Robotic Arm Position Control Project

Aya Sabe Ayoun
Mechanical Engineering Student
American University of Beirut
Beirut, Lebanon
aws08@mail.aub.edu

Jad Bhamdouni
Mechanical Engineering Student
American University of Beirut
Beirut, Lebanon
jnb21@mail.aub.edu

Jihad Jundi
Mechanical Engineering Student
American University of Beirut
Beirut, Lebanon
jaj12@mail.aub.edu

***Abstract*—In this project, different controllers and compensators are studied for controlling the position of a one-DOF robotic arm by targeting certain pre-determined performance requirements. The designs include proprtional, PD, and PID controllers as well as two lead compensators. Time-domain and frequency-domain analysis of simulations to different inputs are implemented, and the system's ability to reject disturbance is also studied. After evaluating the response of all the designs, one of the two lead compensators is chosen as the recommended one. An incomplete physical model of the closed-loop system is built to validate simulation results.**

I. Literature Review

The DC motor has been one of most widely used motors due to its speed control qualities, even though its maintenance is more costly than the induction motor. Consequently, research into position control has been widely studied over the past few decades. The main controllers that have been used to control the position and speed of the DC motor are Proportional-Integral Derivative Controllers.

In a study by Thomas et al., the DC motor was a "shunt motor", where the coil is in parallel with the voltage source instead of in series; this results in the current across the coil and across the armature to be different. This changes the model's equations and consequently the transfer function, but it remains a type 1 3rd order system. Thomas et al. attempted to design the position controller as a PID using Ziegler-Nichols Method and proceeded to tune it using what they call "Genetic Algorithm Approach(GA)". GA is an efficient optimization technique used to solve problems. Based on the results obtained, the GA tuned Position PID Controller has much better performance characteristics than a PID tuned using a classical method validated by simulation on MATLAB [1].

Maung et al. used a 12 V Namiki DC motor with a metal gearbox and an integrated quadrature encoder to measure the motor's speed and position for the output shaft [2]. It should be mentioned that the torque generated at the output shaft of a DC motor can be scaled with the use of gear trains [3].Hence, high torque position and low speed can be obtained with the use of a geared DC motor providing a feedback loop. Bi-directional motor control is possible to obtain with the use of an L298 H-Bridge module with pulse-width modulation (PWM) from an Arduino micro-controller. It should noted that L298 H-Bridge module can only be utilized for motors of a voltage between 5 and 35V DC with a peak current up to 2A.

Maung et al. introduced Coulomb's friction force on the driving velocity for the model used to design their controller, which resulted in better performance characteristics of the desired position validated through experimental results [2].

For Akbar et al., PID controllers of a DC servo motor are designed using automated PID tuning by sisotool for higher order systems using Simulink, to control a robot arm manipulator model. They also used fuzzy logic control (FLC) to control the movement of the robot arm and ensure its smooth precise movement [4].

After performing simulations, they concluded that FLC showed better performance in terms of stability, precision, overshoot, and response speed. The problem with the PID controller is the high overshoot. FLC eliminated these undesired oscillations and provided a smooth transient response [4].

Zhang et al. designed active and energy regenerative controllers for DC motor base suspension specifically in cars, where they can operate in two ways: active control that assures comfort during the ride and the energy regenerative control that takes advantage of the vibrations considering them an energy source. To achieve this, they used a torque tracking loop control consisting of DC motor, a three-phase inverter, a rotor position detector, current sensors, and a DC power supply. A rotary DC motor was used as an actuator and applied the rack and pinion mechanism to convert the rotary motion of the DC motor into linear motion of the suspensions, with the addition of an electromagnetic damper. A hysteresis current controller was used to control the current by comparing the current error with the hysteresis bandwidth. For the main loop controller design, they used an H infinity control strategy to achieve stabilization with the guaranteed performance. It consisted of two controllers, an H infinity robust for active suspension based on full car suspension model and a restricted H infinity which restricts the motor to work in certain areas of energy regeneration [5].

Namazov et al. designed a fuzzy control system to control the position of a DC motor. A proportional derivative fuzzy controller (FPD) and a PID controller were used. The PID controller was tuned using hand tuning or Ziegler Nicholas, where they concluded that the Ziegler Nicholas method gives poor results for systems with very high time lag and that hand tuning would take a long time to find the values desired. A PD controller was also designed and tuned using Signal Constraint block of Simulink Response Optimization Toolbox which was used to determine the parameter values of a linear FPD [6].

Defuzzification was applied to the controller to obtain a certain output value, where SOM defuzzification gave the best results. Upon introducing the system to a disturbance, the FPD controller rejected it without the need to tune it, unlike the PD controller that wasn't able to do so [6].

Previous work has been done on controlling a one-degree robotic arm, where the employed DC motor was mathematically modeled and its transfer function relating angular position and voltage was derived. A simple proportional controller was able to meet the desired requirements [7], but it is worthy to note that these requirements were quite loose, particularly when compared to the ones imposed in our study.

Other researchers have implemented different approaches to obtain satisfying results for controlling the angular position of a DC motor, including state feedback pole placement, Fuzzy Logic, Multi-Layer Perceptron (MLP) Neural Network, and the conventional PID [8], where very similar output responses were obtained across all controllers and with the main differences residing in the transient phase. It was found that PID's resulted in the fastest response but at the expense of a small overshoot. One drawback of this type of controller is the need of the mathematical transfer function of the system, which is not always available with a satisfying degree of accuracy, as opposed to Fuzzy logic controllers which only rely on prior knowledge about the system.

Chowdhury et al. adopted four controllers, including phase-lag, phase-lead, PI and PID, achieving a phase margin of 40 degrees, and were digitally implemented in MATLAB. Assessment of their respective performances was conducted using Bode plots for the compensated open-loop system and by studying the step response characteristics of the closed-loop system. It was found that all controllers resulted in zero steady state error, with the phase-lead and phase-lag resulting also in zero overshoot, and the PID and phase-lead controllers showing the lowest rise time and lowest settling time respectively. [9]

II. Dynamic Model

Assumptions: $K_m = K_b = K$

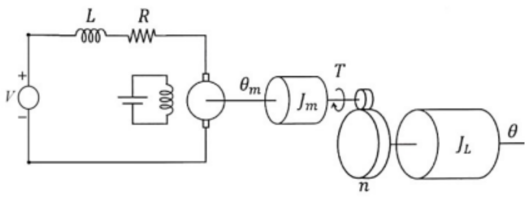


Fig. 1: Free Body Diagram & Circuit

Kirchhoff's Law was applied on the circuit and Moment Conservation on the Motor:

$$V = L\frac{di}{dt} + Ri + K\dot{\theta}_m \tag{1}$$

$$\tau_m = J_m\ddot{\theta}_m + b\dot{\theta}_m \tag{2}$$

$$\tau_L = J_L\ddot{\theta} \tag{3}$$

Due to the gears between the motor and the load:

$$\tau_{total} = n\tau_L + \tau_m = Ki \tag{4}$$

$$\frac{1}{n}(J_m\ddot{\theta} + b\dot{\theta}) = \tau_m \tag{5}$$

$$n(J_L\ddot{\theta}) = n\tau_L \tag{6}$$

A. State Space Representation

The state variables are defined to be:

$$X = \begin{bmatrix} \ddot{\theta} \\ \dot{\theta} \\ \theta \end{bmatrix} \tag{7}$$

and output variable to be:

$$Y = \begin{bmatrix} \theta \end{bmatrix} \tag{8}$$

From Equations (1), (2), and (3), obtain:

$$\dddot{\theta} = -162.4945\ddot{\theta} - 430.2938\dot{\theta} + 1.633V \tag{9}$$

Leading to:

$$\dot{X} = \begin{bmatrix} -162.4945 & -430.2938 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} X + \begin{bmatrix} 1.633 \\ 0 \\ 0 \end{bmatrix} V \tag{10}$$

$$Y = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix} X + \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} V \tag{11}$$

B. Transfer Function

(5) and (6) are replaced in (3) and obtain after simplification: For position output:

$$G(s) = \frac{\theta}{V} = \frac{Kn}{(LJ_o)s^3 + (Lb + RJ_o)s^2 + (Rb + K^2)s} \tag{12}$$

$$G(s) = \frac{\theta}{V} = \frac{0.003333}{0.002004s^3 + 0.3257s^2 + 0.8625s} \tag{13}$$

For velocity output:

$$\frac{\omega}{V} = \frac{Kn}{(LJ_o)s^2 + (Lb + RJ_o)s + (Rb + K^2)} \tag{14}$$

Where $J_o = Jm + (n^2)J_L$ and $n = n_m/n_L$ Our open loop system's Block diagram is shown in Figure(2).

