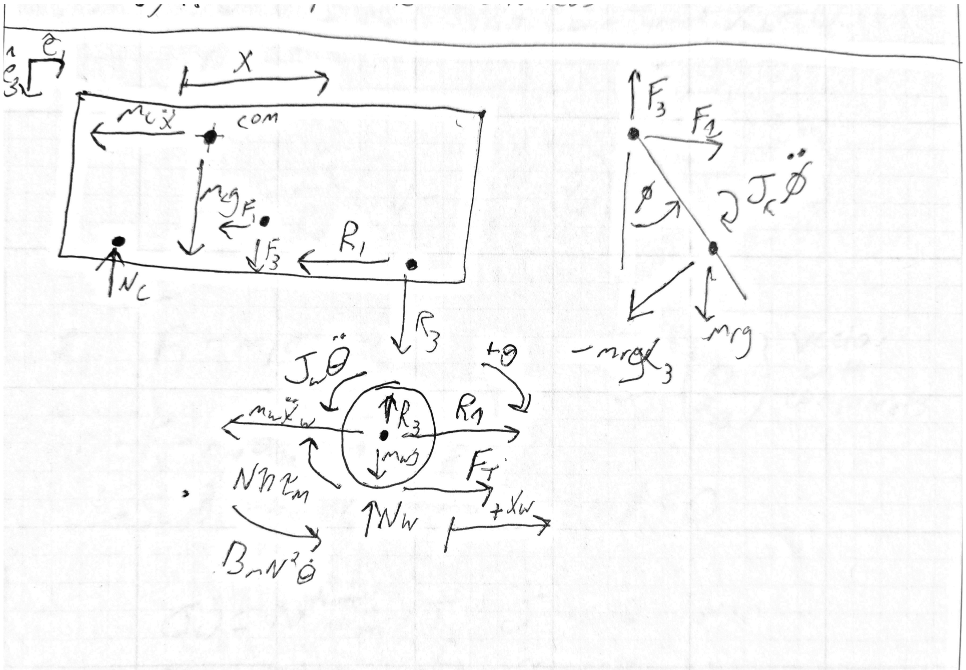


## Task 1: Deriving Diff EQ Model

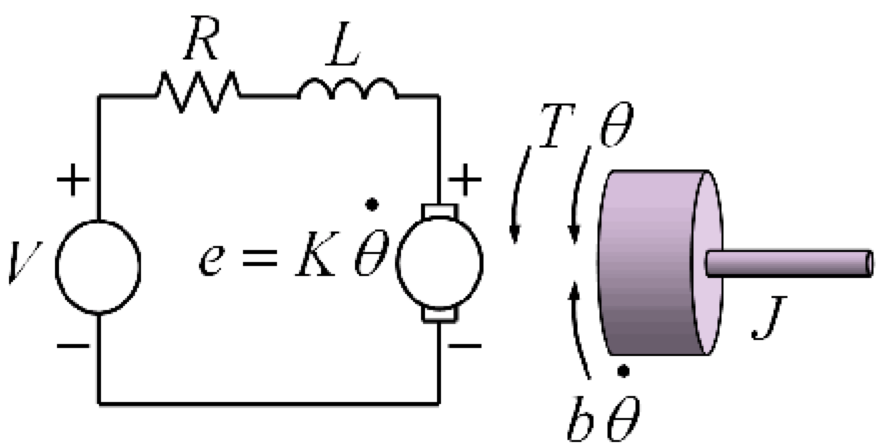
### Assumptions

1. No slip between wheels and track
2. Idler wheel axle is frictionless
3. Idler wheel is massless
4. Motor inductance is negligible
5. Cart's wheels remain on the track surface
6. Rod's pin is frictionless

### Free-Body Diagram and Circuit Diagram



In this diagram the back-EMF is labeled b while the equations have it labeled Ke.




## Equations

$$
\begin{cases}
\text{Cart :} \\
\qquad 0 = -m_c\ddot{x} - R_1 - F_1 \\
\text{Wheel :} \\
\qquad 0 = F_T + R_1 - m_w\ddot{x} \\
\qquad 0 = N\eta\tau_m - F_T r - \dfrac{Bm_wN^2}{r}\dot{x_a} - \dfrac{J_W}{r}\ddot{x} \\
\text{Motor :} \\
\qquad \tau_m = \dfrac{K_t}{R_a}\left(V - \dfrac{K_cN}{r}\dot{x}\right) \\
\text{Rod :} \\
\qquad 0 = F_1 - m_r a_{31} \\
\qquad 0 = m_r g - F_3 - FL_c \\
\qquad 0 = -J_r\ddot{\phi} - FL_c\cos(\phi) - F_3L_c\sin(\phi) \\
\text{Notes :} \\
\qquad J_w = N^2J_m + J_3 \\
\qquad m_t = m_c + m_w
\end{cases}
$$

