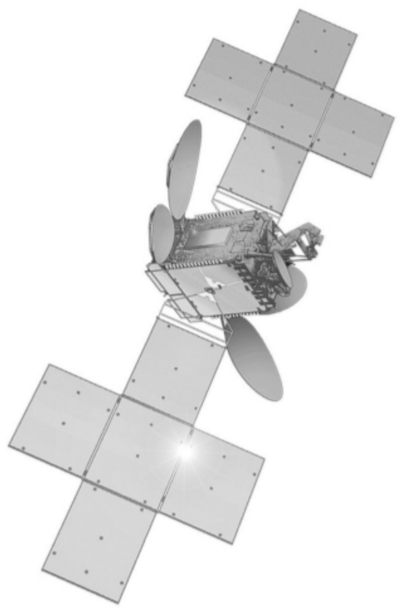
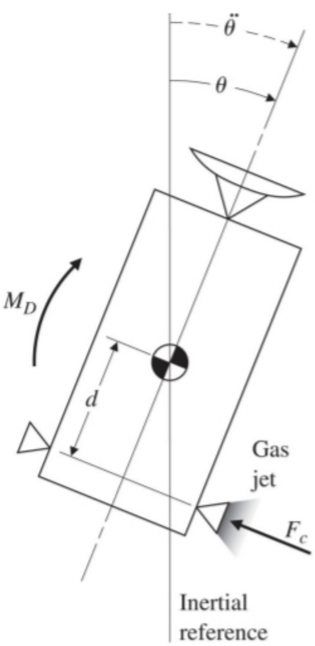
Analysis of Controller Performance for Spacecraft Control

By: Sai Aravind, Erid Pineda

April 26, 2023





Contents

List of Figures

List of Tables

1 Introduction

The goal of this project is to learn and apply numerous control methods for the control of spacecraft. This project will explore Proportional Control, Proportional-Derivative-Integral Control, Lead/Lag Compensation, and Linear Quadratic Regulator.

For the given plant our goal is to design and build different types of controllers. We plan to start with a simple proportional controller, and keep increasing the complexity of the controllers by designing a simple gain controller, compensator, PID controller, and finally an LQR controller. For the most part MATLAB and Simulink will be used to design, simulate and analyze the performance of the above mentioned controllers. At present, we do not have a particular specification for system design such as Rise Time, Steady State Error, or Overshoot percentage. Our goal is to design the best controller hence trade-offs between these specifications is obvious.

Table 1: Responsibilities

Member	**Controller Lead**
Erid Pineda	Proportional Controller Proportional-Integral-Controller
Sai Abhishek Aravind	Compensator Linear Quadratic Regulator

This report will specifically go over the design, analysis, and results of the P and PID controller.

2 Plant

The proposed plant for this was taken from Feedback Control of Dynamic Systems (FPE) 8th edition, Pg 738.

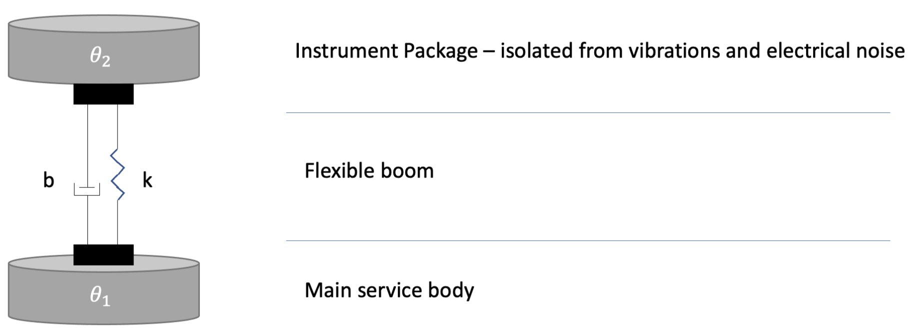


Figure 1: Satellite Two Body Model

Original plant:

$$G(s) = \frac{10bs + 10k}{s^2(s^2 + 11bs + 11k)} \tag{1}$$

Reduced order plant:

$$G(s) = \frac{10bs + 10k}{s^2 + 11bs + 11k} \tag{2}$$

The simplified plant (reduced order) of the satellite was instead Eq.2. The original plant with two poles at the origin will be used for each finished controller design for comparison shown in Eq.1.

The spring (k) and dampening (b) constants are also used from this example and found below. These constants were derived from physical analysis but vary as a result of temperature fluctuations but bounded by their ranges.

$$0.09 \leq k \leq 0.4 \tag{3}$$

$$0.038\sqrt{\frac{k}{10}} \leq b \leq 0.2\sqrt{\frac{k}{10}} \tag{4}$$

The values for spring and dampener will be kept constant and are found to be the average of the ranges above.

$$\begin{aligned}
k &= \frac{0.4 - 0.09}{2} \\
&= 0.1550 \\
b &= \frac{0.2\sqrt{\frac{0.1550}{10}} - 0.038\sqrt{\frac{0.1550}{10}}}{2} \\
&= 0.0225
\end{aligned}$$

The theory was then translated into MATLAB. The step response was looked over of the open loop plant for the original and reduced order plants.

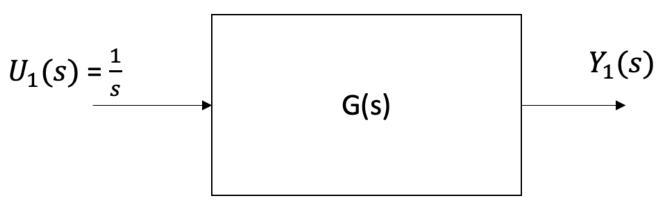
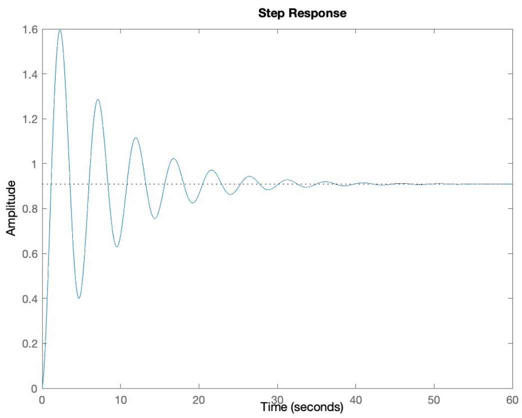
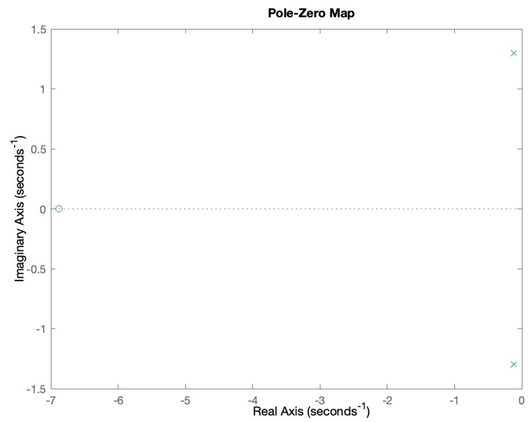


