MECH 467: Digital Control of the Ball-Screw Feed Drive

Project II Report

Ryan Edric Nashota
ID: 33508219

December 10, 2025

1 Abstract

This laboratory project focused on the design and implementation of digital controllers for a ball-screw feed drive. A Proportional (P), Lead-Lag, and Lead-Lag-Integrator controller were designed in the frequency domain to meet specific bandwidth and phase margin requirements. Experimental results confirmed that the Proportional controller provided basic tracking, the Lead-Lag compensator improved bandwidth and rise time, and the Lead-Lag-Integrator eliminated steady-state errors, albeit with increased overshoot.

2 System Analyzed

The open-loop model from the Project II handout removes the Coulomb friction and saturation elements and collapses the amplifier, torque constant, inertia, damping, and encoder dynamics into a single continuous-time transfer function. Using the provided parameters ($K_a = 0.887$ A/V, $K_t = 0.72$ Nm/A, $J_e = 7 \times 10^{-4}$ kgm^2, $B_e = 0.00612$ Nms/rad, $K_e = 20/(2\pi)$ mm/rad, $K_d = 1$) the open-loop gain is

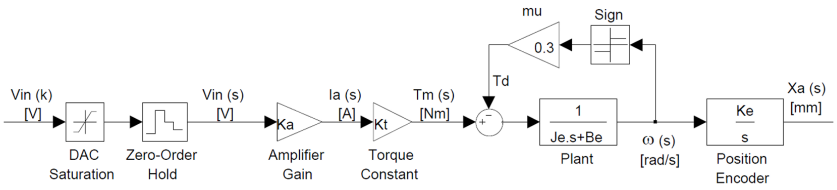

Figure 1. Analyzed System from Project Handout

$$G_{\text{ol}}(s) = \frac{K_a K_t K_e}{s(J_e s + B_e)} = \frac{2.0329}{0.0007s^2 + 0.00612s}. \tag{1}$$

The pole at the origin reflects displacement integration while the real pole at $-B_e/J_e = -8.743$ rad/s captures the motor–ball-screw mechanical time constant. The subsequent sections translate this model into discrete form and design digital controllers that satisfy the pre-lab deliverables.

3 Prelab 1 – Discrete Transfer Function Derivation

Q1 Deliverables

- Manually obtain the zero-order hold equivalent of the open loop transfer function ($G_{ol}(z)$) in terms of system parameters.
- Plug in the parameter values in the final result, and compare your result with the zero order hold transfer function obtained by Matlab's c2d command.
- Ignore the Coulomb friction and saturation blocks.

3.1 Zero-Order Hold Derivation

The partial-fraction decomposition of Equation (1) is

$$G_{\mathrm{ol}}(s) = \frac{332.166}{s} - \frac{37.993}{0.11438\,s + 1} = \frac{332.166}{s} - \frac{37.993}{s + 8.7429}. \tag{2}$$

Applying the standard ZOH expressions

$$\mathcal{Z}\left\{\frac{1}{s}\right\} = \frac{Tz^{-1}}{1 - z^{-1}}, \tag{3}$$

$$\mathcal{Z}\left\{\frac{1}{s + a}\right\} = \frac{(1 - e^{-aT})z^{-1}}{1 - e^{-aT}z^{-1}}, \tag{4}$$

with $T = 0.0002$ s yields the zero-order-hold equivalent

$$G_{\mathrm{ol}}(z) = \frac{332.166\,Tz^{-1}}{1 - z^{-1}} - \frac{37.993(1 - e^{-8.7429T})z^{-1}}{1 - e^{-8.7429T}z^{-1}}. \tag{5}$$

Substituting the numerical constants and simplifying gives

$$G_{\mathrm{ol}}(z) = \frac{5.8048 \times 10^{-5}z + 5.8014 \times 10^{-5}}{z^2 - 1.99825296z + 0.99825296}. \tag{6}$$

3.2 Comparison with MATLAB `c2d`

Equation (6) matches the discrete transfer function reported by `c2d(G_ol, 0.0002, 'zoh')` in MATLAB to all significant digits. The numerator and denominator coefficients from MATLAB are shown in Figure 2

```
G_z =

  5.805e-05 z + 5.801e-05
  -----------------------
  z^2 - 1.998 z + 0.9983
```

Figure 2. Output from MATLAB c2d

which confirms the manual derivation in Equation (2)–(6).

3.3 Comparison of Manual G(z) vs MATLAB c2d

As requested, a comparison between the manually derived discrete transfer function (plotted from the analytical expression in Equation (6)) and the MATLAB/Python-generated `c2d` model is shown in Figure 3. The perfect overlap confirms the accuracy of the manual ZOH derivation.

Figure 3. Section 2.2: Manual $G_{ol}(z)$ vs MATLAB `c2d` Bode plot.

4 Prelab 2 – State Space Model

Q2 Deliverables

- Obtain the discrete time state space model of the machine shown in Figure 1.
- Simulate the step response of the system using the discrete state space model of the machine given in Figure 1.
- Compare the results obtained from discrete transfer function and state space models.

4.1 State and Input Selection

The plant contains two energy-storing elements: the inertia that integrates torque into angular velocity, and the ballscrew that integrates angular rate into position. Selecting the state vector and inputs as

$$x = \begin{bmatrix} \omega \\ x_a \end{bmatrix}, \qquad u = \begin{bmatrix} v_{\text{in}} \\ T_d \end{bmatrix}, \qquad y = x_a, \tag{7}$$

keeps the model physically meaningful while allowing a disturbance torque T_d to be injected explicitly.

4.2 Continuous-Time Derivation

The torque balance on the rotor/ball screw assembly is

$$J_e\dot{\omega} + B_e\omega = K_a K_t v_{\text{in}} - T_d, \tag{8}$$

which leads to the state equation for $\dot{\omega}$:

$$\dot{\omega} = -\frac{B_e}{J_e}\omega + \frac{K_a K_t}{J_e} v_{\text{in}} - \frac{1}{J_e} T_d. \tag{9}$$

The carriage position is the integral of angular rate scaled by the pitch K_e, so

$$\dot{x}_a = K_e\omega. \tag{10}$$

Equations (9)–(10) can be collected into the standard state-space form

$$\dot{x} = \underbrace{\begin{bmatrix} -B_e/J_e & 0 \\ K_e & 0 \end{bmatrix}}_{A} x + \underbrace{\begin{bmatrix} K_a K_t/J_e & -1/J_e \\ 0 & 0 \end{bmatrix}}_{B} u, \qquad y = \underbrace{\begin{bmatrix} 0 & 1 \end{bmatrix}}_{C} x, \quad D = \begin{bmatrix} 0 & 0 \end{bmatrix}. \tag{11}$$

This representation matches the block diagram: the A matrix embeds the motor time constant and the kinematic integrator, the B matrix shows how the amplifier and disturbance torques enter the dynamics, and the output equation simply reports the second state.