Parameter	Values	Unit
J_m	0.04	$kg.m^2$
J_L	0.5	$kg.m^2$
b	0.1	$kg.m^2/s$
K	0.25	$N.m/A$
R	8	Ω
L	0.05	H
n	1/75	–
η	100%	–

TABLE I: Parameters

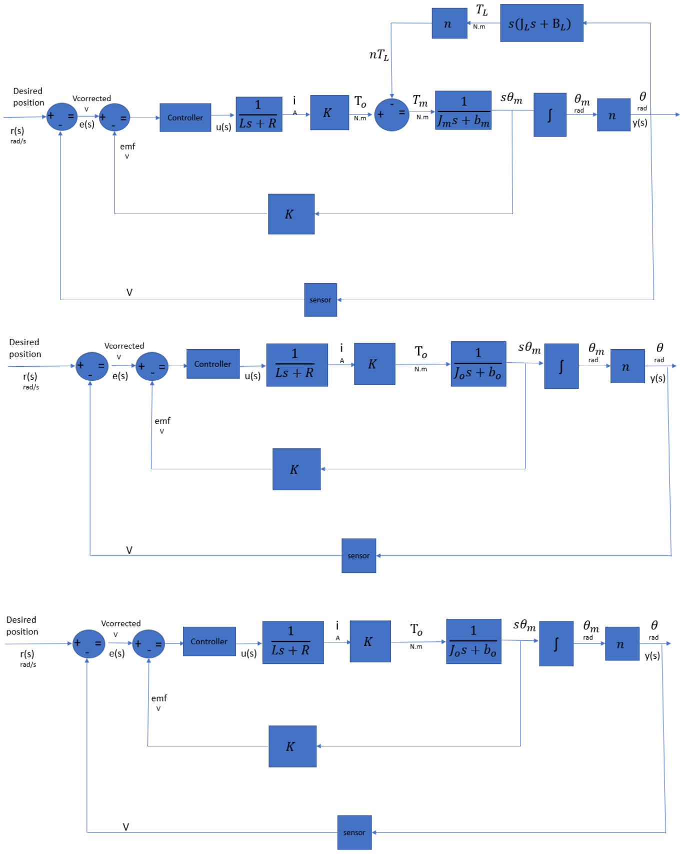


Fig. 2: Block Diagram Variations for Closed Loop System

III. Analysis

First, the performance specifications required by the customer need to be established. The desired position of the arm is 50°, which is our ideal steady state value.

It needs to be reached in $1s$, so our desired rise time is $1s$.

It needs to stop within $\pm 1°$ in the span of $2s$, so our maximum steady state error is 2% and our settling time is $2s$

Work pieces would not fit if it stops $\pm 2°$ from its desired position.

In summary, the following specifications are obtained in Equation (15):

$$\begin{cases} \theta_{desired} = 50° \\ t_r \leq 1s \\ t_s \leq 2s \\ error_{ss} \leq 2\% \\ PO \leq 4\% \end{cases} \tag{15}$$

It should be noted that the 100% criteria for rise time and 2% tolerance band for settling time are used.

A. Open Loop Analysis

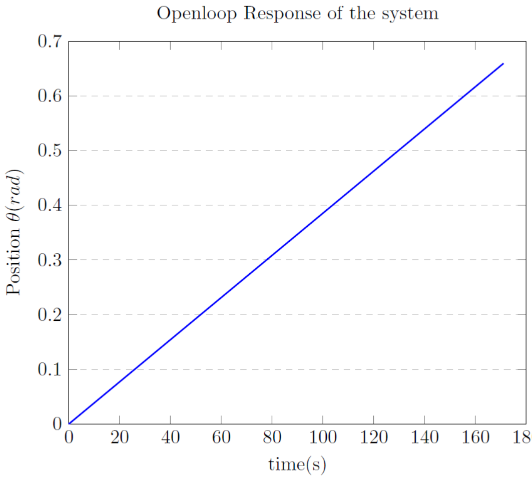


Fig. 3: Open Loop System Position Response for Unit Step input

The Position open-loop system is a type 1, 3rd order system, while the Velocity open-loop system is a type 0, 2nd order system. This drives us to expect that the position open-loop response will be marginally stable due to the pole at s=0 and that the velocity response will be stable if its 2 poles are negative.

The Open-Loop System Step Response of Position is a straight line increasing with time as seen in Figure(3). This is expected, since the input into the DC motor is a constant voltage of $1V$.So, the velocity is going to reach steady state. This results in the position increasing linearly. This means the overshoot, settling time, and steady state error of this step response for position cannot be studied, since it is unstable.

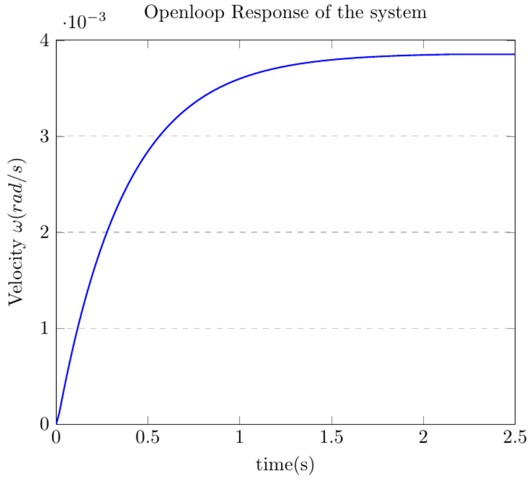


Fig. 4: Open Loop System Step Response for Velocity

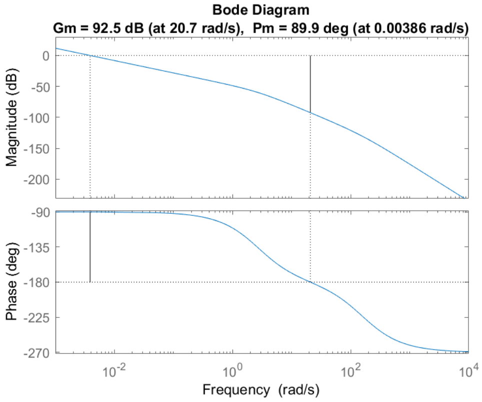


Fig. 5: Phase and Gain Margins using Bode Plot For Openloop

As it was expected from our analysis of the position response to our Open-Loop system, the velocity step response reaches steady state stability at 2.5 seconds. The velocity response looks like a first order system response. So, the system happens to be critically or over damped. It is calculated and found that truly $\zeta = 1$, which indicates that the system is critically-damped explaining this step response. For velocity Step Response it is found that:

$$\begin{cases} t_r = 0.816s \\ t_s = 1.4592s \end{cases} \tag{16}$$

The steady state error for the velocity Open-Loop Response cannot be discussed, since the input and output are of different units, and the desired value of velocity is unknown.

IV. Design Alternatives

Before going into design criteria, establishing the performance specifications is needed. Since a maximum overshoot of 4% along with a settling of $t_s = 2s$ and a rise time of $t_r = 1s$ and a steady state error of 2%.

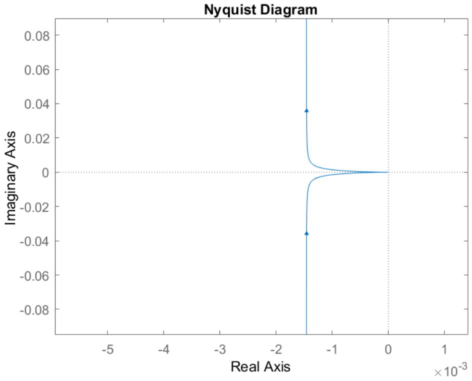


Fig. 6: Nyquist Plot For Openloop

To achieve these performance specifications, assuming the system as a 2nd order, the required damping and natural frequency are:

$$\begin{cases} \zeta \geq 0.7156 \\ \omega_n \geq 2.7947 \end{cases} \tag{17}$$

Resulting with our desired closed loop poles to be at:

$$S_{1,2} = \begin{cases} -\zeta\omega_n \pm \sqrt{1-\zeta^2}j & \zeta \leq 1 \\ -\zeta\omega_n \pm \sqrt{\zeta^2 - 1}j & \zeta > 1 \end{cases} \tag{18}$$

the threshold desired closed-loop poles are:

$$S_{1,2} = -2 \pm 1.952j \tag{19}$$

Something to be noted is that a linear control system is being dealt with, therefore the closed loop system response for a unit step input will be linear correlated to any other step input. In our analysis, a unit step input is used to facilitate our design process. However, it is noted that our input should be 50°.