## Kinematics

$$\overrightarrow{P_3} = \overrightarrow{P}_A + \overrightarrow{P_{\frac{4}{A}}} + \overrightarrow{P_{\frac{3}{4}}}$$

$$\overrightarrow{P_3} = \begin{bmatrix} x_A \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} d \\ 0 \\ 0 \end{bmatrix} + \begin{bmatrix} L_c\sin(\phi) \\ 0 \\ L_c\cos(\phi) \end{bmatrix}$$

$$\ddot{\overrightarrow{P_3}} = \begin{bmatrix} \ddot{x_A} + L_c\cos(\phi) - L_c\dot{\phi}^2\sin(\phi) \\ 0 \\ -L_c\ddot{\phi}\sin(\phi) - L_c\dot{\phi}\cos(\phi) \end{bmatrix} = \begin{bmatrix} a_{13} \\ a_{23} \\ a_{33} \end{bmatrix}$$

## Simplify

$$\left(m_T + m_r + \frac{J_w}{r^2}\right)\ddot{x_A} + m_rL_c\cos(\phi)\ddot{\phi} + \left(\frac{N}{r}\right)^2\left(B_m + \frac{\eta + K_tK_e}{R_a}\right)\dot{x_A}$$

$$-m_rL_c\sin(\phi)\dot{\phi} = \frac{N\eta K_tV}{rR_a}$$

$$0 = m_rL_c\cos(\phi)\ddot{x} + \left(J_r + m_rL_c^2\right)\ddot{\phi} + m_rL_c\, g\sin(\phi)$$

# Task 2: Linearizing the Model

## Linearize

1. Assume $\phi$ is a small angle

2. Assume $\dot{\phi}$'s magnitude is small

- $\sin(\phi) \approx \phi$
- $\cos(\phi) \approx 1$
- $\dot{\phi} \approx 0$

$$\left(m_t + m_r + \frac{J_w}{r^2}\right)\ddot{x_a} + m_r L_c \ddot{\phi} + \left(\frac{N}{r}\right)^2 \left(B_m + \frac{\eta K_t K_e}{R_a}\right)\dot{x_A} = \frac{N\eta K_t V}{rR_a}$$
$$m_r L_c \ddot{x_A} + \left(J_r + m_r {L_c}^2\right)\ddot{\phi} + m_r L_c g\phi = 0$$

### Matrix/State Space

2nd order/mass matrix form

$$M = \begin{bmatrix} m_t + m_r + \frac{J_w}{r} & m_r L_c \\ m_r L_c & J_r + m_r {L_c}^2 \end{bmatrix}$$
$$\widetilde{C} = \begin{bmatrix} \left(\frac{N}{r}\right)^2 \left(B_m + \frac{\eta K_t K_e}{R_a}\right) & 0 \\ 0 & 0 \end{bmatrix}$$
$$K = \begin{bmatrix} 0 & 0 \\ 0 & m_r L_c g \end{bmatrix}$$
$$\widetilde{B} = \begin{bmatrix} \frac{N\eta K_t}{rR_a} \\ 0 \end{bmatrix}$$

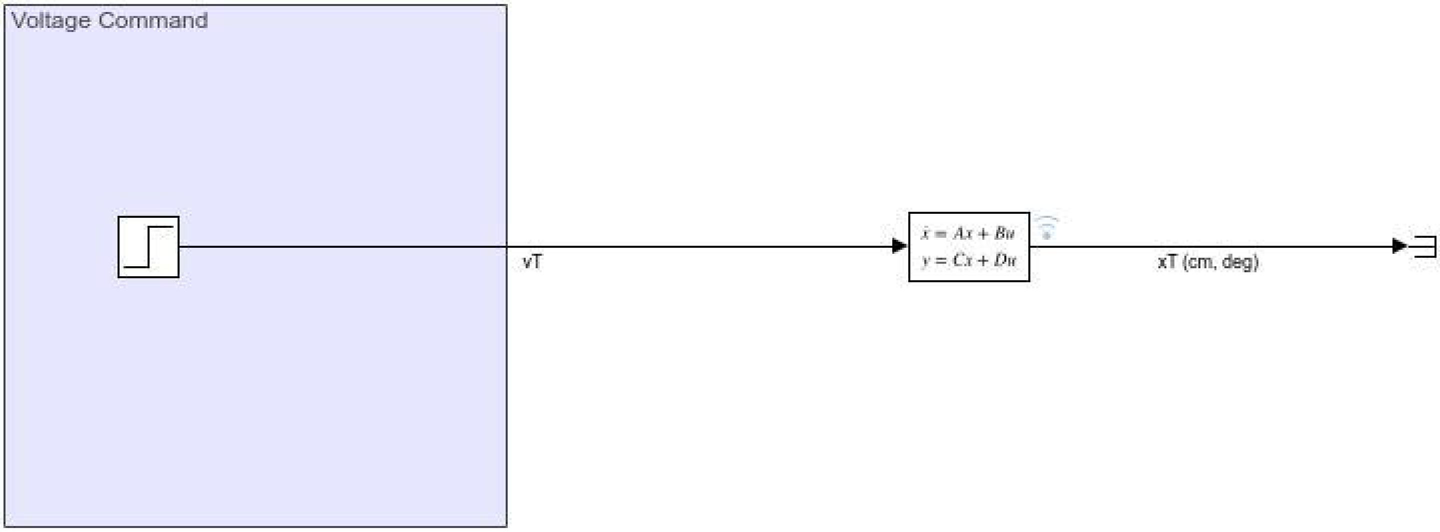
$$M \begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \widetilde{C} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + K \begin{bmatrix} x \\ \theta \end{bmatrix} + \widetilde{B}\, V$$
$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = M^{-1}\left(\widetilde{C} \begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + K \begin{bmatrix} x \\ \theta \end{bmatrix} + \widetilde{B}\, V\right)$$