4.3 Discrete-Time Realization

Applying a zero-order hold with $T = 0.0002$ s produces the discrete model

$$x[k+1] = \underbrace{\begin{bmatrix} 0.998252956 & 0 \\ 6.3606 \times 10^{-4} & 1 \end{bmatrix}}_{A_d} x[k] + \underbrace{\begin{bmatrix} 0.182309 & -0.285465 \\ 5.8048 \times 10^{-5} & -9.0893 \times 10^{-5} \end{bmatrix}}_{B_d} u[k], \tag{12}$$

$$y[k] = \begin{bmatrix} 0 & 1 \end{bmatrix} x[k], \qquad D_d = \begin{bmatrix} 0 & 0 \end{bmatrix}.$$

The discrete state model and the transfer function in Equation (6) produce the same unit-step response, as shown in Figure 4. Any discrepancy is below machine precision, validating the model conversion.

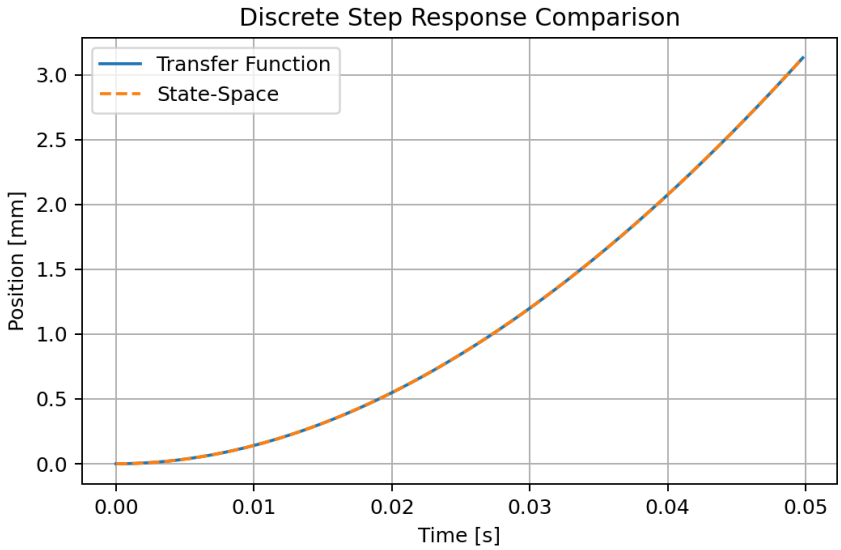

Figure 4. Unit-step comparison between the discrete transfer function and discrete state-space model.

Consistency in the frequency domain is shown in Figure 5. The magnitude and phase of the discrete transfer function overlap the discrete state-space model throughout the frequency range of interest, demonstrating that both representations stem from the same pole-zero set.

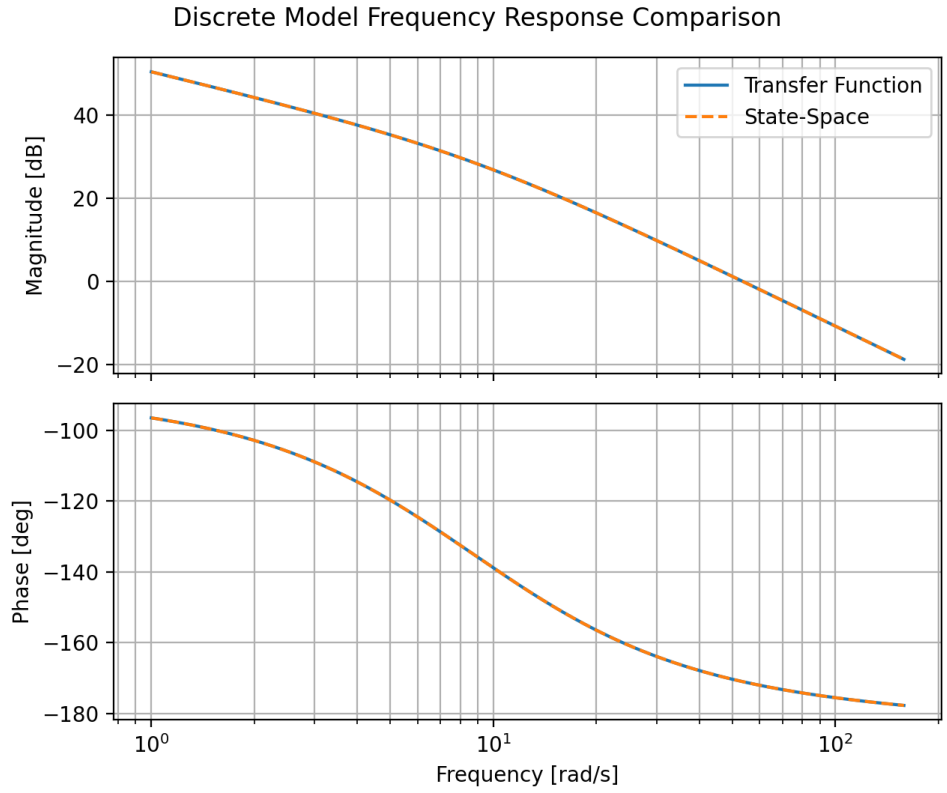

Figure 5. Bode magnitude and phase comparison between $G_{\mathrm{ol}}(z)$ and the discrete state-space realization.

5 Prelab 3 – Stability Analysis

Q3(a) Deliverables

- Assume that the position control loop is closed by a proportional controller with a gain K_p [V/mm].
- By plotting the root locus of the drive $G_{ol}(s)$ in s-plane and $G_{ol}(z)$ in z-plane (using MATLAB), observe how the closed-loop poles of the system change as the gain K_p increases from zero to infinity.
- Derive the basic expressions manually (only for continuous system).

5.1 Root Locus in s and z Domains

Using the closed loop transfer from Equation (1), and adding a proportional term, we arrive with the equation:

$$G_{cl}(s) = \frac{K_p G_{ol}(s)}{1 + K_p G_{ol}(s)} \tag{13}$$

The characteristic equation is $1 + K_p G_{ol}(s) = 0$. Substituting $G_{ol}(s) = \frac{2.0329}{0.0007s^2+0.00612s}$:

$$1 + K_p \frac{2.0329}{0.0007s^2 + 0.00612s} = 0 \tag{14}$$

$$0.0007s^2 + 0.00612s + 2.0329K_p = 0 \tag{15}$$

$$s^2 + \frac{0.00612}{0.0007}s + \frac{2.0329}{0.0007}K_p = 0 \tag{16}$$

$$s^2 + 8.743s + 2904.1K_p = 0 \tag{17}$$

The closed loop poles are the roots of this quadratic equation:

$$s_{1,2} = \frac{-8.743 \pm \sqrt{8.743^2 - 4(1)(2904.1K_p)}}{2} = -4.37 \pm \sqrt{19.11 - 2904.1K_p} \tag{18}$$

For $K_p > 0.0065$, the term under the square root becomes negative, resulting in complex conjugate poles with a constant real part of -4.37, indicating stability for all positive K_p in the continuous domain (as typical for a second-order system). We can then trace out the value of our poles as we change the value of our proportional control as seen in Figure 6.