It is assumed that unity feedback will be used and the encoder will give the position value in the same units as that needed to be inputted.

A. Proportional Controller

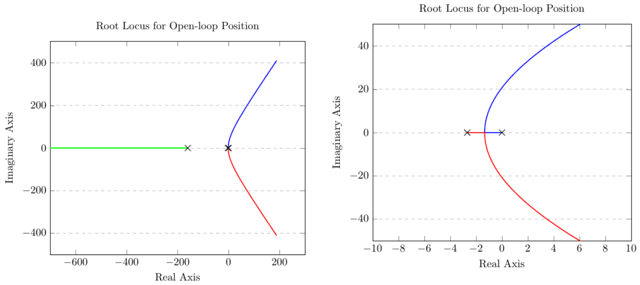


Fig. 7: Root Locus of Openloop

It is observed that one of the 3 open loop poles is more than 10 times greater than the other two, so it can be neglected and assumed to act like a second order system.

It is also observed that the region that it is allowed to pick gains for our desired poles doesn't include the Root locus; however, it can at least satisfy the damping ratio $\zeta = 0.716$. The obtained gain and closed loop poles are:

$$\begin{cases} K = 333.93 \\ S_{1,2} = -1.34 \pm 1.3j \\ S_3 = -159.82 \end{cases} \tag{20}$$

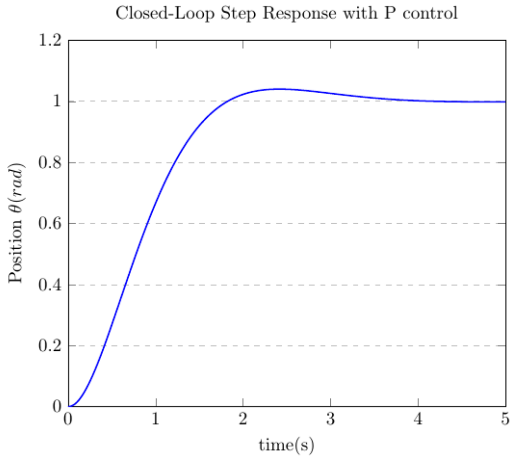


Fig. 8: Step Response for Closed loop Position Proportional Controller

The performance of the closed loop using this Proportional Controller is:

$$\begin{cases} t_r = 1.85s \\ t_s = 3.19s \\ PO = 3.9735\% \\ error_{ss} = 0.16\% \end{cases} \tag{21}$$

It can be seen that all of the performance criteria are met except for the settling time. For the entire range of gains, a satisfactory settling time cannot be obtained. Therefore, a proportional controller isn't enough and other options need to be investigated.

It should be noted that the steady state error is almost zero, which is expected since the open-loop is type 1.

It can be observed in Figure(9) that our Gain margin $G_m = 42dB$ at $w = 20.7rad/s$ and our Phase margin $P_m = 65.9°$ at $w = 1.18rad/s$. The cutoff frequency is found to be at $w_b = 1.57rad/s$.

B. PI Controller

For the PI controller, a pole at s=0 is added, and the angle of deficiency to get our desired closed loop poles is evaluated and obtained to be $\phi = 162.5635°$. Using angle of deficiency method, the zero at s=4.2149 is obtained, which results in an unstable closed loop pole. Hence, a PI controller cannot be used to obtain our desired poles. This is expected, since PIs are only used to eliminate steady state error and not to improve settling time or maximum overshoot.

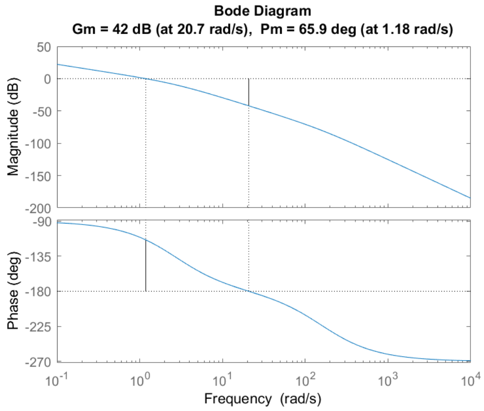


Fig. 9: Phase and Gain Margins using Bode Plot For P controller

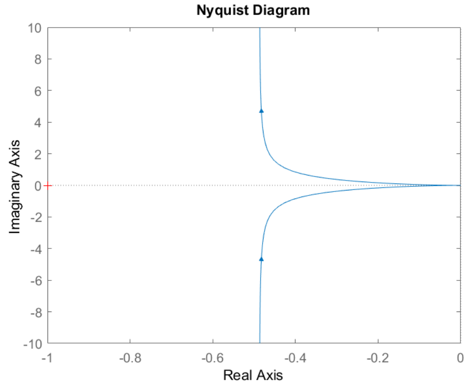


Fig. 10: Nyquist Plot For P controller

C. PD Controller

Knowing that a PD controller is used to control the transient response of a system, high hopes exist for improving the overshoot, rise time ,and settling time using it.
The angle of deficiency can be calculated to be $\phi = 26.8673°$ and place our zero at s= -5.853.

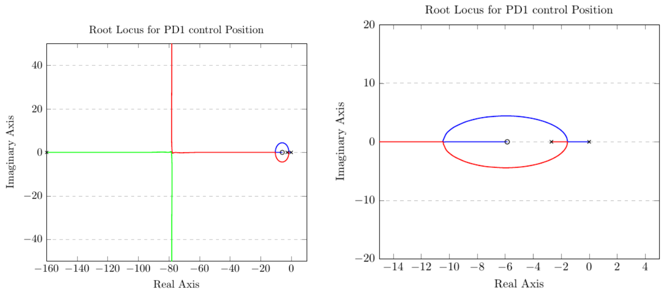


Fig. 11: Root Locus PD Controlled Openloop

One of the open loop poles is more than 10 times greater than the other two, so it can be assumed the system to behave like a 2nd order.
The threshold desired closed-loop poles Equation(19) are chosen, and they intersect with the Root locus due to our PD controller. The gain and closed loop poles obtained are:

$$\begin{cases} K = 127.0095 \\ S_{1,2} = -2 \pm 1.952j \\ S_3 = -158.5 \end{cases} \tag{22}$$

So, the transfer function of this controller is:

$$PD_1(s) = 127.0095(5.853 + s) \tag{23}$$

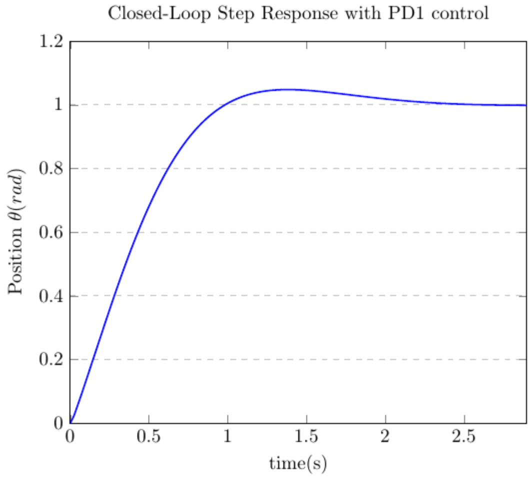


Fig. 12: Step Response for Closed loop Position PD1 Controller

The performance of the closed loop using this PD Controller is:

$$\begin{cases} t_r = 0.974s \\ t_s = 1.9656s \\ PO = 4.7337\% \\ error_{ss} = 0.18\% \end{cases} \tag{24}$$