States space representation:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix}(t) = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix}(t) = \begin{bmatrix} \bar{0} & I_2 \\ M^{-1}K & M^{-1}\widetilde{C} \end{bmatrix} \vec{x}\,(t) + \begin{bmatrix} \bar{0} \\ M^{-1}\widetilde{B} \end{bmatrix} \vec{u}\,(t)$$
$$\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = I_4 x(t) + \bar{0} u(t)$$

### Simulink Model

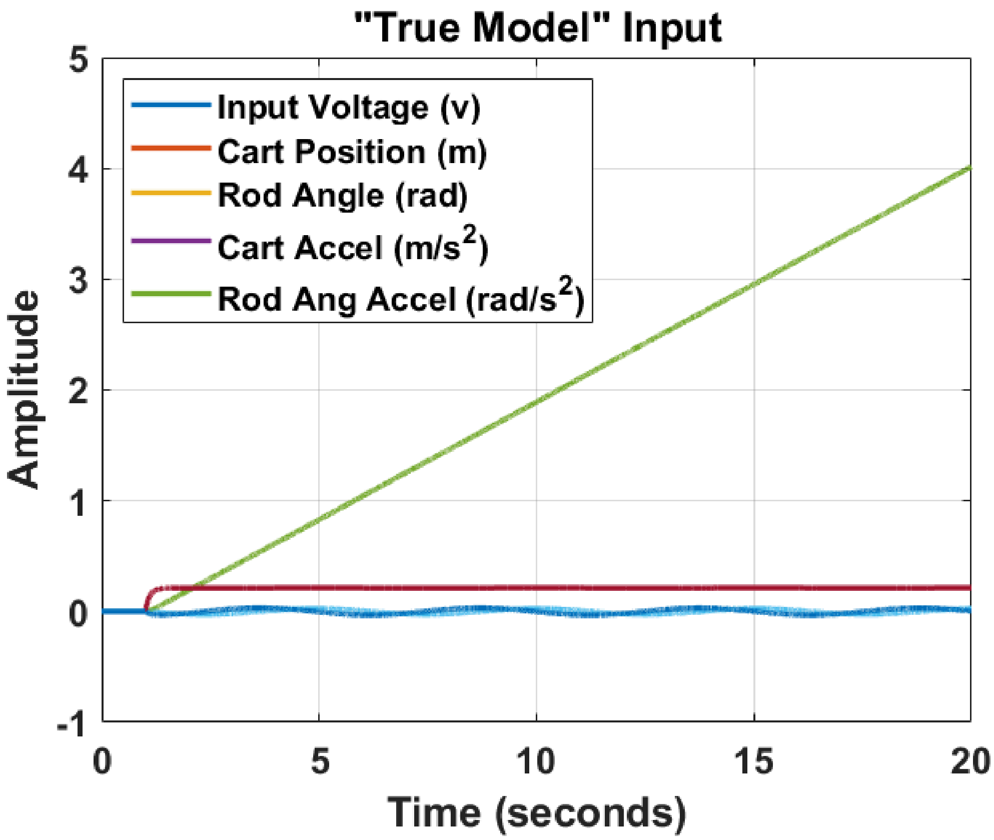
The simulink model developed is very basic. First a step input was put in, then a state-space block was attached that represents the system derived above. The signal out of the state-space block is logged, and then is input to a terminator. There is a file called "simTrue.m" that has the model's physical parameters and the state-space matrices. This is called in this file to setup the simulation, and then the simulation is ran.




```
run('setupTrue.m') % Setup model parameters

true_out = sim('simTrue');

figure()
plot(true_out.logsout{1}.Values, 'LineWidth', 2)
hold on
plot(true_out.logsout{1}.Values, 'LineWidth', 2)
title('"True Model" Input')
ylabel('Amplitude')
legend({'Input Voltage (v)', 'Cart Position (m)', 'Rod Angle (rad)', 'Cart Accel (m/s^2)', 'Roc
grid on
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
hold off
```




The true model was validated against Marcello Guadagno's Simulink model. We both chose a step input of 1 volt to apply to the model. The reponse of the models looked very similar and therefore we concluded that our models are valid against each other. Looking at other models on the Project 1 discussion post, the plots look similar to those as well.