Figure 2: Step Inputs Gs

2.1 Reduced Order Plant Response



(a) Step Response of G(s)



(b) Poles and zeros of G(s)

The step response of the plant with no control is very oscillatilatory and takes more than 30s to reach the desired value. This behavior makes sense because of the two poles that are near the IM axes as shown in the figure above.

A ramp response was also simulated purely to observe the behavior.

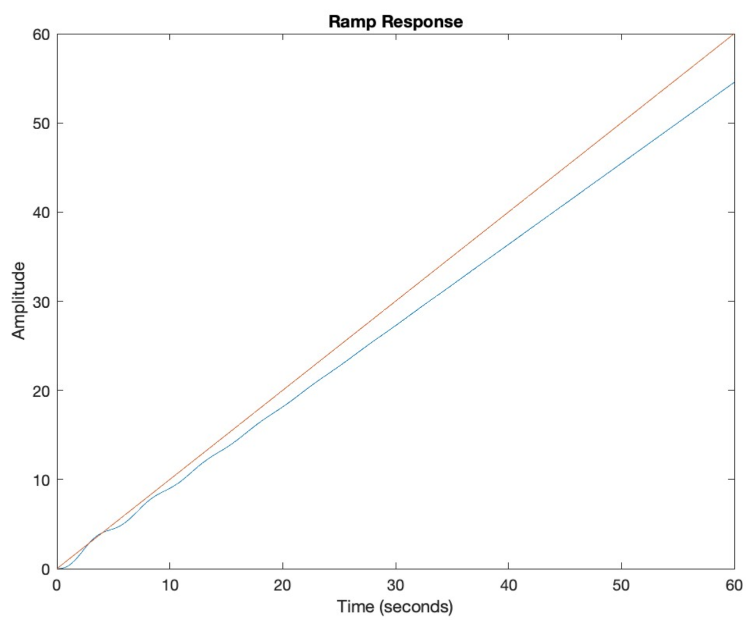
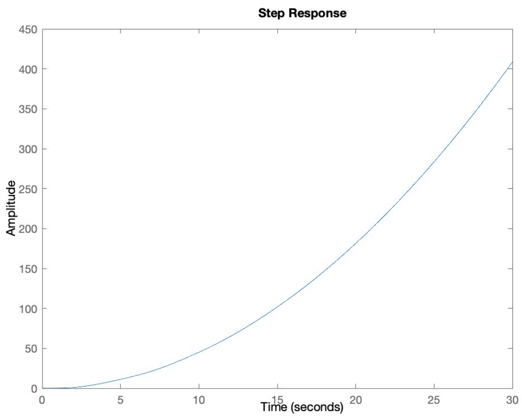


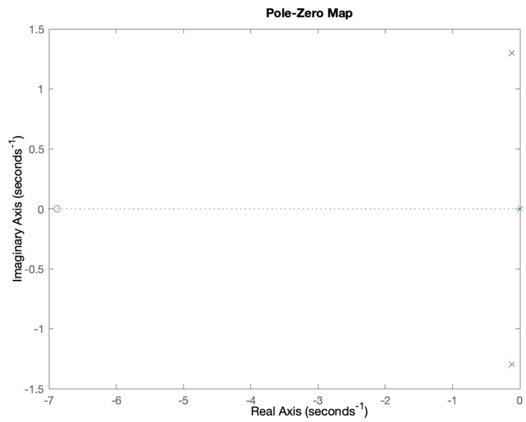
Figure 4: Ramp Response

The above behavior shows the error growing with time and so at time goes to infinity the error will follow.

2.2 Original Plant Response



(a) Step Response of original G(s)



(b) Poles and zeros of original G(s)

The original plant under no control is showing to be unstable with a step input. That will also mean that to a ramp and further order responses will also be unstable. This plant will require a controller in order to get a stable response.

2.3 Performance Specifications

Since our system controls a satellite, we should be careful about fast and sudden movements due to the flexible nature of the panels. If we make the system move fast than a specific limit, the system will oscillate due to flexible wings and reach a steady state after a long time. Overshoot is also an issue due to the same reason. A satellite system also has a lot of delicate parts so we cannot afford to damage certain features just for a faster response.

The specifications are found in the table below. As we design different controllers, our goal will be to use the previous control law's specifications and make it better (faster and more stable).

Table 2: Performance Specifications

Parameters	**Specification**
Overshoot (%)	10
Settling Time (s)	20s
Rise Time (s)	N/A
Steady State Error (step)	0

3 Proportional Controller

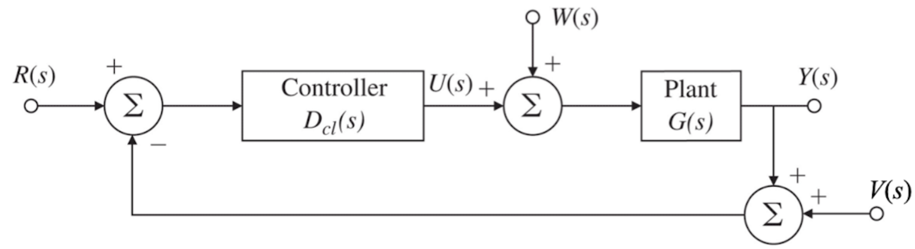


Figure 6: Block Diagram from Figure 4.2 in FPE 8^{th} edition

A unity feedback control system is found above where $D_{cl}(s)$ is equal to:

$$D_{cl}(s) = K_p \tag{5}$$

Where the closed loop transfer from R(s) to Y(s) is equal to:

$$G_{cl}(s) = \frac{D_{cl}(s)G(s)}{1 + D_{cl}(s)G(s)} \tag{6}$$

The closed loop transfer function when substituting $D_{cl}(s)$ and $G(s)$ we get

$$G_{cl}(s) = \frac{10K_pbs + 10kK_p}{s^2 + s(11b + 10K_pb) + k(10K_p + 11)} \tag{7}$$