It can be observed that all performance criteria are acceptable except for the Maximum overshoot. To decrease the maximum overshoot, the damping ratio and our natural frequency will be increased to $\zeta = 0.75$ and $\omega = 3$ respectively, which makes the desired poles:

$$S_{1,2} = -2 \pm 1.82j \tag{25}$$

Following the same steps done for the first PD controller, but for the poles in Equation(25),an open-loop zero at s=-4.9087 will be added,and the obtained gain and closed loop poles are:

$$\begin{cases} K = 174.2554 \\ S_{1,2} = -2.25 \pm 1.98j \\ S_3 = -157.99 \end{cases} \tag{26}$$

So, the transfer function of this controller is:

$$PD_2(s) = 174.2554(4.9087 + s) \tag{27}$$

and our performance is:

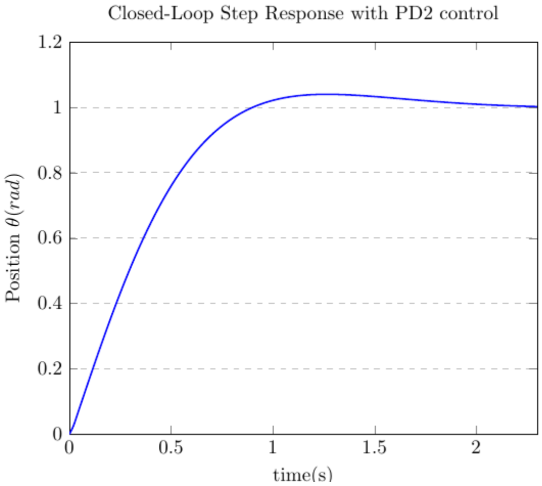


Fig. 13: Step Response for Closed loop Position PD2 Controller

$$\begin{cases} t_r = 0.898s \\ t_s = 1.7514s \\ PO = 3.9684\% \\ error_{ss} = 0.016\% \end{cases} \tag{28}$$

All our performance specification are met by using PD2 controller, making it a good candidate to consider.

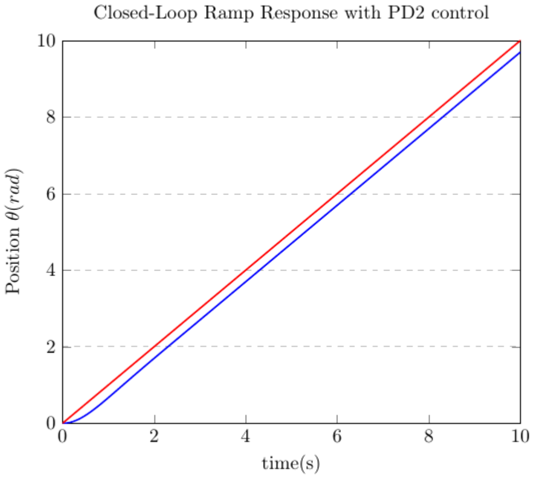


Fig. 14: Ramp Response for Closed loop Position PD2 Controller

The system was also simulated for a unity ramp input and it was obtained that the steady state error is 30.256% as shown in Figure(14). for $R(s) = \frac{1}{s^2}$

$$
\begin{aligned}
e_{ss} &= \lim_{s\to 0} \frac{sR(s)}{1+PD_2(s)*G(s)} \\
&= \lim_{s\to 0} \frac{1}{s+sPD_2(s)*G(s)} \\
&= \lim_{s\to 0} \frac{0.002004s+0.3257s+0.8625}{0.002004s^3+0.3257s^2+0.8625s+0.5809s+2.851} \\
&= \frac{0.8625}{2.851} \\
&= 0.3025
\end{aligned}
$$

As seen above, the analytical calculation of the error agrees with our simulation findings of a steady state error of 30.25%

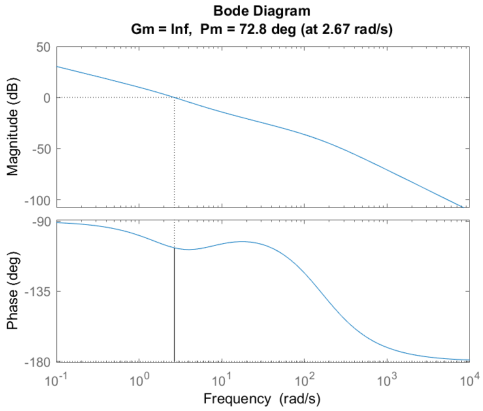


Fig. 15: Phase and Gain Margins using Bode Plot For PD2 controller

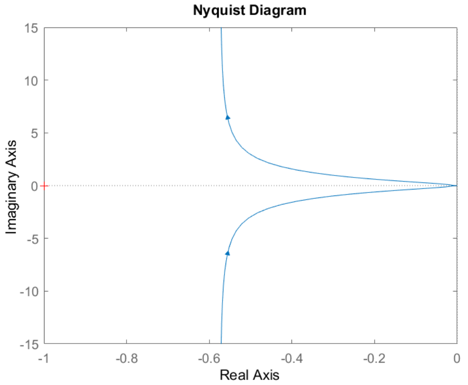


Fig. 16: Nyquist Plot For PD2 controller

It can be observed in Figure(15) that our gain margin is $G_m = \inf dB$ at $w = \inf rad/s$ and our phase margin is $P_m = 72.8°$ at $w = 2.67 rad/s$. It was also found that the cutoff frequency to be at $w_b = 3.5 rad/s$

D. PID Controller

Knowing that PID controller are used to tackle both the transient and steady state response, it is expected that it will be able to give us our desired performance. To design this controller, an open-loop at s=0 and an open loop zero at s=-5 are added, and the angle of deficiency is calculated to be $\phi = 129.5°$. The 2nd open loop zero is calculated to be at s=-0.3902.

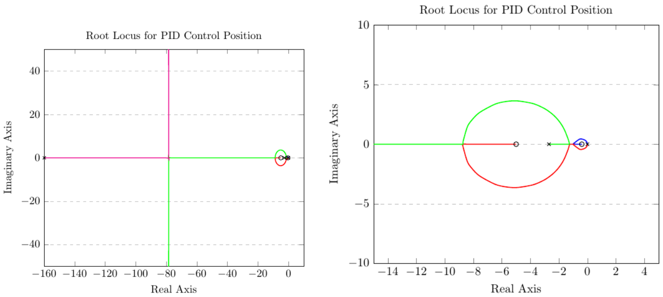


Fig. 17: Root Locus PID Controlled Openloop

The threshold desired closed-loop poles in Equation(19) were picked, which intersect with the Root locus due to our PID controller. The gain and closed loop poles obtained are:

$$
\begin{cases}
K = 169.8302 \\
S_{1,2} = -2 \pm 1.95j \\
S_3 = -158.5 \\
S_4 = -0.45
\end{cases} \tag{29}
$$

So, the transfer function of this controller is:

$$
PID(s) = \frac{169.83(5+s)(0.3902+s)}{s} \tag{30}
$$

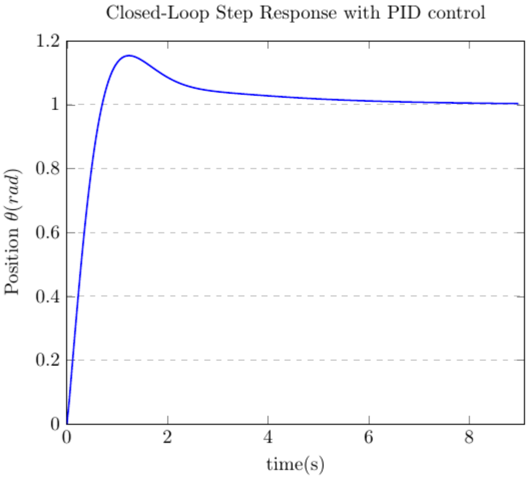


Fig. 18: Step Response for Closed loop Position PID Controller

and the performance of our system with this PID is:

$$
\begin{cases}
t_r = 0.5226s \\
t_s = 4.7085s \\
PO = 15.3078\% \\
error_{ss} = 0.3\%
\end{cases} \tag{31}
$$

It can be seen that the settling time and the overshoot is too high. Even the steady state error is greater than that in the PD controller. This is expected, since a pole at s=-0.45 is present, which slows down the system a lot. No matter, how the magnitude of the gain selected, there will always be a pole that is very close to the imaginary axis. So, the PID will always be slow and will not be able to reach our desired settling time. If the first zero is selected to be at s=0, the PID controller will cancel out a pole and add another at s=0, resulting in the same transfer function for the PID as that of the PD in Equation(23). If our initial zero is taken between our 2 open loop pole s=0 and s=-2.6927, an open-loop zero in the right half plane will be added by the PID, which results in an unstable closed loop system for all gain K.