## Task 3: Making an Uncoupled Model

Make an uncoupled, linearized model. This model will include the couple model but the terms that couple them will be removed. Start out with the coupled model:

$$\left(m_t + m_r + \frac{J_w}{r^2}\right)\ddot{x_a} + m_r L_c \ddot{\phi} + \left(\frac{N}{r}\right)^2 \left(B_m + \frac{\eta K_t K_e}{R_a}\right)\dot{x_A} = \frac{N\eta K_t V}{r R_a}$$

$$m_r L_c \ddot{x_A} + \left(J_r + m_r L_c^2\right)\ddot{\phi} + m_r L_c g \phi = 0$$

Terms that couple the models:

- $\ddot{\phi}$
- $\ddot{x_A}$

I will remove $\ddot{x_A}$ from the second equation and $\ddot{\phi}$ from the first.

$$\left(m_t + m_r + \frac{J_w}{r^2}\right)\ddot{x_a} + \left(\frac{N}{r}\right)^2\left(B_m + \frac{\eta K_t K_e}{R_a}\right)\dot{x_A} = \frac{N\eta K_t V}{rR_a}$$

$$\left(J_r + m_r L_c^{\,2}\right)\ddot{\phi} + m_r L_c g\phi = 0$$

**Matrix/State Space**

2nd order/mass matrix form

$$M = \begin{bmatrix} m_t + m_r + \frac{J_w}{r} & 0 \\ 0 & J_r + m_r L_c^{\,2} \end{bmatrix}$$

$$\widetilde{C} = \begin{bmatrix} \left(\frac{N}{r}\right)^2\left(B_m + \frac{\eta K_t K_e}{R_a}\right) & 0 \\ 0 & 0 \end{bmatrix}$$

$$K = \begin{bmatrix} 0 & 0 \\ 0 & m_r L_c g \end{bmatrix}$$

$$\widetilde{B} = \begin{bmatrix} \frac{N\eta K_t}{rR_a} \\ 0 \end{bmatrix}$$

$$M\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = \widetilde{C}\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + K\begin{bmatrix} x \\ \theta \end{bmatrix} + \widetilde{B}\,V$$

$$\begin{bmatrix} \ddot{x} \\ \ddot{\theta} \end{bmatrix} = M^{-1}\left(\widetilde{C}\begin{bmatrix} \dot{x} \\ \dot{\theta} \end{bmatrix} + K\begin{bmatrix} x \\ \theta \end{bmatrix} + \widetilde{B}\,V\right)$$

States space representation:

$$\begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \end{bmatrix}(t) = \begin{bmatrix} \dot{x} \\ \dot{\theta} \\ \ddot{x} \\ \ddot{\theta} \end{bmatrix}(t) = \begin{bmatrix} \overline{0} & I_2 \\ M^{-1}K & M^{-1}\widetilde{C} \end{bmatrix}\vec{x}(t) + \begin{bmatrix} \overline{0} \\ M^{-1}\widetilde{B} \end{bmatrix}\vec{u}(t)$$

$$\begin{bmatrix} x \\ \theta \\ \dot{x} \\ \dot{\theta} \end{bmatrix} = I_4 x(t) + \overline{0}u(t)$$

```
run('setupBoth.m') % Setup model parameters
```

## Task 5: Designing a Proportional Controller

It can easily be seen by looking above from Task 3 that the tranfer function for voltage in to positoin out is the following:

$$\frac{V(s)}{X(S)} = \frac{c_3}{c_1 s^2 + c_2 s}$$

Where

$$c_1 = \left(m_t + m_r + \frac{J_w}{r^2}\right)\ddot{x_a}$$

$$c_2 = \left(\frac{N}{r}\right)^2 \left(B_m + \frac{\eta K_t K_e}{R_a}\right)$$

$$c_3 = \frac{N\eta K_t}{rR_a}$$

And so adding propotional control takes the following form:

$$\frac{X(s)}{X_r(x)} = \frac{c_3 K_p}{c_1 s^2 + c_2 s + c_3 K_p}$$

Using the closed form solution for percent overshoot, $\xi$ can be calculated for 20% overshoot, then $\omega_n$ can be calculated too, leading to $K_p$.