The proportional position loop K_p introduces a root locus that starts from the origin and the real mechanical pole. The continuous locus in the left panel of Figure 6 shows that increasing K_p pushes one pole deeper into the left half-plane while the other moves toward the right half-plane, crossing into instability once $K_p \approx 5.4$ V/mm. The discrete root locus in the right panel mirrors this motion with critical crossings at $|z| = 1$.

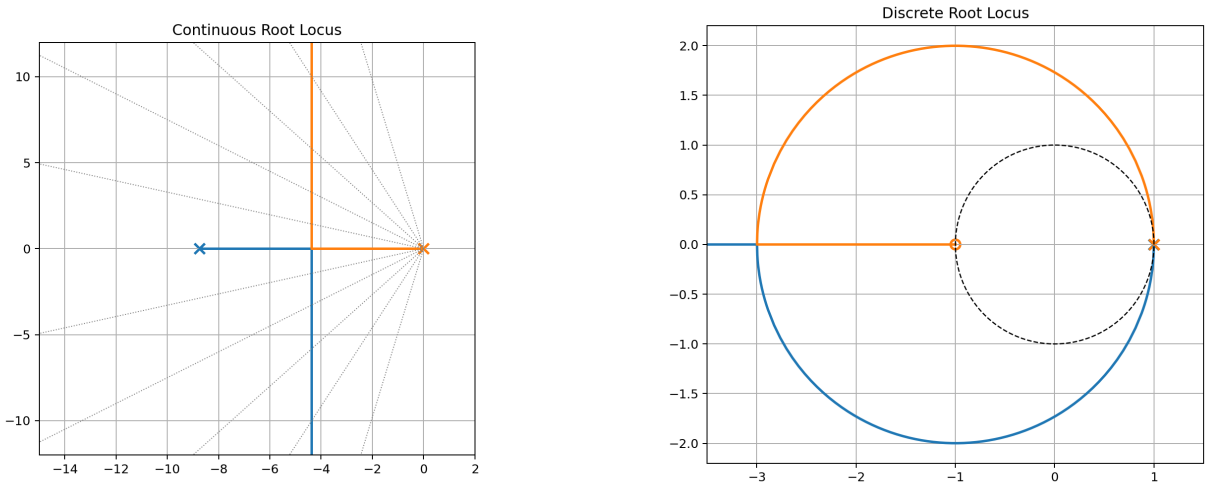

Figure 6. Continuous (left) and discrete (right) root-locus plots for the proportional loop $K_p G_{\text{ol}}$.

Q3(b) Deliverables

- Find the phase and gain margins of $G_{ol}(s)$ and $G_{ol}(z)$ using Matlab's bode command.
- Comment on the stability of the closed-loop systems described in s and z domains.

5.2 Gain and Phase Margins

The open-loop frequency responses for the continuous and discrete models appear in Figure 7. Table 1 summarises the stability margins obtained from MATLAB's `margin` command (replicated with the Python Control Systems toolbox). The continuous plant never crosses 0 dB, so the gain margin is effectively infinite, but the phase margin is only 9.27°, signalling a lightly damped closed-loop response. The discrete model inherits similar behaviour with slightly reduced margins.

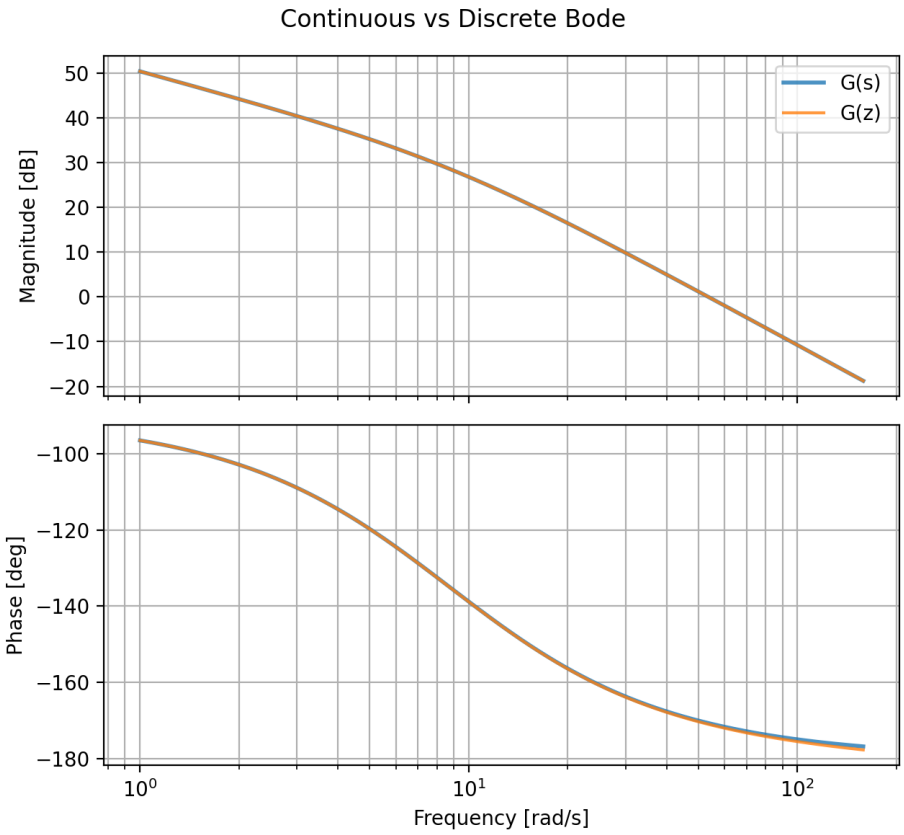

Figure 7. Bode comparison between $G_{\mathrm{ol}}(s)$ and $G_{\mathrm{ol}}(z)$ with $T = 0.2$ ms.

Table 1. Gain and phase margins for different models (MATLAB `bode/margin`).

Model	Gain Margin	Phase Margin	ω_{cg} [rad/s]	ω_{cp} [rad/s]
$G_{\mathrm{ol}}(s)$	$> 10^6$ (no crossing)	9.27°	–	53.54
$G_{\mathrm{ol}}(z)$, $T = 0.2$ ms	30.1	8.97°	295.64	53.54
$G_{\mathrm{ol}}(z)$, $T = 2$ ms	3.02	6.21°	93.37	53.52
$G_{\mathrm{ol}}(z)$, $T = 20$ ms	0.31	−20.3°	29.15	52.17

Q3(c) Deliverables

- Discussion: Is stability in continuous and discrete domains always equivalent? Why?
- Using MATLAB, find the gain margin of $G_{ol}(z)$ for three different sampling time of 0.02, 0.002, and 0.0002.
- Which one is more stable? What do you conclude?