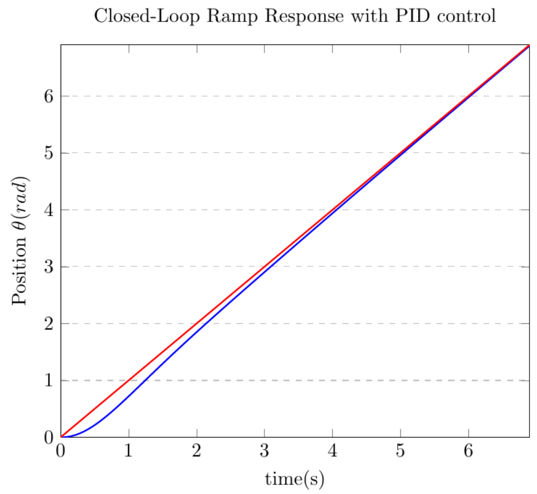


Fig. 19: Ramp Response for Closed loop Position PID Controller

It can be seen in Figure(19) that the steady state error for a unit ramp input is 0, this is expected since the system is a type 2 system.

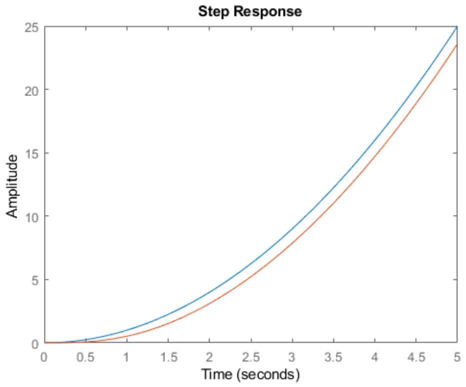


Fig. 20: Parabola Response for Closed loop Position PID Controller

The system is also simulated for a unity parabola input ,and the steady state error is obtained to be 156.25% as shown in Figure(20). for $R(s) = \frac{2}{s^3}$

$$\begin{aligned} e_{ss} &= \lim_{s \to 0} \frac{sR(s)}{1 + PID(s) * G(s)} \\ &= \lim_{s \to 0} \frac{2}{s^2 + s^2 PID(s) * G(s)} \\ &= \frac{1.7250}{1.104} \\ &= 1.5625 \end{aligned}$$

As shown above, the analytical calculation of the error agrees with our simulation findings of a steady state error of 156.25%.

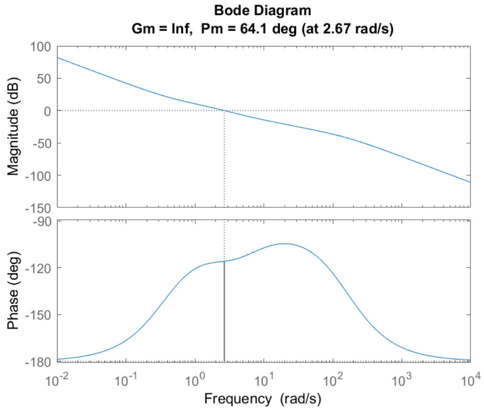


Fig. 21: Phase and Gain Margins using Bode Plot For PID controller

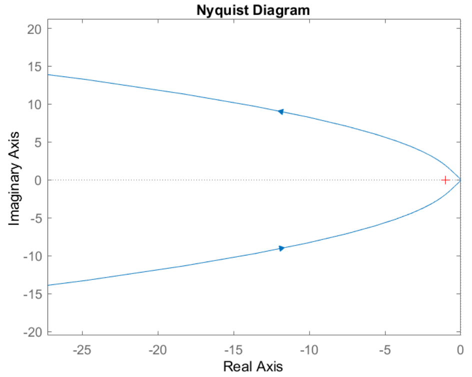


Fig. 22: Nyquist Plot For PID controller

It can be observed in Figure(21) that our Gain margin $G_m = \inf dB$ at $w = \inf rad/s$ and our Phase margin $P_m = 64.1°$ at $w = 2.67rad/s$. The cutoff frequency is found to be at $w_b = 3.5rad/s$.

E. Lead Compensators

Knowing that a Lead compensator is used to control the transient response of a system, it is expected to decrease our settling time and overshoot.

1) Time Domain Design: An open loop zero is placed on top of one of the already existing poles at s=-2.6927. Afterwards, the angle of deficiency is calculated to be $\phi = 43.5952°$, consequently obtaining an open pole at s=-4.0501. One of the poles is more than 10 times greater than the other 3, so it can be neglected. One of the other 3 is negated by the zero we placed using our lead controller. Hence, we can assume the system to behave like a 2nd order.
The threshold desired closedloop poles from Equation(19) are selected, which interest with the Root locus due to our Lead Compensator. The gain and closed loop poles are obtained to be:

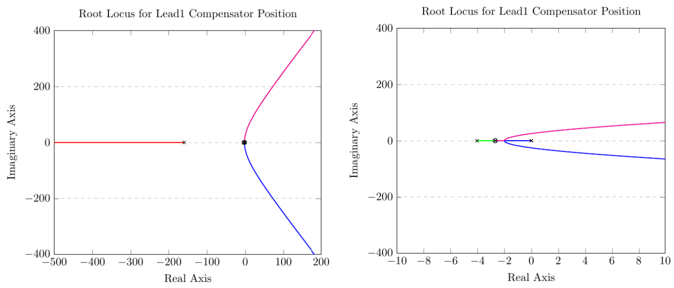


Fig. 23: Root Locus Lead Compensator Openloop

$$\begin{cases} K = 750.3234 \\ S_{1,2} = -2 \pm 1.95j \\ S_3 = -159.85 \\ S_4 = -2.69 \end{cases} \tag{32}$$

So, the transfer function of our lead compensator is:

$$Lead_1(s) = \frac{750.3234(2.6927 + s)}{4.0501 + s} \tag{33}$$

and the achieved performance of our system with this Lead compensator is:

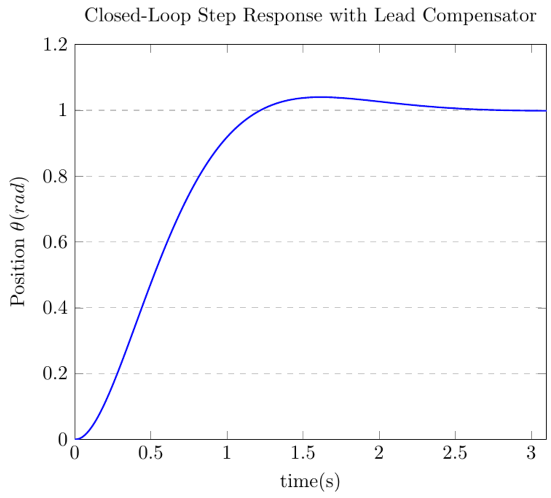


Fig. 24: Step Response for Closed loop Position Lead1 Compensation

$$\begin{cases} t_r = 1.22s \\ t_s = 2.1312s \\ PO = 3.9913\% \\ error_{ss} = 0.16\% \end{cases} \tag{34}$$

It can be seen that the rise time and settling time are too high for us, This might be due to the pole at s=-2.69, which is relatively close to the imaginary axis, slowing down the response. To decrease the settling and rise time, our natural frequency will be increased to $w = 4$ and to counteract the high overshoot by increasing our damping to $\zeta = 0.75$, making our desired poles ot become:

$$S_{1,2} = -3 + 2.6458j \tag{35}$$

The previous design steps for $Lead_1$ are repeated , but for the desired poles in Equation(35), our open loop pole is obtained at s=-6.104. Our gain and closed loop poles are obtained to be:

$$\begin{cases} K = 1534.7 \\ S_{1,2} = -3 \pm 2.6458j \\ S_3 = -159.91 \\ S_4 = -2.69 \end{cases} \tag{36}$$