$$\%M_p = e^{-\frac{(\pi\zeta)}{\sqrt{1-\zeta^2}}}$$

$$c_1 = 2\xi\omega_n$$

$$\omega_n^2 = K_p c_2$$

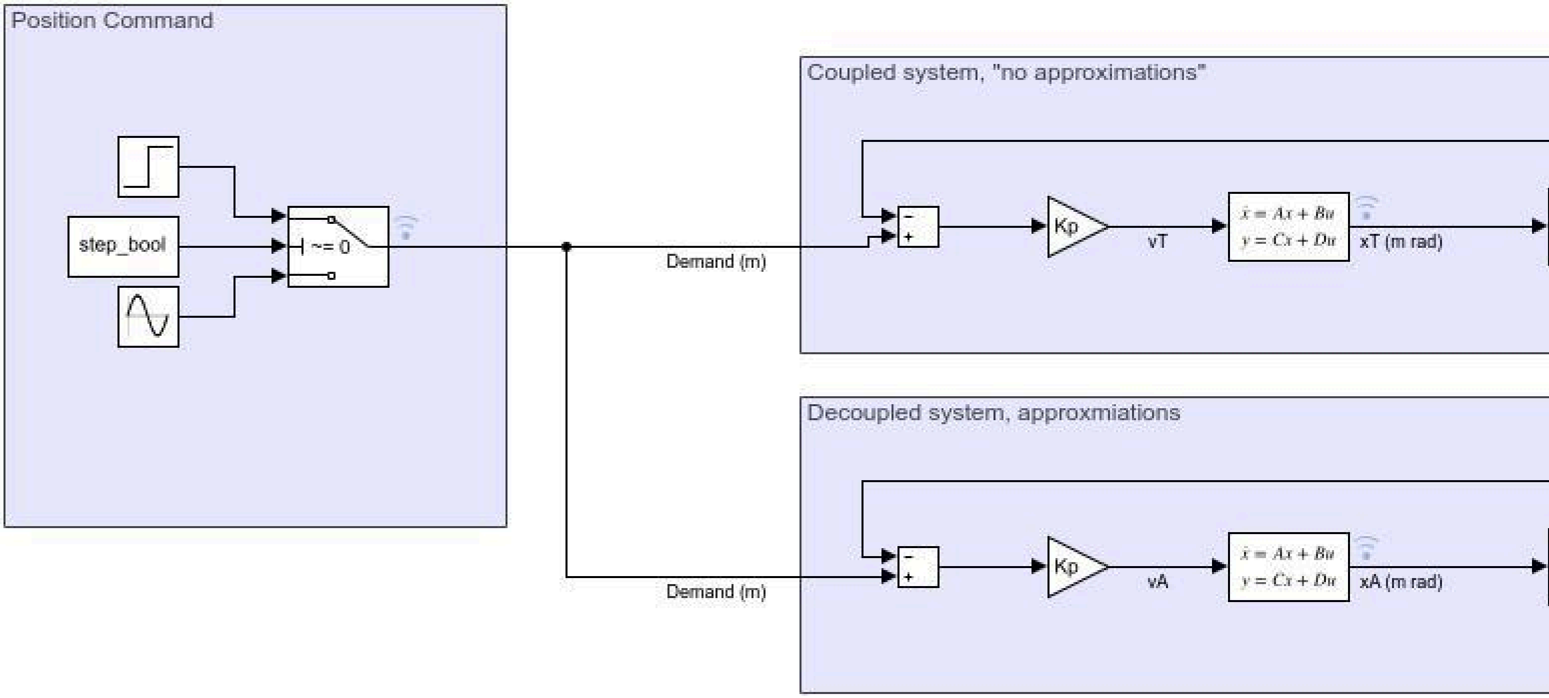
```
zeta = -log(0.2)/sqrt(log(0.2)^2+pi^2);
c1 = mt + mr + Jm/r^2;
c2 = (N/r)^2*(Bm+n*Kt*Ke/Ra);
wn = c1 / 2 / zeta;
Kp = wn^2/c2
```

```
Kp = 0.0952
```

Theoretically the $K_p$ value of 0.095 should result in a controller with 20% overshoot.