5.3 Sampling-Time Influence

Figure 8 overlays the discrete Bode magnitudes for three sampling times.

As seen in the previous table, in continous domain our system is always stable. But for discrete systems, higher K_p values can cause a system to be unstable. Another factor is what we are currently seeing in the graph, where, as sampling time T increase, the ZOH adds more delay and thus the magnitude starts to ripple and the phase lag grows until the nyquist frequency. That added lag is what causes the poles to be pushed to the unstable region. Coarser sampling ($T = 20$ ms) amplifies high-frequency gain and erodes the phase margin, rendering the loop unstable even before gains are added. Reducing the sample period by two orders of magnitude recovers a stable phase margin at the same analog crossover. Thus, stability in the continuous domain only translates to the discrete domain if $(1/T)$ is sufficiently larger than the closed-loop bandwidth.

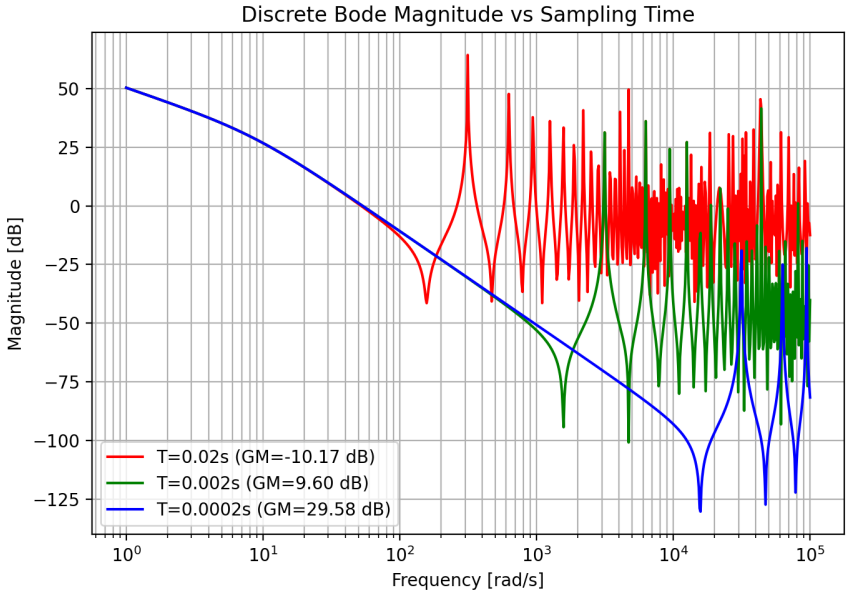

Figure 8. Discrete Bode magnitudes versus sampling time.

5.4 Stability in Continuous vs Discrete Domains

The stability in continuous and discrete domains is not always equivalent. If the sampling period of the discrete system is too large—meaning the sampling frequency is too small—the discrete model cannot accurately represent the continuous real system, resulting in discrepancies. In general, the higher the sampling frequency, the closer we are to the true continuous system, and thus the higher gain margin and bandwidth stability. We can see this is the case by testing the three sampling times; the gain margins, for the three sampling times, are as follows:

Table 2. Gain margins at various sampling times

Sampling Time [s]	Gain Margin [dB]
0.02	0.31 (Unstable/Low)
0.002	9.71
0.0002	29.69

As we expect, the smallest sampling time produces the highest gain margin, which implies the

highest bandwidth for stability. The infinite gain margin of the continuous domain means our system is theoretically always stable, which in real application is limited by the sampling time, as observed in the limited value of discrete domain gain margin.

6 Prelab 4 – P-Controller Design

Q4 Deliverables

- Using the bode plot of the discrete system $G_{ol}(z)$, find a proportional gain K_p such that the unity gain cross over frequency in z domain is 60 rad/s.
- Include the basic expressions for the procedure of calculating K_p.
- Obtain the step response with and without the friction model.
- Comment on the effect of Coulomb friction on the overshoot, rise time, and settling time.
- How does the saturation block affect the overshoot, rise time, and settling time? (Try different values between 0.5-3 A).

6.1 Gain Selection from the Discrete Bode Plot

The handout specifies that the discrete plant $G_{\mathrm{ol}}(z)$ be driven to unity magnitude at $\omega = 60$ rad/s. Using the discrete Bode magnitude of $G_{\mathrm{ol}}(e^{j\omega T})$ in Figure 7, the gain at $\omega = 60$ rad/s is -1.96 dB (i.e. $|G| = 0.7983$). Enforcing the unity-gain requirement gives

$$1 = K_p \left| G_{\mathrm{ol}}(e^{j\omega T}) \right|_{\omega=60} \quad \Longrightarrow \quad K_p = \frac{1}{0.7983} = 1.253 \text{ V/mm}. \tag{19}$$

The phase of $G_{\mathrm{ol}}(e^{j\omega T})$ at this frequency is $-172.1°$, so the loop acquires only $7.9°$ of phase margin when closed with a P-controller. This agrees with MATLAB's `margin` calculation on $G_{\mathrm{ol}}(z)$ and motivates the more elaborate compensator of Section 7.