The steady state error was found using MATLAB and verified using the final limit theorem. The error of this unity feedback system is found to be:

$$E(s) = R(s) - Y(s) \tag{8}$$

Substituting Y(s) yields:

$$\begin{aligned} E(s) &= R(S)(1 - Y(s)) \\ &= R(S)(\frac{s^2 + 11kbs + 11k}{s^2 + s(11b + 10K_pb) + k(10K_p + 11)}) \end{aligned}$$

At this point we are now able to use final limit theorem to find the steady state error for a given response, in this case starting with a step response.

$$e_{ss} = \lim_{s \to 0} sE(s) = s\frac{1}{s}(\frac{s^2 + 11kbs + 11k}{s^2 + s(11b + 10K_pb) + k(10K_p + 11)}) \tag{9}$$

$$e_{ss} = \frac{11k}{k(10K_p + 11)} \tag{10}$$

So as K_p increase the error decreases.The gain was adjusted between 1 and 10 and this behavior was observed in the plot below.

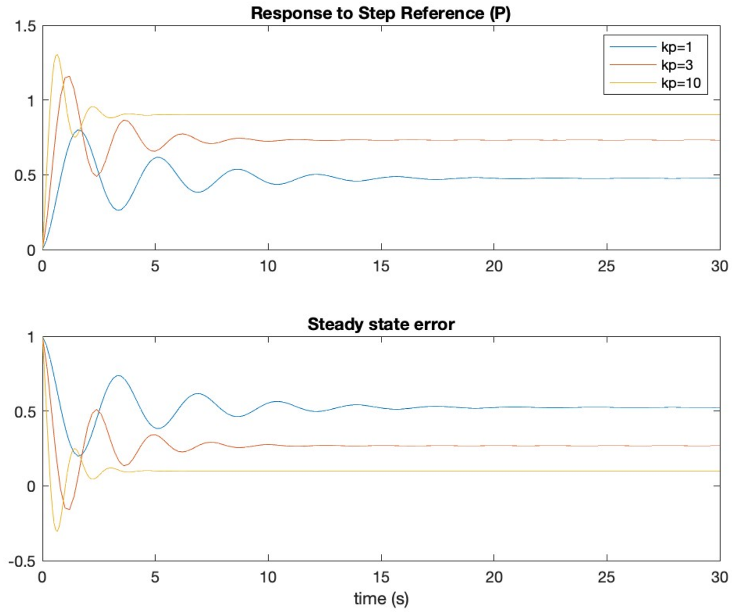


Figure 7: Step Response & Steady State Error with multiple gains.

It was also observed that when increasing the gain, the OS also increased. The target for this controller was use a gain such that the OS was no bigger than 10% OS. So the gain of $K_p = 2$ was used to satisfy that requirement.

Table 3: Proportional Control - System characteristics

Gains(K_p)	**OS(%)**	**Error**
1	N/A	0.52
3	15.9	0.27
10	30.4	0.01

A steady state error of 0 can not be achieved using proportional controller without sacrificing overshoot. In the final design of this proportional controller the final gain was chosen with overshoot being the more important parameter. In order to achieve the zero steady state requirement set early in the report, a pole at the origin would needed to be added.

Table 4: Proportional Control - Final System characteristics

Gains(K_p)	**OS(%)**	**Error**
2.3	9.54	0.32

3.0.1 P-Controller Output

After a controller is designed the next is to optimize and/or try a more advance controller. Then once the needs are satisfied from the simulations ran evaluate its on a prototype and therefore on actual hardware. During the simulation stage of the design process it is very important to look at the output of the controller before the plant as seen in the figure below. This needs to be done to ensure that motor saturation does not take place during prototype testing because ultimately it could lead to broken motor and/or unable to control the given system if not sized properly.

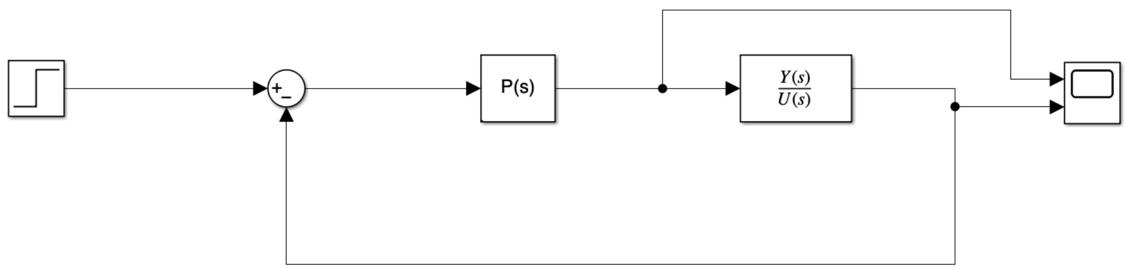


Figure 8: SIMULINK Simulation - Controller Output Signal

For the application of this satellite control system it was found that typically for a system such as this one, 10V is the order what is used nominally. This means that for the controllers designed must not go above this number.

For this proportional controller, its output was obtained using SIMULINK and its output was found below.