### Simulink Development

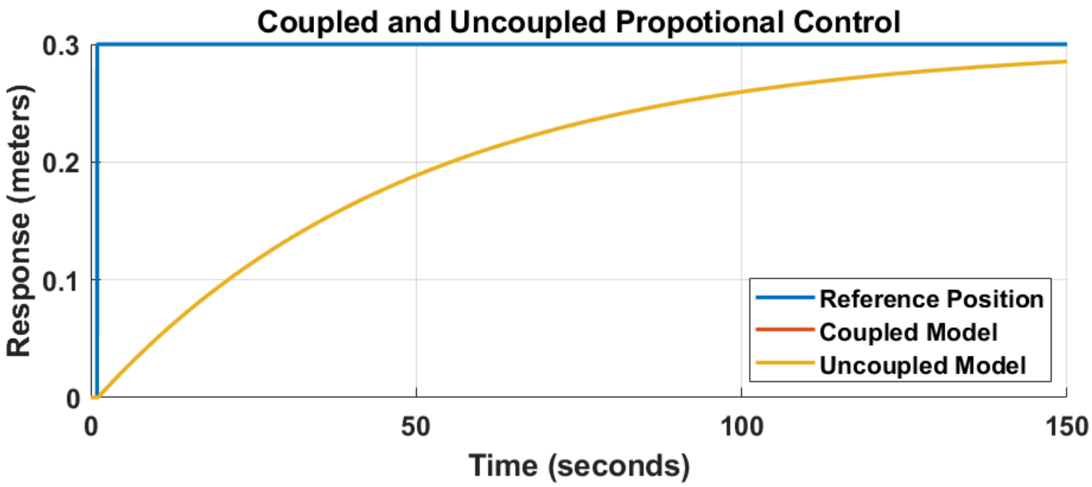
The Simulink model was developed after completing the proprtional controller. This model has a dual sine wave and step input that can be changed using a boolean variable. A value of 1 of the variable "step_bool" sets the demanded postion to a step input. A value of 0 sets the input to a sine wave input. After selecting the input, the input goes to both the coupled and decoupled models. This has a proportinoal feedback loop around the state-space block. For signals logged, the demanded position, the states, and the difference between the cart positions are logged.

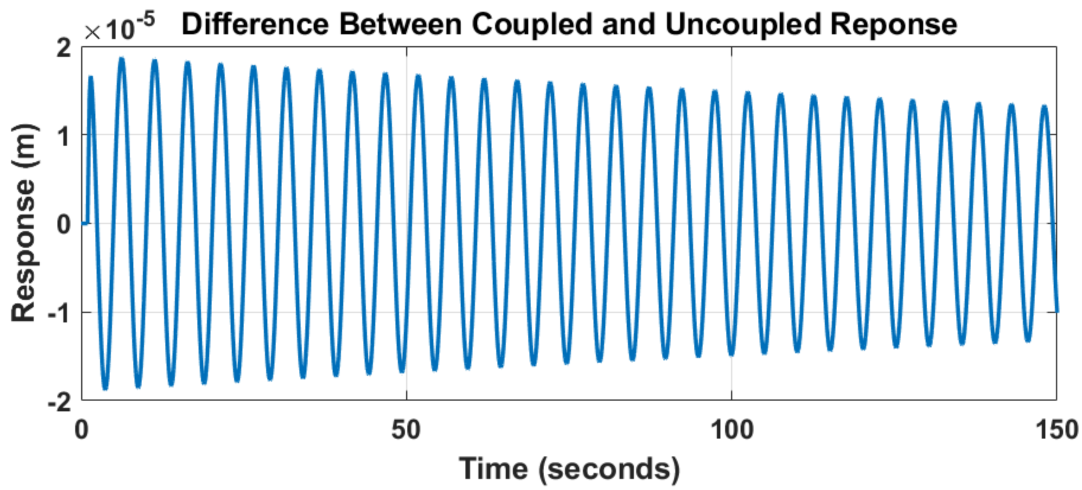



```
both_out = sim('simBoth');

figure()
subplot(2, 1, 1)
hold on
plot(both_out.logsout{4}.Values, 'LineWidth', 2)
plot(both_out.tout, both_out.logsout{1}.Values.Data(:, 1), 'LineWidth', 2)
plot(both_out.tout, both_out.logsout{2}.Values.Data(:, 1), 'LineWidth', 2)
xlabel('Time (seconds)')
ylabel('Response (meters)')
title('Coupled and Uncoupled Propotional Control')
legend({'Reference Position', 'Coupled Model', 'Uncoupled Model'}, 'Location', "southeast")
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
grid on
hold off

subplot(2, 1, 2)
plot(both_out.logsout{3}.Values, 'LineWidth', 2)
title('Difference Between Coupled and Uncoupled Reponse')
ylabel('Response (m)')
grid on
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
set(gcf, 'Position', [0, 0, 850, 700])
```




Looking at the above plots, the reponse to the step input has an extremly slow rise time, but the reponse is what I would expect it to look like.