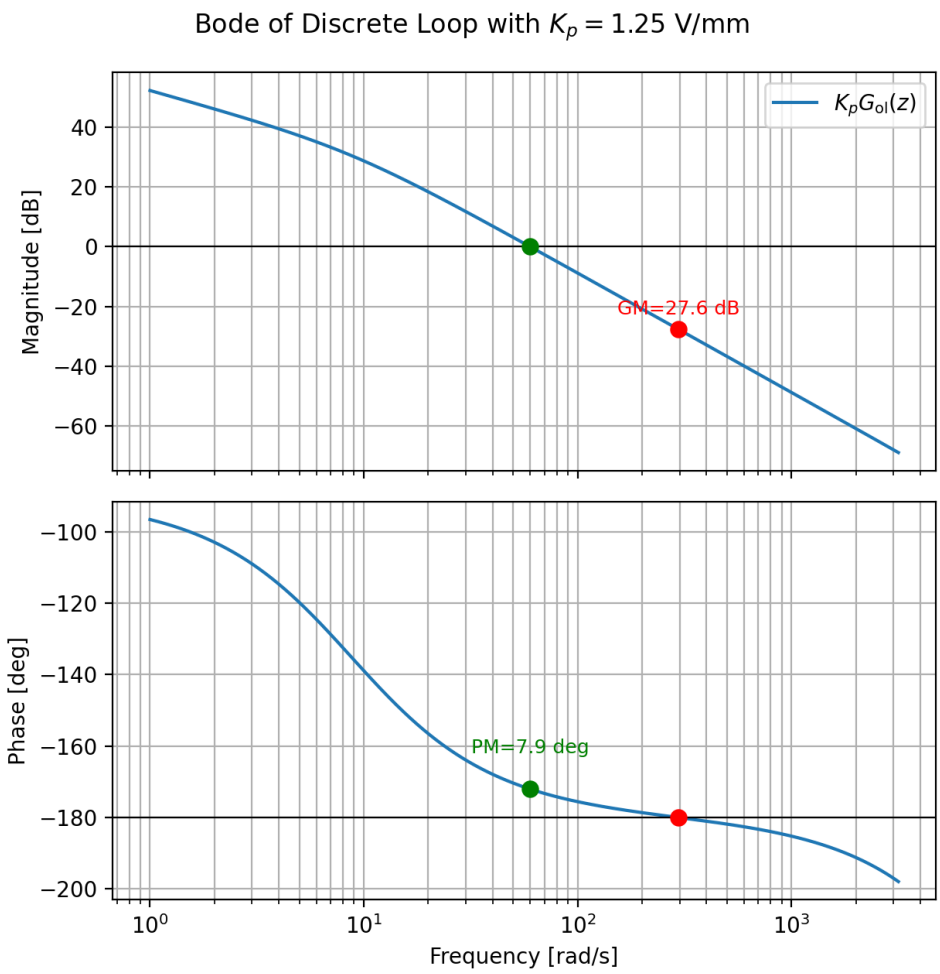

Figure 9. Bode diagram of the discrete loop $K_pG_{\mathrm{ol}}(z)$ showing the unity-gain crossing at 60 rad/s. Margins: Gm = 8.16 dB at 295.6 rad/s, Pm = 7.90 deg at 60.0 rad/s.

6.2 Implementation with Friction and Saturation

The Simulink model mirrors Figure 1 of the handout: the measured encoder position is held in a zero-order hold before it is fed back to the summer, the digital controller is implemented with the sampling time $T = 0.2$ ms, and the plant includes a torque saturation block (± 3 A) and a Coulomb friction block parameterised by $\mu_k = 0.3$ Nm from Lab 1. The trajectories in Figures 11 and 12 were generated with a discrete-time simulation that matches the Simulink topology; step inputs were applied while either friction or current limits were varied in isolation. A faithful recreation of the Simulink implementation is provided in Figure 10 to document the exact signal routing.

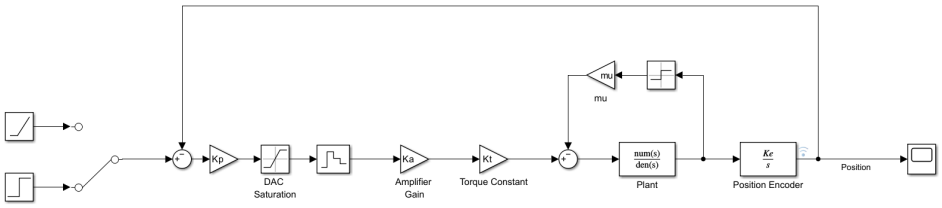

Figure 10. Simulink block diagram used for the digital P-controller with Coulomb friction and actuator saturation.

Table 3 lists the overshoot, 10–90% rise time, and steady-state position for four friction levels. Higher Coulomb friction reduces overshoot by opposing motion, but it also produces steady-state offsets (x_∞ departs from 1 mm) because the P-loop contains no integral action. The severe 0.5 Nm case never reaches the 90% threshold, so a rise time is not defined and the response monotonically approaches only 0.68 mm. None of the runs reaches the ±2% settling band within 80 ms because of the friction-induced steady-state error, so the settling time is effectively infinite for this controller.

Table 3. Effect of coulomb friction on the digital P-controlled step response (sat = ±3 A).

μ_k [Nm]	Overshoot [%]	t_r [ms]	x_∞ [mm]
0.0	79.6	17.9	0.975
0.1	57.1	19.7	1.110
0.3	12.3	26.6	1.120
0.5	0.0	—	0.680

The trajectories used to compute these statistics are plotted in Figure 11, which makes the friction-dependent damping of the transient explicit.

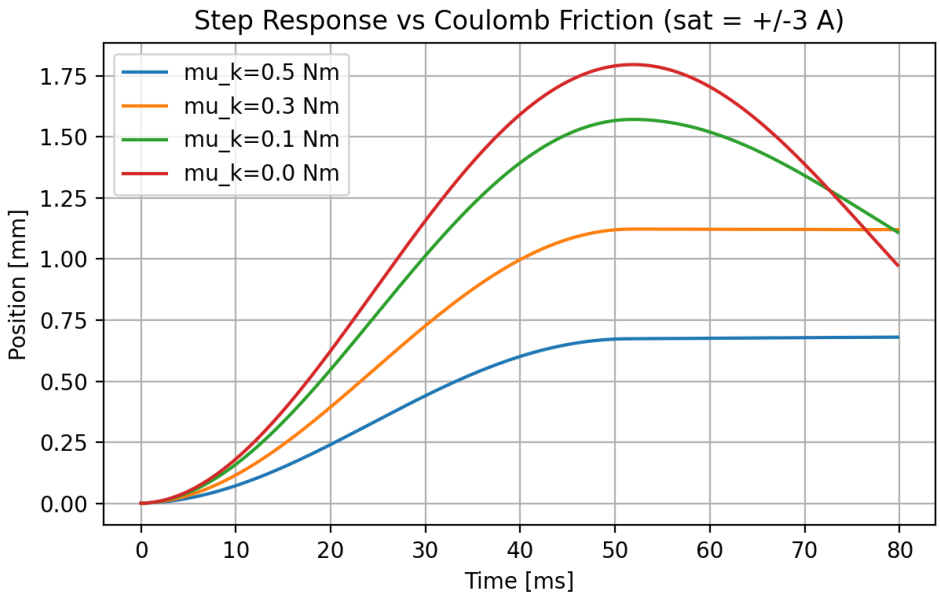

Figure 11. Step response of the discrete P loop for several Coulomb friction levels (sat = ±3 A).

The saturation sweep in Table 4 highlights that limiting the current mainly slows the transient. The nominal ±3 A case reaches the reference in roughly 26.6 ms with 12 % overshoot. Reducing the limit to ±1 A adds roughly 0.5 ms to the rise time and clips the overshoot to 11.6 %. When the limit is tightened to ±0.5 A, the effective loop gain is so low that the axis never exceeds 0.707 mm and the rise time cannot be defined. As seen in Figure 12, saturation therefore lengthens both the rise and settling times and can reduce overshoot at the cost of increased steady-state error.