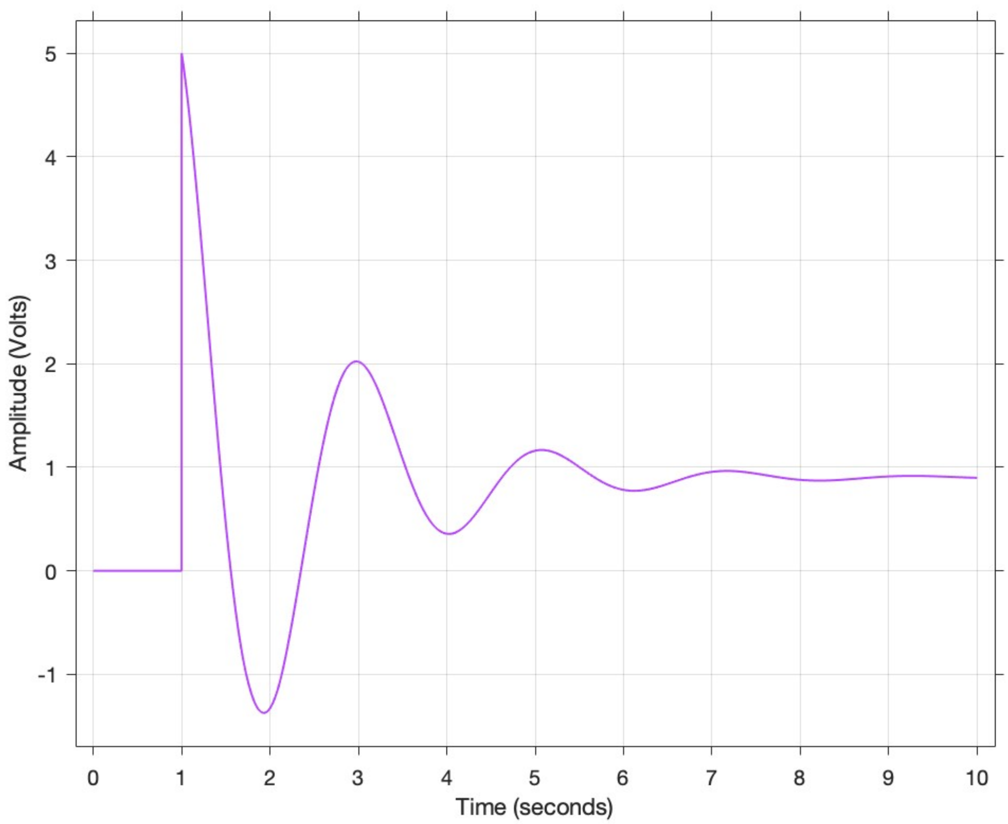
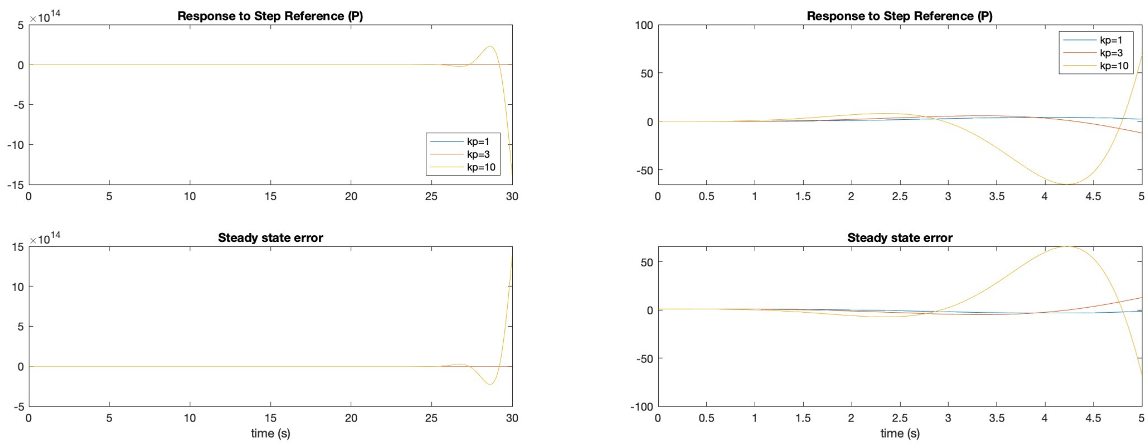


Figure 9: P-Controller Output Signal

Given that this controller did not reach near the intended saturation limit of 10V this controller would be safe to implement on hardware.

3.1 Original Plant with P-Control

The original plant G(s) will now be compared with the proportional controller designed earlier and the differences will be observed.



(a) Step Response Unbounded **(b)** Step Response first 5 seconds

Figure 10: Step Response & Steady State Error with multiple gains of original plant.

The step response with proportional control did not stabilize the original plant as shown above. This behavior makes some sense and is expected. The original plant was unstable already unstable as shown in section 2.2. In order to stabilize this system two integrators would needed to be added to increase the system type of this plant and therefore reducing the error.

4 Promotional-Integral-Derivative Controller

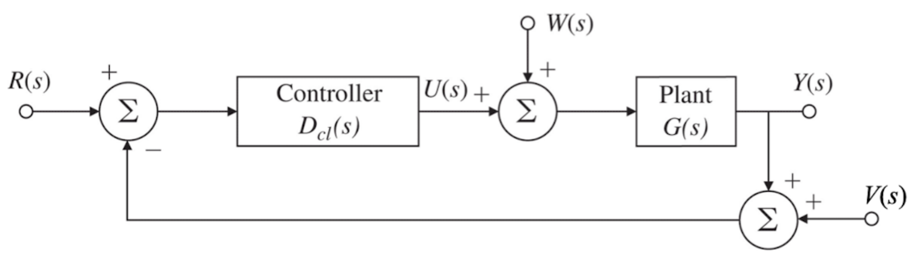


Figure 11: Block Diagram from Figure 4.2 in FPE 8^{th} edition

For the this system $D_c(s)$ is found below.

$$D_c(s) = K_P + \frac{K_I}{s} + K_D s \tag{11}$$

With the implimentation of this controller the closed loop transfer function from R(s) to Y(s) is found to be the equation below.

$$G_{cl}(s) = \frac{10K_d b s^3 + (10K_p b + +10K_d k)s^2 + (10K_p k + 10K_I b)s + 10K_I k}{s^3(1 + 10K_d b) + s^2(11b + 10K_p b + 10K_d k) + s(11k + 10K_p k + 10K_I b) + 10K_I k} \tag{12}$$

From performance specifications found in Table 2 the desired performance region can be found.

For the desired OS, a dampening coeff (ζ) can be found.

$$\zeta \geq \frac{ln(M_p)}{\sqrt{\pi^2 + (ln(M_p))^2}} \tag{13}$$

Where M_p is the minimum OS.