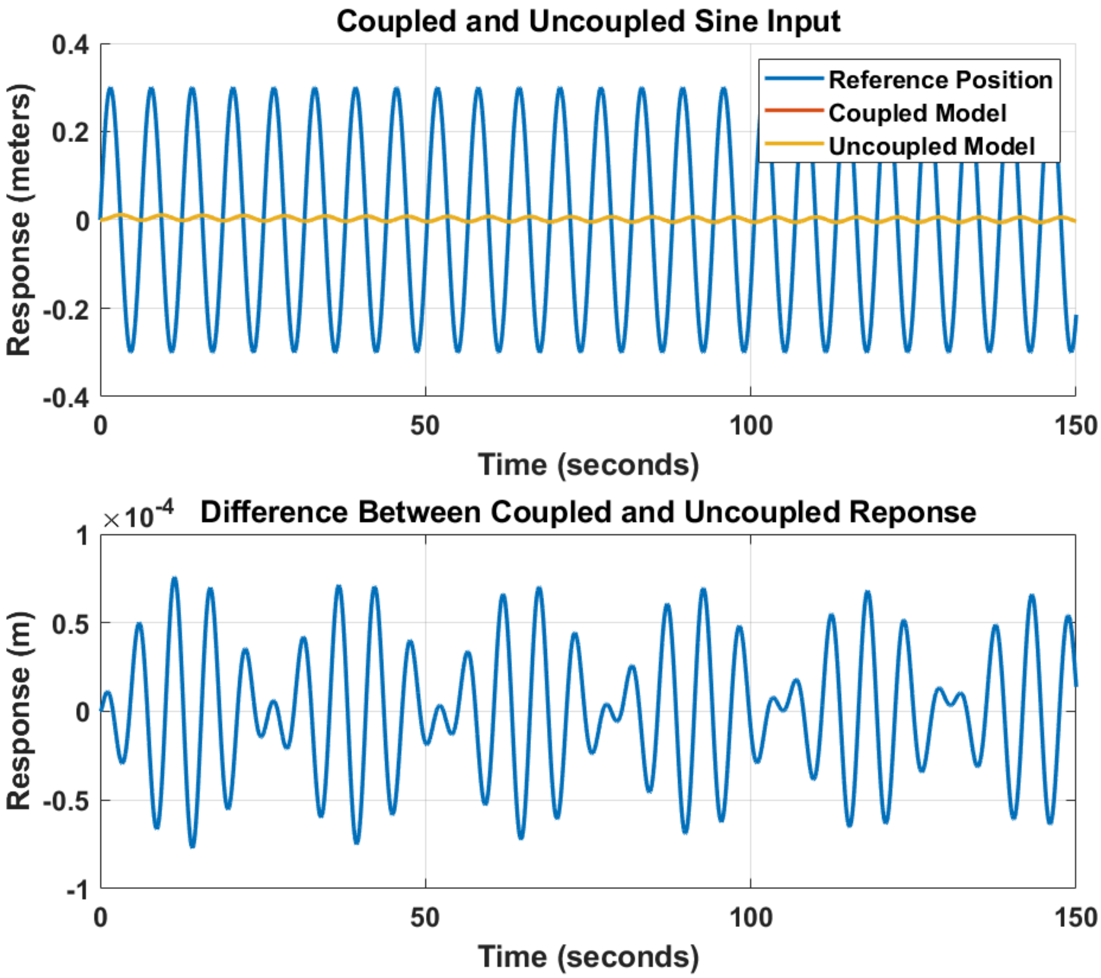
## Task 6: Sine Wave Input

```
sin_freq  = 1;
step_bool = 0;
sin_out = sim('simBoth');

figure()
subplot(2, 1, 1)
hold on
plot(sin_out.logsout{4}.Values, 'LineWidth', 2)
plot(sin_out.tout, sin_out.logsout{1}.Values.Data(:, 1), 'LineWidth', 2)
plot(sin_out.tout, sin_out.logsout{2}.Values.Data(:, 1), 'LineWidth', 2)
xlabel('Time (seconds)')
ylabel('Response (meters)')
title('Coupled and Uncoupled Sine Input')
legend('Reference Position', 'Coupled Model', 'Uncoupled Model')
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
```

```
grid on
hold off

subplot(2, 1, 2)
plot(sin_out.logsout{3}.Values, 'LineWidth', 2)
title('Difference Between Coupled and Uncoupled Reponse')
ylabel('Response (m)')
grid on
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
set(gcf, 'Position', [0, 0, 850, 700])
```