So, the transfer function for this Lead compensator is:

$$Lead_2(s) = \frac{1534.7(2.6927 + s)}{6.104 + s} \tag{37}$$

and our performance is:

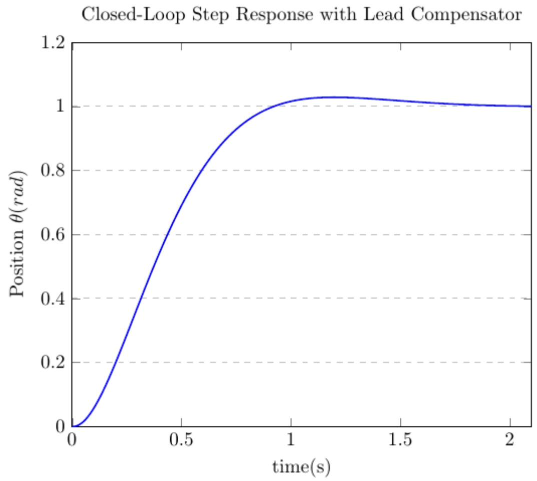


Fig. 25: Step Response for Closed loop Position Lead2 Compensation

$$\begin{cases} t_r = 0.919s \\ t_s = 1.4415s \\ PO = 2.8071\% \\ error_{ss} = 0\% \end{cases} \tag{38}$$

It can be observed that this compensator achieves all our performance specifications, making it a good candidate to consider.

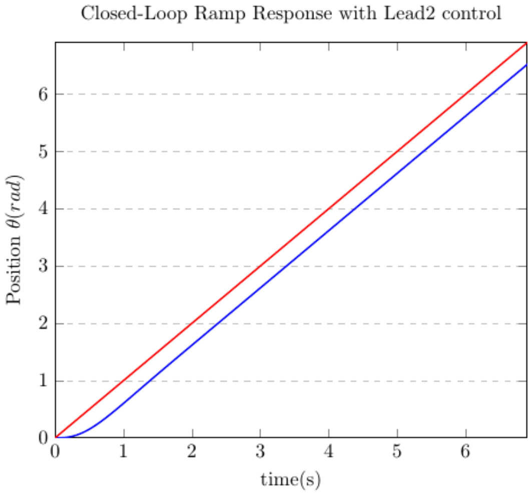


Fig. 26: Ramp Response for Closed loop Position Lead2 Controller

The system was simulated for a unity ramp input and the steady state error is obtained to be 38.24% as shown in Figure(26). for $R(s) = \frac{1}{s^2}$:

$$\begin{aligned} e_{ss} &= \lim_{s \to 0} \frac{sR(s)}{1 + Lead_2(s) * G(s)} \\ &= \lim_{s \to 0} \frac{1}{s + sLead_2(s) * G(s)} \\ &= \frac{5.265}{13.77} \\ &= 0.3824 \end{aligned}$$

As seen above, the analytical calculation of the error matches with our simulation findings of a steady state error of 30.25%

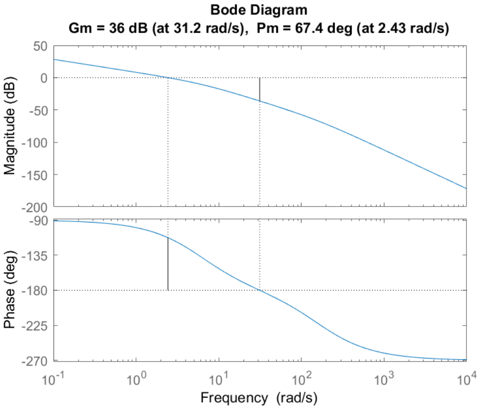


Fig. 27: Phase and Gain Margins using Bode Plot For Lead2 controller

It can be observed in Figure(27) that our Gain margin $G_m = 35.9dB$ at $w = 31.2rad/s$ and our Phase margin $P_m = 67.4°$ at $w = 2.43rad/s$. The cutoff frequency is found to be at $w_b = 3.26rad/s$

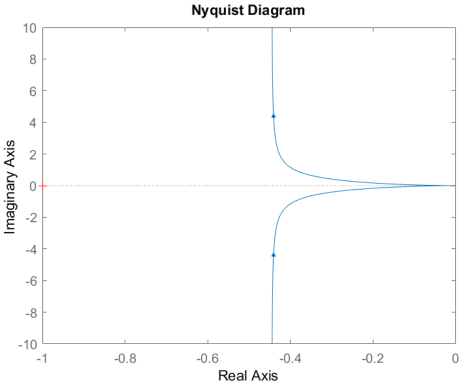


Fig. 28: Nyquist Plot For Lead2 controller

F. Lead Frequency Domain Design

The frequency specification are picked to be the characteristics of the $Lead_2$ compensator, since it gives desired time specifications.
The velocity static error is known to be equal to the gain crossover frequency $K_v = 2.43rad/s$. The gain and phase margins are known to be $G_m = 35.9dB$ and $P_m = 67.4°$ respectively.

$$\begin{aligned} K_v &= \lim_{s \to 0} sKG(s) \\ &= \lim_{s \to 0} \frac{0.003333K}{0.002004s^2 + 0.3257s + 0.8625} \\ &= 0.0039K \\ &= 2.43 \end{aligned}$$

So, $K = 628.8254$ $G_1(s) = KG(s)$ is defined, and its bode plot is shown in Figure(29). It can be observed that the gain margin and phase margin are 36.5 and 53.2° respectively.

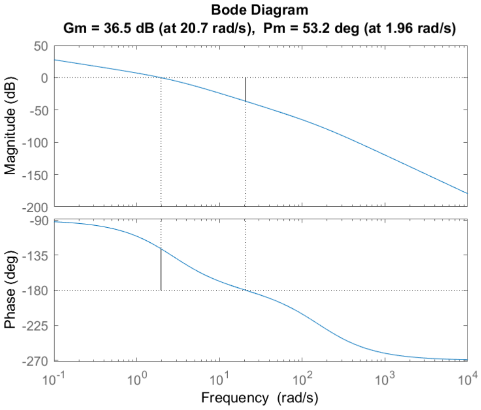


Fig. 29: Phase and Gain Margins using Bode Plot For $G_1(s)$

The phase-lead angle is calculated to be $\phi_m = 14.2°$ and an additional 8° is added to reach $\phi_m = 22.2°$, since the lead compensator shifts the gain cross-over frequency to the right and decreases phase margin.

$$sin(\phi_m) = \frac{1-\alpha}{1+\alpha} \tag{39}$$

Next $\alpha = 0.4515$ is obtained from the relationship in Equation(39)

The new gain cross over frequency is found by locating the frequency at which:

$$|G_1(s)| = -20log(\frac{1}{\sqrt{\alpha}}) = -3.453 \tag{40}$$

$W_m = 2.6rad/s$ is obtained from the Bode Plot in Figure(29)

$$w_m = \frac{1}{\sqrt{\alpha}T} \tag{41}$$

T is found from the relationship in Equation(41) to be $T = 0.57239$. The Lead Compensator's pole, zero and gain is found to be:

$$\begin{cases} K_c = \frac{K}{\alpha} = 1392.7472 \\ zero = -\frac{1}{T} = -1.747 \\ pole = -\frac{1}{\alpha T} = -3.869458 \end{cases} \tag{42}$$

So, the $Lead_3$ compensator's transfer function is:

$$Lead_3 = \frac{1392.7472(s+1.747)}{s+3.869458} \tag{43}$$

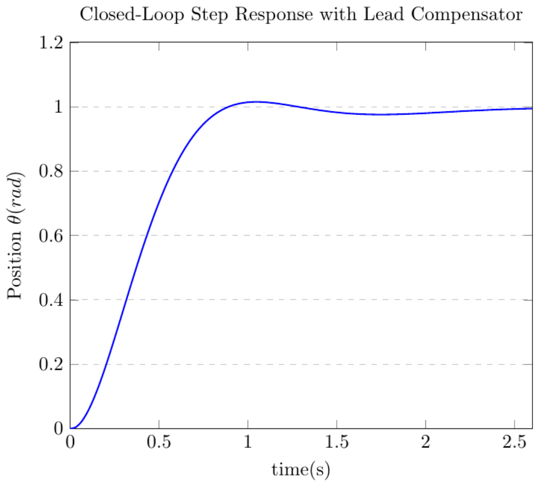


Fig. 30: Step Response for Closed loop Position Lead3 Compensation

The following time domain characteristics are obtained:

$$\begin{cases} t_r = 0.894s \\ t_s = 1.9962s \\ PO = 1.5048\% \\ error_{ss} = 0.29\% \end{cases} \tag{44}$$

All the time domain characteristics are satisfactory, which makes this lead compensator a viable candidate to be used.