With this dampening value, θ_{min} can also be found to further clarifying our performance region.

$$\theta_{min} = sin^{-1}(\zeta) \tag{14}$$

$$\begin{aligned}\theta_{min} &= sin^{-1}(0.59)\\ &= 36°\end{aligned}$$

Given that the system needs to have a settling time of less than 20sec. A desired ω can be found.

$$t_s = \frac{4.6}{\zeta\omega_n} \tag{15}$$

$$\omega_n \geq \frac{4.6}{\zeta t_s} \tag{16}$$

Table 5: Performance Region Values

Parameters	Specification	Desired Values
Overshoot (%)	10	$\zeta \geq 0.59$
Settling Time (s)	20s	$\omega_n \geq 0.39$
Rise Time (s)	N/A	N/A
Steady State Error (step)	0	N/A

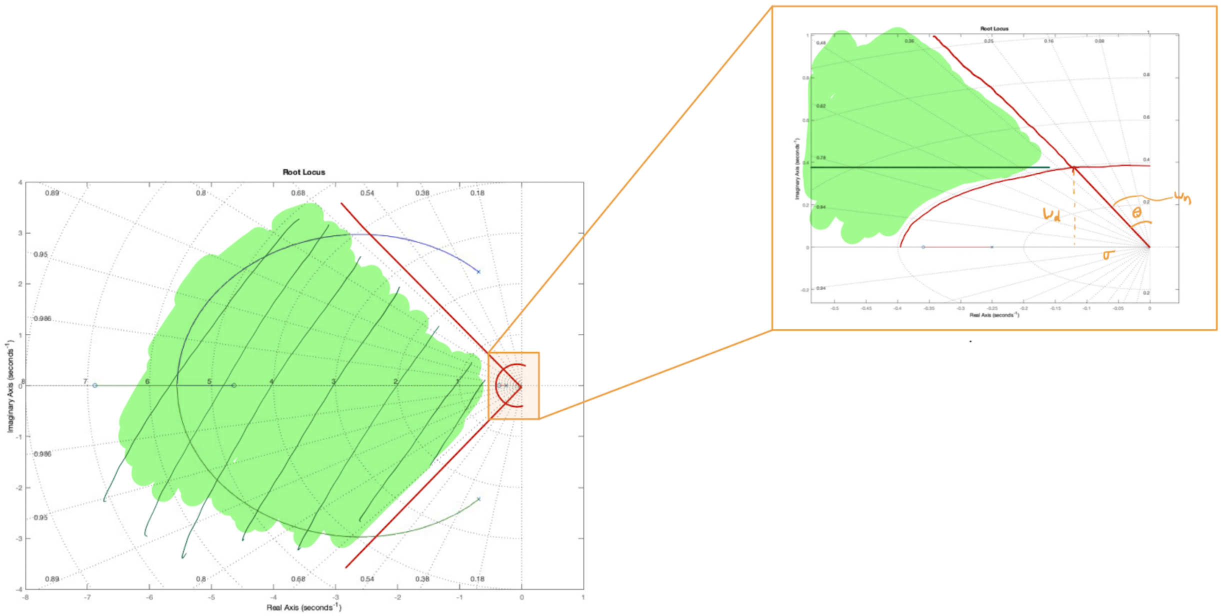


Figure 12: Root Locus with the performance region

This PID controller contains 1 integrator and therefore will increase the system type of this closed loop transfer function. As a result the error to the step response will be observed to be zero.

$$
\begin{aligned}
E(s) &= R(s) - Y(s) \\
&= R(s)\left[\frac{s^3 + 11bs^2 + 11ks}{s^3(1+10K_d b) + s^2(11b + 10K_p b + 10K_d k) + s(11k + 10K_p k + 10K_I b) + 10K_I k}\right] \\
&= \frac{1}{s}\left[\frac{s^3 + 11bs^2 + 11ks}{s^3(1+10K_d b) + s^2(11b + 10K_p b + 10K_d k) + s(11k + 10K_p k + 10K_I b) + 10K_I k}\right]
\end{aligned}
$$

Then using Final Value Theorem,

$$e_{ss} = \lim_{s\to 0} sE(s) = s\frac{1}{s}\left[\frac{s^3 + 11bs^2 + 11ks}{s^3(1+10K_d b) + s^2(11b + 10K_p b + 10K_d k) + s(11k + 10K_p k + 10K_I b) + 10K_I k}\right] \tag{17}$$

$$e_{ss} = \frac{0}{10K_I k} = 0 \tag{18}$$

The controller was first designed with the following constants.

Table 6: Initial PID Constants

Controller Constants	**Value**
K_P	2.3
K_I	1
K_D	0.6

The constant for the proportional gain (K_P) from the P-Controller was carried over and the constant for the derivative was found to be 0.6.

To complete this PID controller, the integrator constant can be found iteratively by checking the OS for each iteration.

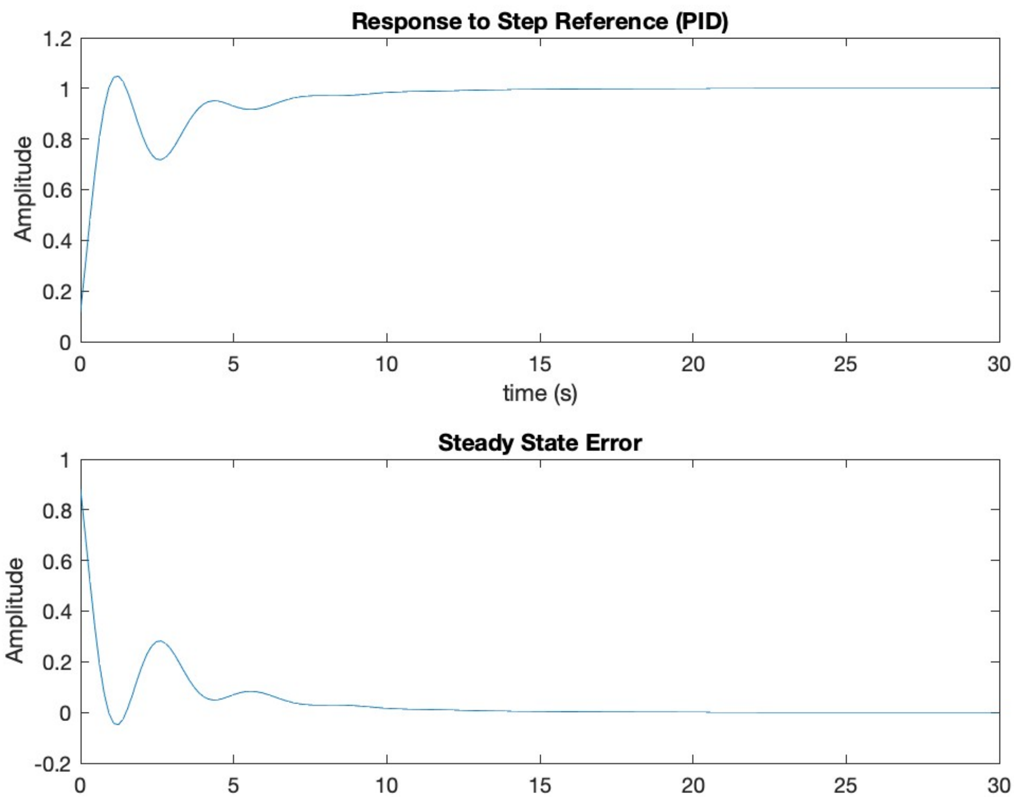


Figure 13: Initial Controller Constants

With the first set of values found in Table 6 the OS was found to be 4.6% satisfying the requirements set out for each controller design.