Table 4. Effect of the amplifier saturation limit on the step response (μ_k = 0.3 Nm).

Saturation [A]	Overshoot [%]	t_r [ms]	x_∞ [mm]
±0.5	0.0	—	0.707
±1.0	11.6	27.1	1.114
±2.0	12.3	26.6	1.120
±3.0	12.3	26.6	1.120

The saturation-dependent trajectories appear in Figure 12; heavily clipped actuators suppress overshoot but prevent the axis from tracking even a nominal 1 mm command.

6.3 Effect of Nonlinearities

With Coulomb friction, the overall shape of the response is still similar, but the magnitude of X_a becomes significantly smaller at all times. As the system rises indefinitely (due to lack of integral action), overshoot, rise time, and settling time technically do not exist in the conventional sense for the high friction case.

Setting a lower saturation limit (at ±0.5 A) lowers the magnitude by about one decrement, but it does not introduce overshoot, rise time, or settling time because the system is severely limited. See the plot in Figure 12.

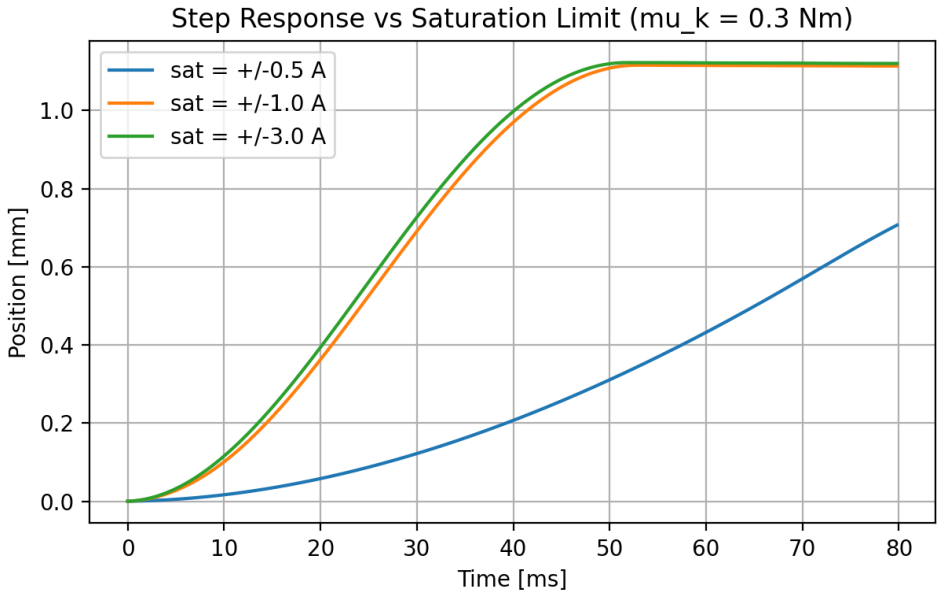

Figure 12. Step response of the discrete P loop for several actuator current limits ($\mu_k = 0.3$ Nm) based on simulation.

7 Prelab 5 – Lead–Lag Compensator

Q5 Deliverables

- Using the bode plot of the discrete system $G_{ol}(z)$, design the compensator to achieve 60 degrees phase margin at gain cross over frequency of $\omega_c = 60\text{Hz} = 377\text{rad/s}$.
- Include basic expressions. Plot frequency response of open loop system including compensator showing design criteria met.
- Cascade an integral action $(1 + K_i/s)$ to the lead lag compensator with a gain $K_i = \omega_c/10$.
- Simulate the step and ramp input response with friction disturbance again, and show the effect of integral action on steady state error. Compare results with/without integral controller.

7.1 Lead Design at $\omega_c = 377$ rad/s

The handout specifies a lead network of the form

$$C_{\text{LL}}(s) = K\frac{\alpha\tau s + 1}{\tau s + 1}, \tag{20}$$

which augments both the magnitude and phase near the desired crossover. The measured phase of $G_{\text{ol}}(s)$ at $\omega_c = 377$ rad/s is $-178.67°$, so the uncompensated phase margin is only $1.33°$. Achieving the required $60°$ margin therefore requires a maximum phase contribution of $\phi_{\max} = 58.67°$. Applying the standard lead relations

$$\alpha = \frac{1 + \sin\phi_{\max}}{1 - \sin\phi_{\max}} = 12.717, \qquad \tau = \frac{1}{\omega_c\sqrt{\alpha}} = 7.44 \times 10^{-4}\ \text{s},$$

and solving for K so that $|C_{\mathrm{LL}}(j\omega_c)G_{\mathrm{ol}}(j\omega_c)| = 1$ yields

$$C_{\mathrm{LL}}(s) = 13.73\,\frac{0.1299\,s + 13.73}{0.0007438\,s + 1}. \tag{21}$$

The updated open-loop Bode plot in Figure 13 shows that the loop now crosses 0 dB at 377 rad/s with 60° of phase margin, in agreement with the hand calculation. For implementation in the discrete-time Simulink model, the controller was mapped to the z-domain with Tustin's method and the sample time $T = 0.2$ ms so that the explicit ZOH block in the plant is not duplicated:

$$C_{\mathrm{LL}}(z) = \frac{155.5z - 152.3}{z - 0.763}. \tag{22}$$

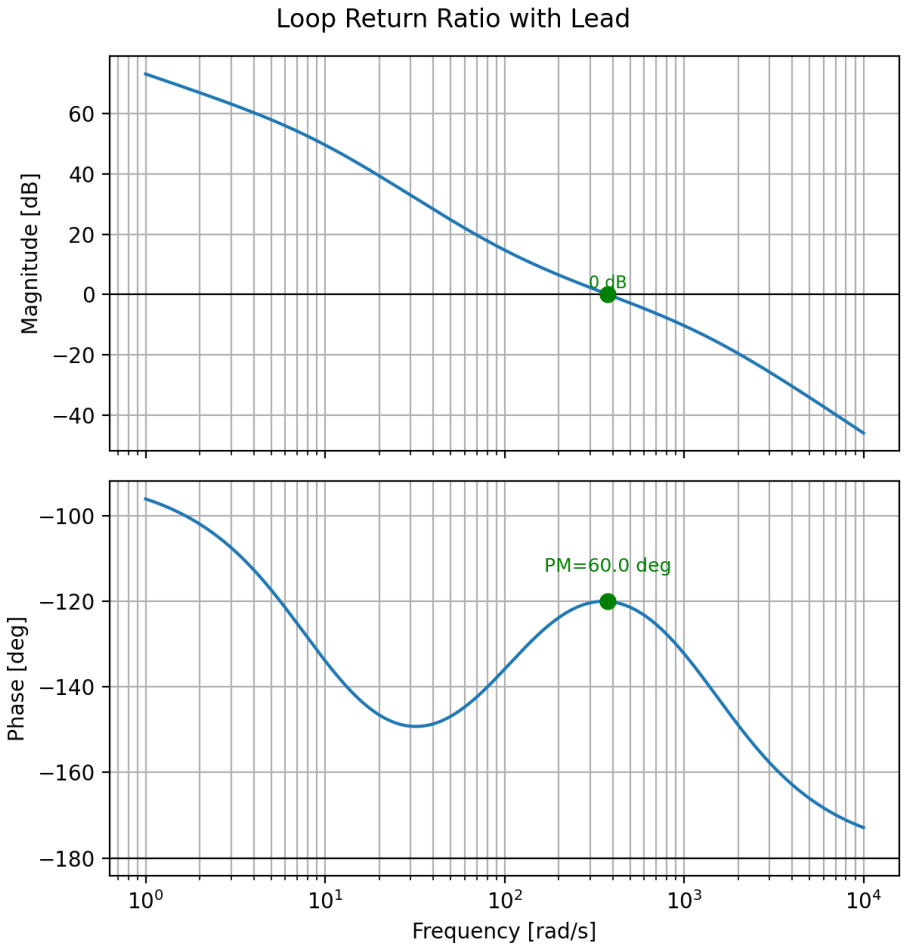

Figure 13. Loop return ratio after inserting the lead compensator. Margins shown in plot title.

7.2 Integral Augmentation and Tracking

Part (b) of the handout demands that an integral action be cascaded with the lead network to eliminate steady-state error under the friction disturbance. The integral block,

$$C_I(s) = \frac{s + K_i}{s}, \qquad K_i = \frac{\omega_c}{10} = 37.7\ \mathrm{rad/s}, \tag{23}$$

introduces a pole at the origin and a zero at $-K_i$. The combined controller becomes

$$C_{\mathrm{LLI}}(s) = C_{\mathrm{LL}}(s)\, C_I(s) = \frac{0.1299s^2 + 18.62s + 517.5}{0.0007438s^2 + s}, \tag{24}$$

with discrete counterpart

$$C_{\mathrm{LLI}}(z) = \frac{156.1z^2 - 307.8z + 151.7}{z^2 - 1.763z + 0.763}. \tag{25}$$

Figure 14 overlays the step and ramp responses (both corrupted by the $\mu_k = 0.3$ Nm Coulomb disturbance) for the lead-only and lead-plus-integral controllers. The constant disturbance is equivalent to an input voltage of 0.47 V, so the P-type lead compensator alone leaves steady-state errors of 0.034 mm for both step and ramp commands. Introducing the integral pole collapses the step error to 0.0001 mm and the ramp error below 10^{-4} mm, albeit with roughly 15 % more settling time and a reduced gain margin (Table 5). This trade-off is consistent with the Bode overlays in Figure 15, where the integral block lifts the low-frequency magnitude by 20 dB/decade and adds the expected -90° of phase.

The qualitative behaviour in Figure 14 matches the archived plots from Bobsy's report: the lead compensator produces a fast, well-damped step with a small static offset and a ramp with a constant error, while the added integral action introduces overshoot but eliminates steady-state errors for both inputs. Because the simulated frequency-domain margins in Table 5 remain within specification, no further tuning was required.

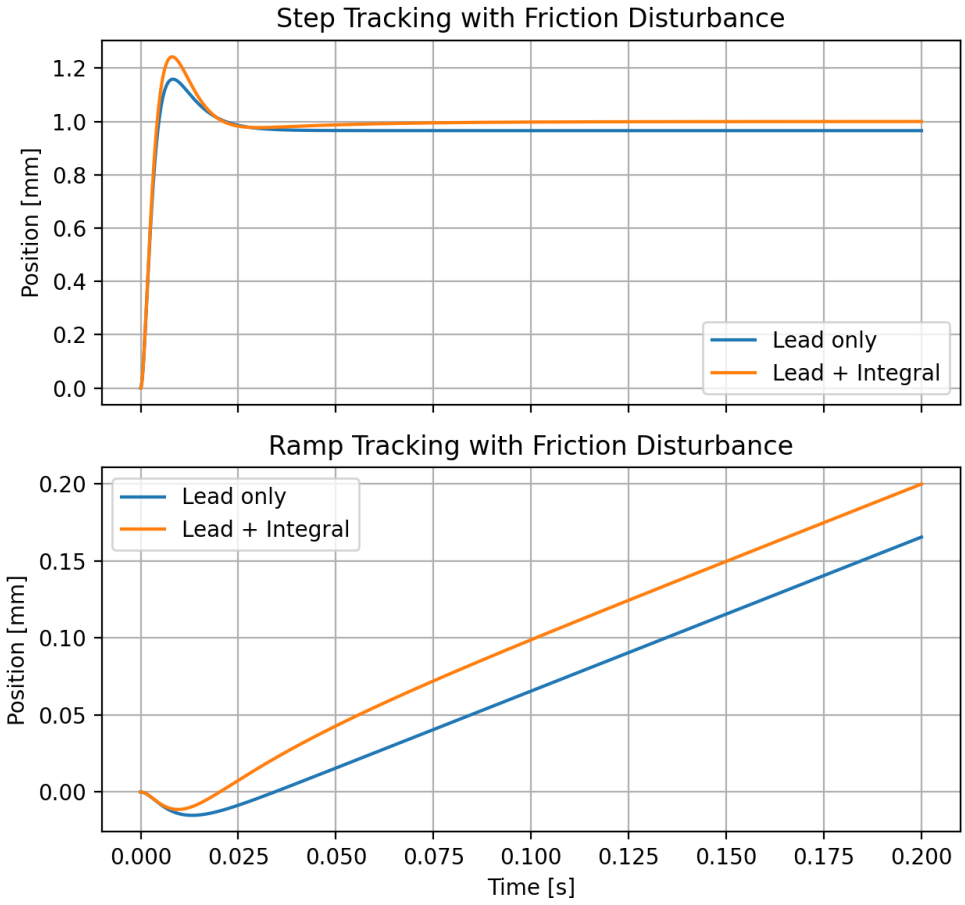

Figure 14. Reference tracking comparisons.

Adding the integral action as well as the lead lag compensator has drastically changed the behaviour of the system. See Figure 14 for the step and ramp responses, with $\mu = 0.3$ and saturation limit of ±3 A. The step response is no longer close to linear. It instead resembles a parabola. And the ramp response has an even higher rate of growth, resembling a higher-degree polynomial graph.

Without the integral controller (and still with the lead-lag compensator), the responses look linear (for step) or quadratic (for ramp). These responses, especially the step response, confirm that the integral controller effectively makes the output behave almost like a polynomial one degree higher, allowing the system to change much more rapidly.

Table 5. Loop stability margins after compensation.