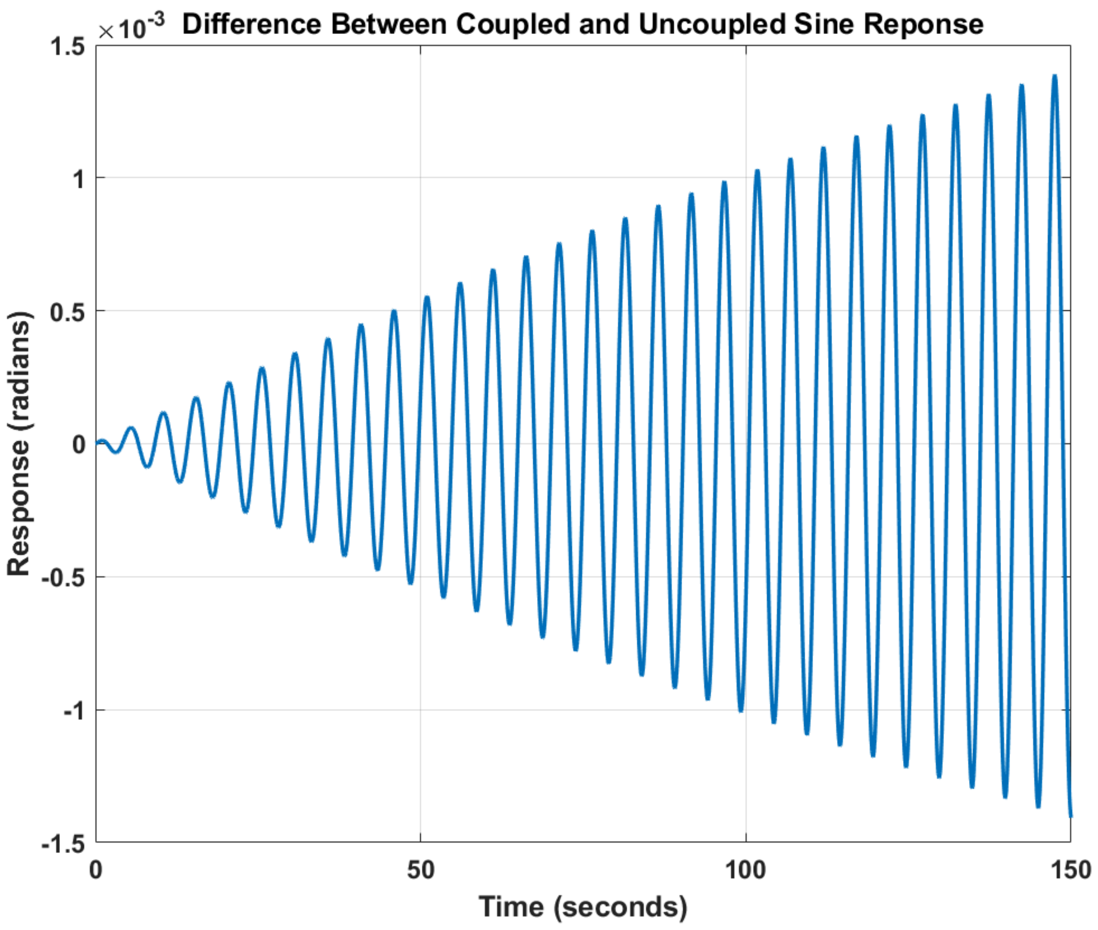

Looking at the above responses, the reponses between the coupled and uncoupled models are near identitcal. The lower plot shows the difference, which is on the order of micrometers. I do not consider this significant, and find a sensor that can accurately measure that distance wold be hard and expensive. This means that using the same calculated gain for the coupled and uncoupled models is a good choice, and that the rod most likely has a small overall effect on the cart's position. What is interesting is that the difference in frequency between the coupled and decoupled models is easily seen in difference plot, meaning the rod does have some small effect on the cart's frequency.

## Task 7: Rod Natrual Frequency Input

```
zero_cross = @(v) find(v(:).*circshift(v(:), [-1 0]) <= 0); % Finds indices where zero crossing

cross_i = zero_cross(true_out.logsout{1}.Values.Data(1001:end, 2)); % Indices where the rod cro
freq = 0.5 / mean(diff(true_out.tout(cross_i(1:end-1)))); % Frequency, 1 / mean of the periods
sin_freq  = freq * 2 * pi;
step_bool = 0;
nat_out = sim('simBoth');

figure()
plot(nat_out.logsout{3}.Values, 'LineWidth', 2)
title('Difference Between Coupled and Uncoupled Sine Reponse')
ylabel('Response (radians)')
grid on
set(gca, 'FontSize', 14)
set(gca, 'FontWeight', 'Bold')
set(gcf, 'Position', [0, 0, 900, 700])
```




First the natrual frequency of the rod was calculated. This is probably possible to do by inspection of the differential equations, but instead I looked at the step reponse of the "True Model" and found the frequency at which the rod rotated at by inspection. I then applied a sine wave position demand to the cart at that frequency with an amplitude of 30 centimeters. The rod on the uncoupled model barely moved, which makes sense as it is not coupled to the cart's postion or acceleration. The rod on the coupled one though does respond. As I would expect the reponse grows with time, as the input frequency is the natrual frequency of the rod.

## Conclusion

For this system, I would suggest to design it with a PID controller rather than just a propotional controller. The rod itself does not add much to the system's response as seen in the plots above, so even if PID isn't optimal it will most likely work well as the rod can just about be ignored. When driven at the rod's natrual frequency though the PID controller may not work well as they do not perform well with a sinusoidal changing error signal. It would also be of interest to model the cart and rod system with the non-linearities kept in to see what the error between the linear and non-linear "real" system is at high accelerations, or even just at low accelerations to see what the error is in the ranges displayed above.

The model displayed in this report suggests that the small angle apprximation, which accounts for most non-linearities, is a good one as the angle is kept in the milliradian range. I would be worried if the rod angle exceeded 20 degrees, as that is about when the small angle approximation goes above a 5% error margin that I would say would make it an unreasonable approximation.

```
rmpath Models\
rmpath scripts\
```