4.1 Full PID Performance

This controller can be further pushed and provide better performance by first looking at the ramp response of this closed loop transfer function. Ramp functions could be used to tracking an object if its slow enough. Given that this ramp function could be useful for that application reducing this error would next step to get its full performance.

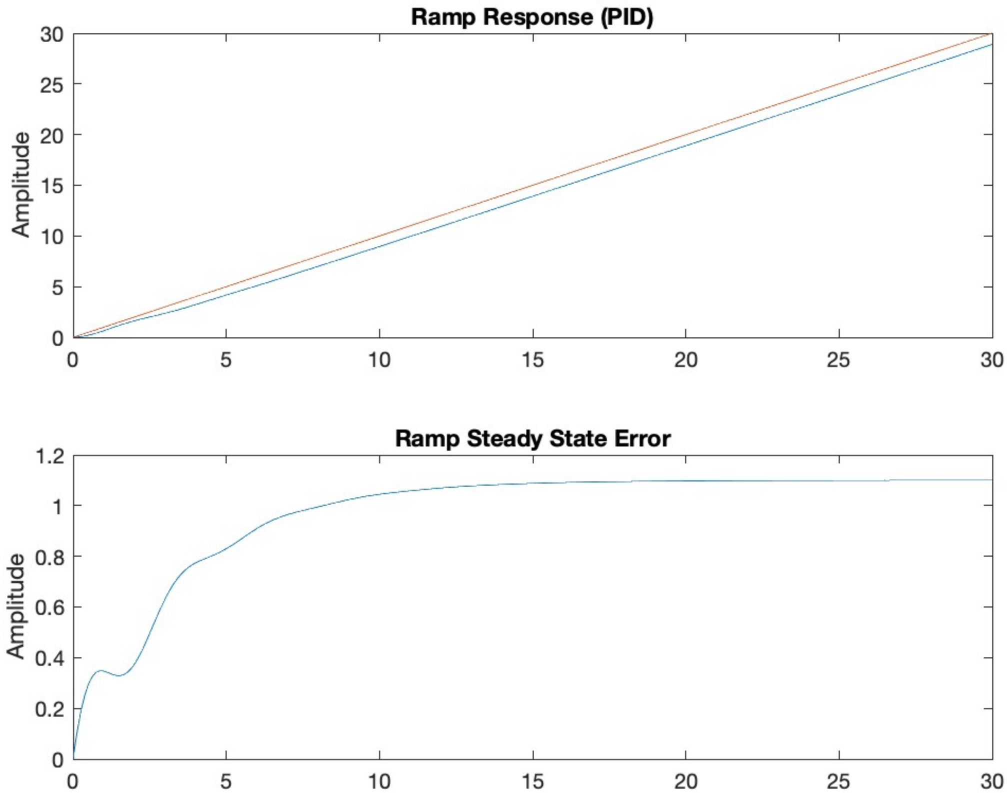


Figure 14: Initial Controller Constants

From the figure this closed loop transfer function the error can be observed and is found to be approximately 1. To find the lowest possible error to this ramp response, the dampening value must be set to its max from 0.6 to 0.99. Then making small increments of K_I. This is shown by deriving the error from R(s) to Y(s).

$$\begin{aligned}
E(s) &= R(s) - Y(s) \\
&= R(s)[\frac{s^3 + 11bs^2 + 11ks}{s^3(1 + 10K_db) + s^2(11b + 10K_pb + 10K_dk) + s(11k + 10K_pk + 10K_Ib) + 10K_Ik}] \\
&= \frac{1}{s^2}[\frac{s^3 + 11bs^2 + 11ks}{s^3(1 + 10K_db) + s^2(11b + 10K_pb + 10K_dk) + s(11k + 10K_pk + 10K_Ib) + 10K_Ik}]
\end{aligned}$$

Then using Final Value Theorem, the steady state error can be obtained once again.

$$e_{ss} = \lim_{s \to 0} sE(s) = s\frac{1}{s^2}[\frac{s^3 + 11bs^2 + 11ks}{s^3(1 + 10K_db) + s^2(11b + 10K_pb + 10K_dk) + s(11k + 10K_pk + 10K_Ib) + 10K_Ik}] \tag{19}$$

$$e_{ss} = \frac{11k}{10K_Ik} \tag{20}$$

From the equation above it is shown that increasing K_I further reduces the error of this ramp response.

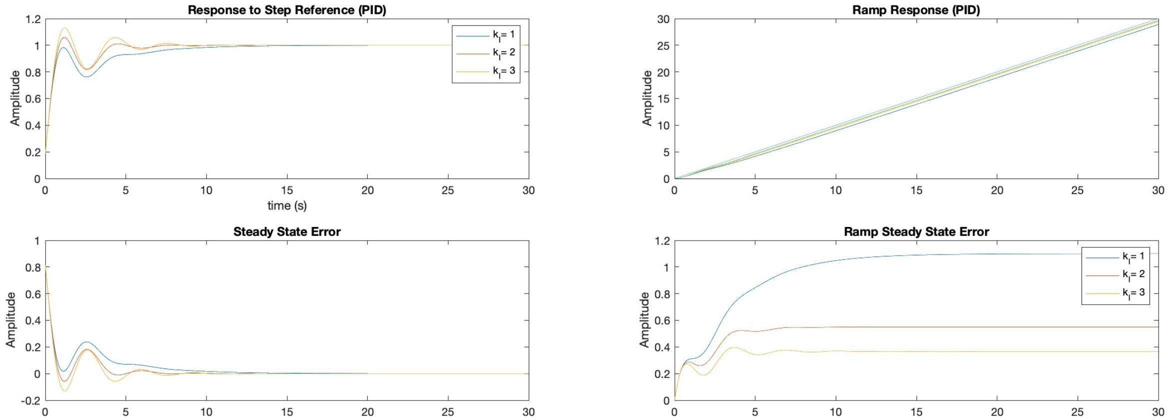


Figure 15: Highly Damped Step/Ramp Responses with ranging values of K_I

By increasing K_I the error was reduced greatly but by increasing constant to the Integrator in this controller the OS was also affected. The resulting values are found in the table below.