For the frequency domain characteristics, the phase margin is satisfactory, but the gain margin isn't. Repeating the design is needed until satisfactory characteristics are obtained.

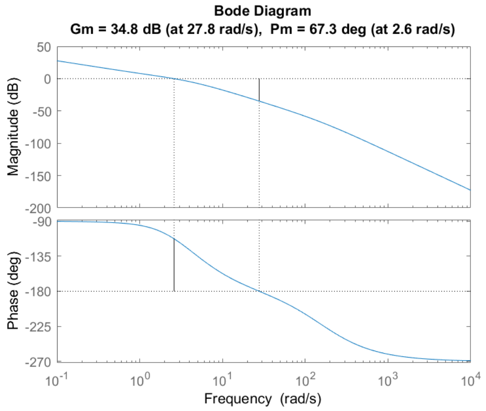


Fig. 31: Phase and Gain Margins using Bode Plot For Lead3 Compensator

G. Lag Compensators

It is known that lag compensators are used to improve our steady state response, when the transient response is satisfactory. Since this is not the case for this system, lag compensators will not be even considered to control our system.

H. Lead-Lag Compensators

An acceptable steady state error for our system was already obtained by using the lead compensator in Equation(37), so a lag compensator does not need to be added to it. If a lag-lead compensator is used, it will only slow down our system and decrease the relative stability of our system by adding a not fully negated pole near the imaginary axis. It will decrease the steady state error for a ramp input, but that is not of value to us according to the requirements.

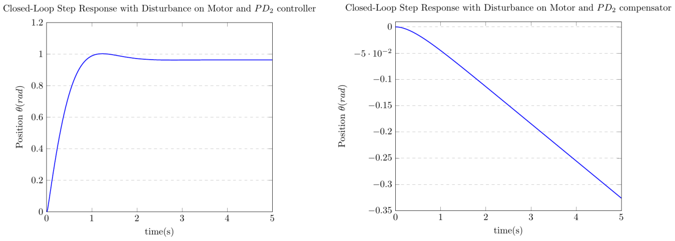
Using Nyquist Plots it can be deduced that for the original open loop, PD_1, PD_2, $Lead_1$, $Lead_2$, $Lead_3$ and PID: there are 3 open loop poles, but none of them are in the right hand side of the s-plane then $P = 0$(where P is the number of open loop poles in the right had side). From the Nyquist plot it can be seen that the Nyquist curve does not encircle the point $s = -1$, then $N = 0$, where N is the number of clockwise encirclements of the point $s = -1 + 0j$. Then, $Z = N - P = 0 - 0 = 0$ (where Z is the number of open loop zeros in the right had side). This indicates that the system is stable for all values of K.

It can also be inferred from the Bode Plots of the system with every controller that they all act as low pass filters. This causes any noise that passes through them to be negated. The range of signals that pass have a frequency between $0 \leq w \leq w_b$ for each controller.

V. Final Design

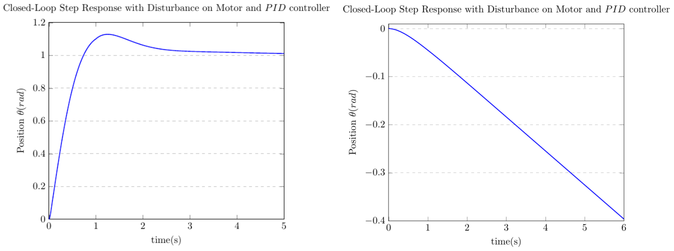
Besides the PI controller, all of the designs resulted in stable output for a step input, but only the PD2 controller as well as the Lead2 and $Lead_3$ compensators met our performance requirements. From Equations (28), (38) and (43), we observe that the steady state error for all three is practically negligible, and that the rise time is roughly similar across the designs. $Lead_3$ provides the lowest overshoot, but the settling time is almost at 2 seconds, which is not desirable from a robustness perspective, as any change in the conditions might slow down the system beyond the lowest desired speed. Therefore, we settle for $Lead_2$ as it results in the lowest settling time without sacrificing much on overshoot. Stability, performance and robustness are now all met.

A. Rejecting Disturbances



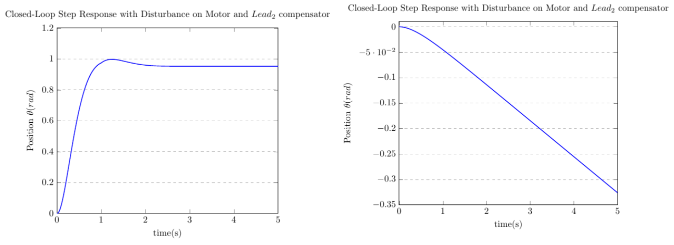
(a) Without Saturation (b) With Saturation

Fig. 32: Closed Loop Unit Step response for Unit step Disturbance on Motor & PD_2 Controller



(a) Without Saturation (b) With Saturation

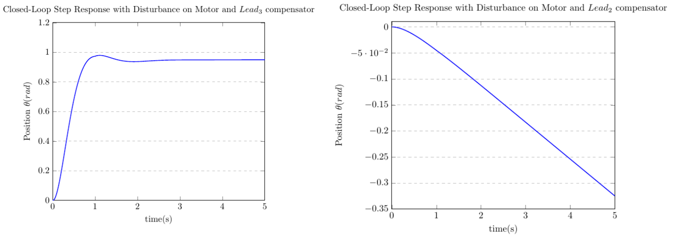
Fig. 33: Closed Loop Unit Step response for Unit step Disturbance on Motor & PID Controller



(a) Without Saturation (b) With Saturation

Fig. 34: Closed Loop Unit Step response for Unit step Disturbance on Motor & $Lead_2$ Controller

Upon adding a unit disturbance torque to the motor shaft, it is shown that the response of the system with different controllers without adding a saturation limit to the voltage can counteract this torque disturbance without affecting the response curve, where the maximum value of the response to a unit disturbance is 1 rad/s since the input is a step of 1 rad/s. However, when a saturation limit of 10-25V is added to the voltage, the response of the system completely changes. This is due to the fact that saturation slows down the system's response where the voltage induced torque takes a longer time to counteract the disturbance. So, the from the system's response it would not be able to reject the disturbance induced.



(a) Without Saturation (b) With Saturation

Fig. 35: Closed Loop Unit Step response for Unit step Disturbance on Motor & $Lead_3$ Controller

In the Simulink model without saturation, the voltage is observed to suddenly peak to 18000 V at the start and slowly decrease. This high voltage is needed to reach the desired performance specification. If saturation according to the description motor between 10-25 V DC is added, it is not possible to reach these specifications. This shows that the motor used does not provide enough power to obtain the required performance. Therefore, another motor that is capable of providing this power is needed which will increase the cost.

VI. Physical Model

To validate $Lead_1$ compensator, creating a physical controller and system was attempted, but the gearing and the load were not taken into account, and it was assumed that the characteristics of the used physical motor were the same as that of the one in the system.

Our transfer function was reevaluated based on these changes and assumptions, resulting in the following open loop transfer function:

$$G(s) = \frac{0.003333}{0.002s^3 + 0.325s^2 + 0.8625s} \tag{45}$$

For the same specifications as $Lead_1$,the same desired poles in Equation(19) were obtained.

The Lead compensator obtained is:

$$Lead_{1mod}(s) = \frac{729.8322(s + 2.6987)}{s + 4.0501} \tag{46}$$

For the physical model, a myDAQ, breadboard, 5V DC motor is used to replicate our open-loop system. An optical encoder is used to measure the position, after designing a cardboard shape to check for the change of position θ with

a resolution of 10° as shown in Figure(36).
The video of the running open-loop system can be accessed through this link.