Table 7: Integrator Constant vs. Error & OS

K_I	e_{ss}	**OS(%)**
1	1.09	-0.003
2	0.55	5.881
3	0.37	12.954

From the table it was easily seen that the integrator constant K_I needed to be between 2 and 3. The final value was found to be 2.5 and resulted in the final response for both step and ramp with the minimized min value of 0.44.

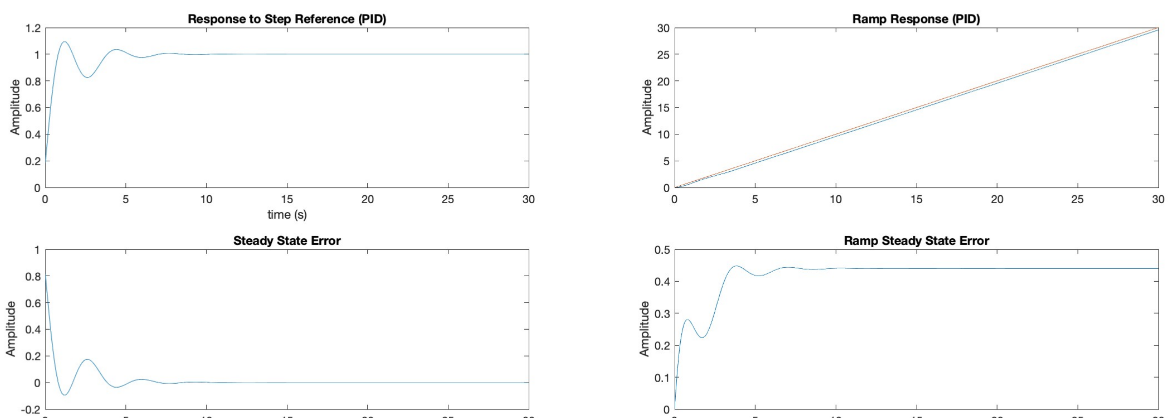


Figure 16: Final PID Controller Responses

Table 8: Final PID Values

Controller Constants	Value
K_P	2.3
K_I	2.5
K_D	0.99

4.1.1 PID Controller Output

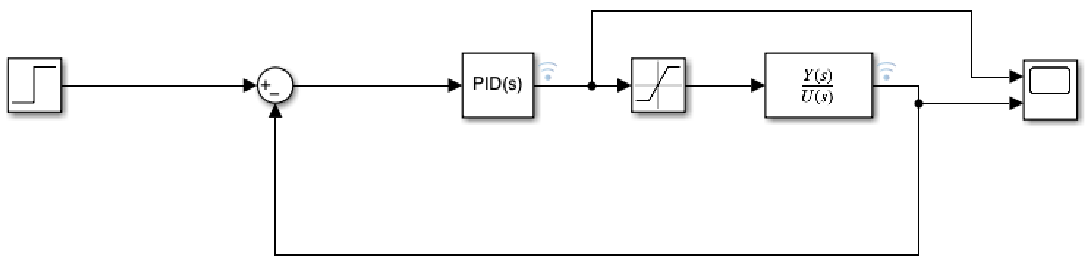


Figure 17: SIMULINK Simulation - PID Controller Output Signal

For the PID controller designed the following output signal was found below.

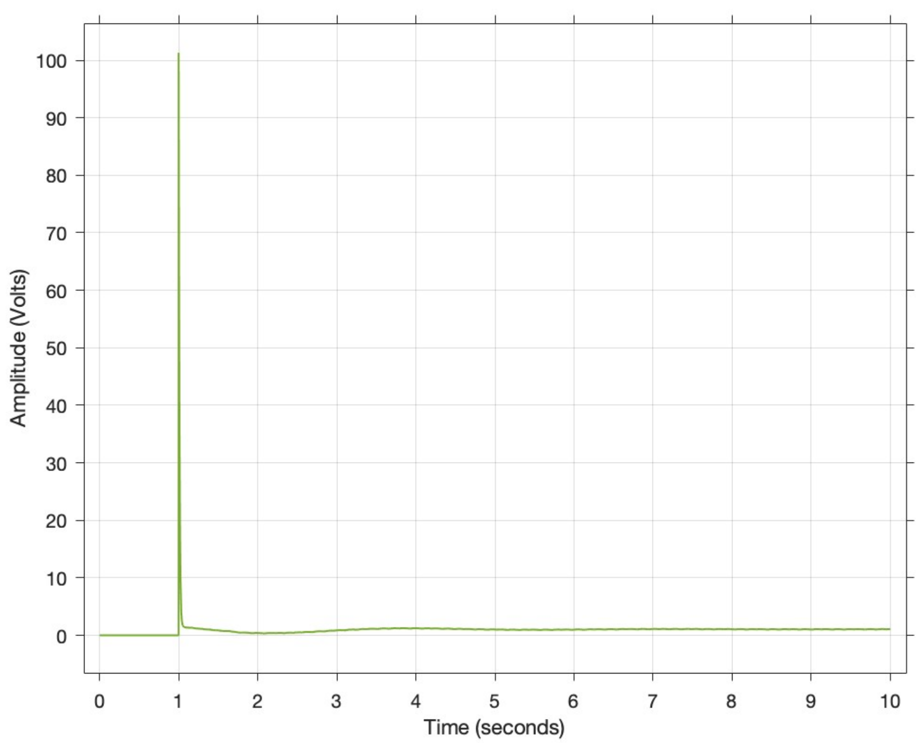
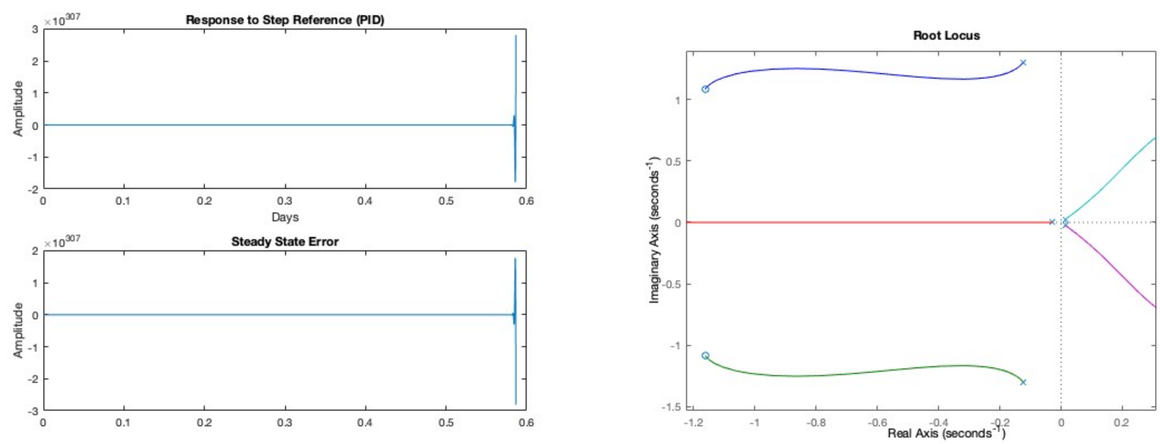


Figure 18: PID Controller Output Signal

This PID controller was found to go beyond the designated limit of 10V by a factor of 10 and will therefore saturate the motor for a small amount of time. This controller looks to be a good example of integrator windup and would benefit from an anti-windup scheme.

4.2 Original Plant with PID Control

The original plant higher order plant with the PID controller designed was initially was also expected to provide a stable system but that was not that case given by the step response below.

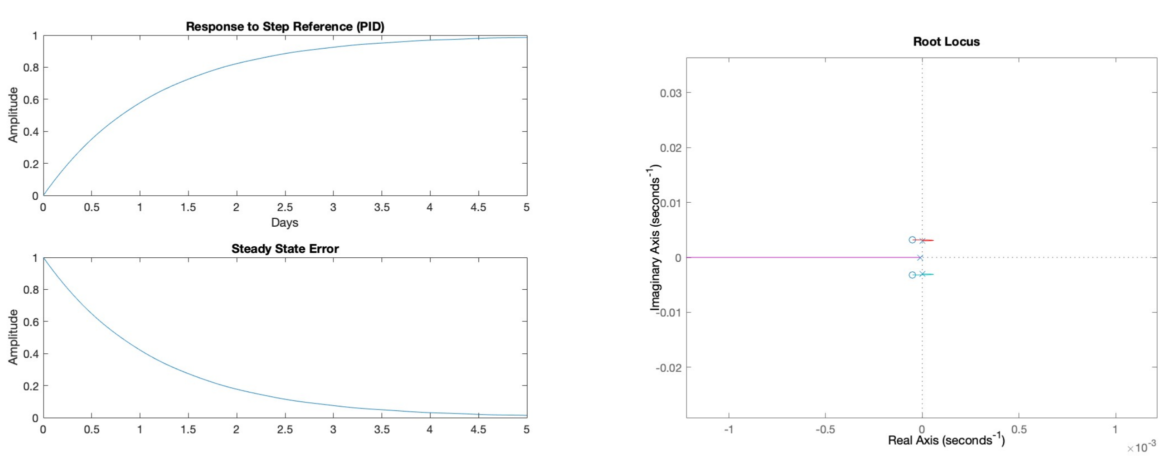


(a) Step Response **(b)** Root Locus

Figure 19: Original higher order plant with PID control

The Root Locus found in figure 19b shows two poles on the right hand plane. This finding meant that the gain chosen could be to large driving those two poles to the right hand plane.

This unstable system was found to be unstable because of the resonance and so the gain was lowered until stability was reached. The steady state value was reached in this case was in the order of days instead of seconds.



(a) Step Response **(b)** Root Locus

Figure 20: Original higher order plant with PID control - lowered gains

5 Discussion

When going through the design process for these two controllers the theory fell inline with the simulation results as seen in the design of the P-Controller and PID controller. The equations below governed how the simulations behaved and allowed the controller to satisfy the specified requirements.

1. $t_s = \frac{4.6}{\zeta\omega_n}$
2. $\zeta \geq \frac{ln(M_p)}{\sqrt{\pi^2+(ln(M_p))^2}}$
3. $\theta_{min} = sin^{-1}(\zeta)$
4. $e_{ss} = \lim_{s\to 0} sE(s)$

Using the first three equations above allowed to meet the steady state requires set earlier in the design phase (overshoot, rise time, settling time, etc).

What was a bit more confusing following this design process was seeing the effects that final controller used on the reduced order plant had on the original higher order plant. I initially believed that PID controller would also stabilize the higher order plant but that was not the case. A mixture of frequency response modeling would have illuminated some of the design decisions made in section 4.2.

The final results of the controllers deigned in this project is also found below.

Table 9: Controller Comparison

Specification	**P-Control**	**PID-Control**	**Compensator**	**LQR**
$OS \leq 10\%$	9.5%	4.6%	2%	5.7%
$T_s \leq 20s$	9.87s	6.37s	18s	1.45s
$T_r = N/A$	0.49s	0.63s	1.8s	0.50s
$e_{ss} = 0$	0.32	0	0	0

What me and my partner expected to see was that each controller designed from left to right on the table above would be better. In the case of the PID and compensator controllers it was not the case with the settling time of the compensator near the limit of the specification. But overall the LQR controller was found to be the best at almost all parameters specified on the table. With these more advanced controllers there seemly is less control over the design given that there are functions embedded that only allow for optimal parameters, in this case with the costing function associated with Sai's LQR controller. There are less iterations made through out the design process and ultimately end up with the superior controller in the end like seen in the table above.

6 Conclusion

Through the course of this project some valuable lessons were learned and were more so illuminated throughout this course. The design process was more so instilled as each controller was designed an evaluated.

- (4) Construct a linear model
- (5) Try Simple PID or Lead-Lag design
- (6) Evaluate/modify plant
- (7) Try an optimal design (advanced controller)
- (8) Simulate performance including nonlinearities, actuator saturation, etc

During the course of this project steps 4, 5, and 8 from the general design guidelines were seen in this report. Step 8 was one that I found to be the most important and challenging. This step is barrier before implementing on hardware. The challenge is in both the controller design and the plant. If the plant is not close to characterizing the physical model the controller designed may not work as intended. Also if the controller designed isn't able to fulfill the requirements more advanced controllers techniques or filters may be needed. That in essence is what I can see being done in industry. By first starting with basic linear models and controller and as the grow in complexity through the design life cycle of a project. This experience gave me a glimpse of what can happen when applied to any project.