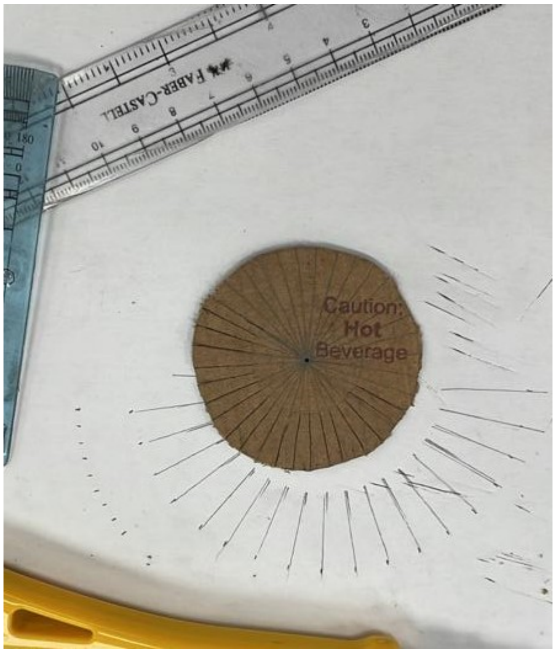


Fig. 36: Designing Cardboard Piece for using Optical Encoder

The resistors and capacitors needed are found using the following equations:

$$\begin{cases} 742 = \frac{R_4 C_1}{R_3 C_2} \\ 2.699 = \frac{1}{R_1 C_1} \\ 4.0501 = \frac{1}{R_2 C_2} \end{cases} \tag{47}$$

Assuming $R_1 = R_2 = R_3 = 10k\Omega$:

$$\begin{cases} C_1 = 37\mu F \\ C_2 = 24.7\mu F \\ R_4 = 4.9M\Omega \end{cases} \tag{48}$$

Due to the limitations of the resistors and capacitors available, the used resistances and capacitors are as shown in Figure(37) Unfortunetly, after assembling the entire circuits. The motor did not spin; however, the voltage given to the motor was controlled to the MyDAQ's voltage capacity. Initially, this could be due to the low impedance of the motor compared to the rest of our circuit, so a buffer was added, but it did not improve it. This is due to the fact that the MyDAQ does not offer enough voltage to control the motor and a motor driver is needed.

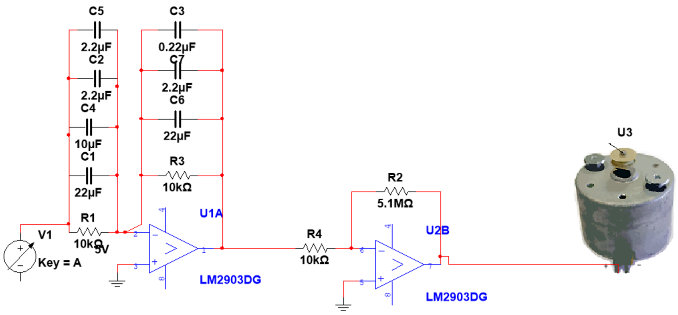


Fig. 37: Openloop System with Lead Compensator

The video of the final measurements of the voltage between the motor terminals using a multimeter can be accessed through this link.

Finally, the final circuit and system can be seen in Figure(38).

VII. Acknowledgment

We would like to thank Dr. Naseem Daher for his help and guidance throughout this project.

References

[1] N. Thomas and D. P. Poongodi, "Position control of dc motor using genetic algorithm based pid controller," in *Proceedings of the world congress on engineering*, vol. 2, pp. 1–3, London, UK, 2009.

[2] M. M. Maung, M. M. Latt, and C. M. Nwe, "Dc motor angular position control using pid controller with friction compensation," *International journal of scientific and research publications*, vol. 8, no. 11, p. 149, 2018.

[3] H. Paul and S. Lasrado, "Precise control of angular position of geared dc motors for low cost applications," *International Journal of Innovative Research in Science, Engineering and Technology*, vol. 5, no. 9, pp. 574–578, 2016.

[4] M. A. Akbar *et al.*, "Simulation of fuzzy logic control for dc servo motor using arduino based on matlab/simulink," in *2014 International Conference on Intelligent Autonomous Agents, Networks and Systems*, pp. 42–46, IEEE, 2014.

[5] G. Zhang, J. Cao, and F. Yu, "Design of active and energy-regenerative controllers for dc-motor-based suspension," *Mechatronics*, vol. 22, no. 8, pp. 1124–1134, 2012.

[6] M. Namazov and O. Basturk, "Dc motor position control using fuzzy proportional-derivative controllers with different defuzzification methods," 2010.

[7] C. H. Aung, K. T. Lwin, and Y. M. Myint, "Modeling motion control system for motorized robot arm using matlab," *World Academy of Science, Engineering and Technology*, vol. 42, pp. 372–375, 2008.

[8] M. M. Ali and A. Al-Khawaldeh, "A simulation study of multi-disciplinary position control methods of robot arm dc motor," in *2016 13th International Multi-Conference on Systems, Signals & Devices (SSD)*, pp. 489–493, IEEE, 2016.

[9] D. Chowdhury and M. Sarkar, "Digital controllers in discrete and continuous time domains for a robot arm manipulator," *arXiv preprint arXiv:1912.09020*, 2019.

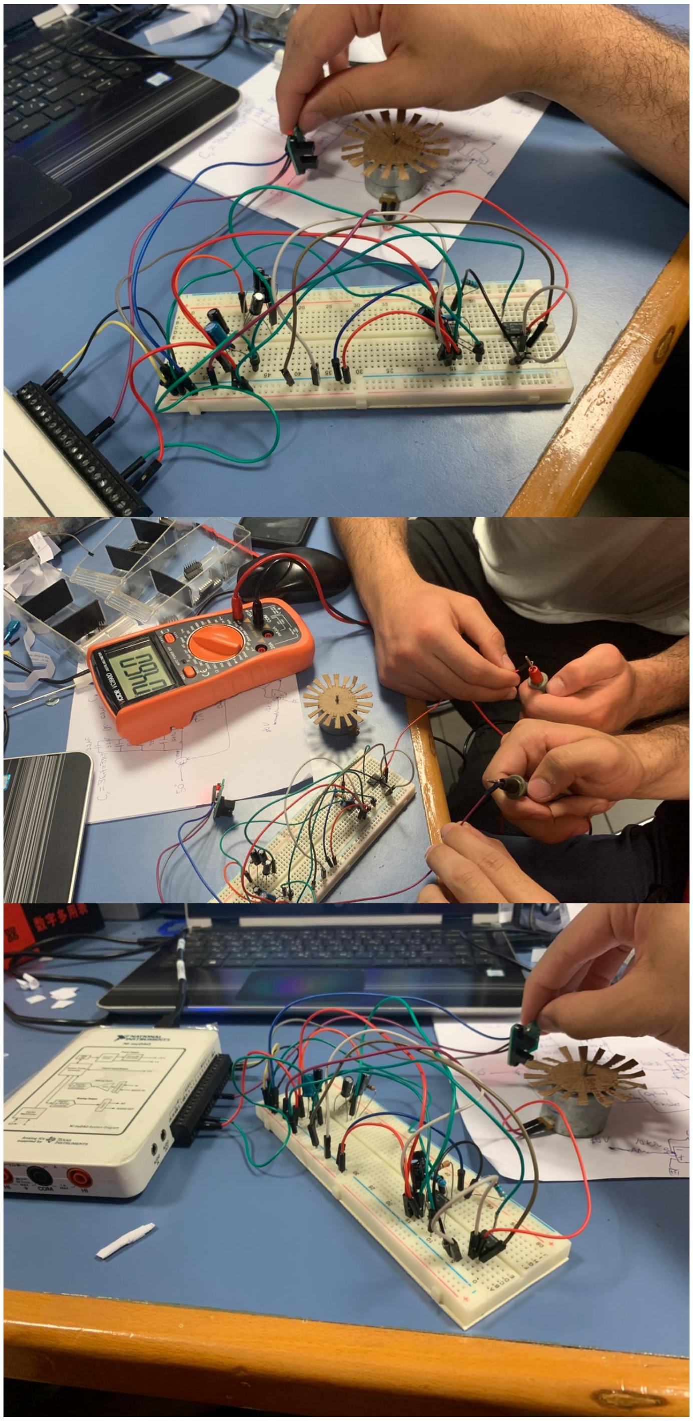


Fig. 38: Modified Physical Model of the system