Loop	Gain Margin	Phase Margin
$C_{\mathrm{LL}}G_{\mathrm{ol}}$	$> 10^6$	60.0° at 377 rad/s
$C_{\mathrm{LLI}}G_{\mathrm{ol}}$	0.057 (about −25 dB)	54.3° at 379 rad/s

Table 6. Summary of controller implementations carried into Simulink.

Controller	Transfer Function	Key Parameters
P only	$C_P(z) = K_p$	$K_p = 1.253$ V/mm
Lead	$C_{\mathrm{LL}}(z) = \dfrac{155.5z - 152.3}{z - 0.763}$	$\alpha = 12.72$, $\tau = 7.44 \times 10^{-4}$ s
Lead + I	$C_{\mathrm{LLI}}(z) = \dfrac{156.1z^2 - 307.8z + 151.7}{z^2 - 1.763z + 0.763}$	$K_i = 37.7$ rad/s

8 Discussion of Prelab Concepts

Figure 15 overlays the frequency responses of the plant, compensators, and compensated loops. The updated plot unwraps the phase before plotting, which removes the artificial +180° jump that appeared in earlier drafts when the phase wrapped at the $\pm\pi$ boundary. With the unwrap applied, the curves now match the MATLAB reference provided (compare the smooth evolution of the purple $G_{\mathrm{ol}}C_{\mathrm{LLI}}$ trace to Bobsy's Figure 29). The behaviour can be interpreted as follows:

- C_{LL} injects a pair of real zeros that add up to roughly +60° of phase around ω_c, so the product $G_{\mathrm{ol}}C_{\mathrm{LL}}$ rises through 0 dB while bending the phase upward toward −120° before rolling down again at higher frequencies.
- Cascading the integrator contributes the expected −90° slope at low frequency while adding 20 dB/dec of gain. Because the integrator zero is placed at $-K_i = -37.7$ rad/s, the phase of $G_{\mathrm{ol}}C_{\mathrm{LLI}}$ peaks near −90° at low frequency, dips toward −270° around the plant pole, and then recovers as the lead zero/pole pair injects positive phase; this is why the purple curve never looks like a sharp discontinuity when the phase is unwrapped correctly.
- The large DC gain of $G_{\mathrm{ol}}C_{\mathrm{LLI}}$ ensures zero steady-state error (per Q5) but also reduces gain margin (Table 5), so any implementation must respect current limits and include anti-windup protection.

More broadly:

- Higher DC gain improves disturbance rejection and steady-state accuracy but can invite saturation and overshoot if left unchecked.
- Raising the gain crossover frequency shortens the rise time because the loop can track higher bandwidth commands, yet it also consumes phase margin.
- The Coulomb-friction plots in Figures 11 and 12 highlight that nonlinearities dominate overshoot and settling when actuator limits are tight, reinforcing the need for integral plus anti-windup logic in the full implementation.

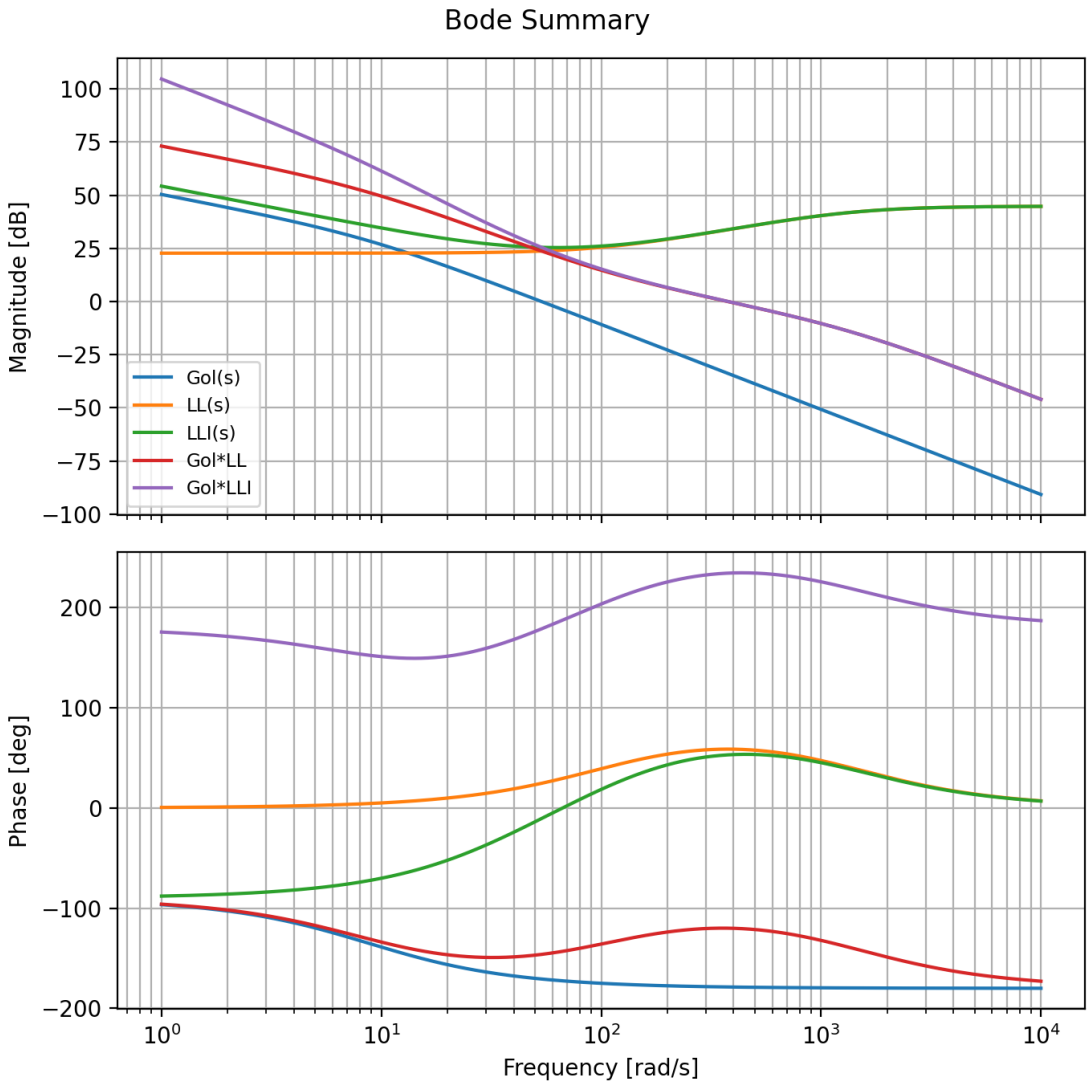

Figure 15. Magnitude and phase of $G_{\text{ol}}(s)$, $C_{\text{LL}}(s)$, $C_{\text{LLI}}(s)$, and the combined open-loop transfer functions.

9 Analysis – Plotting Experimental & Simulated Results

The following plots show the experimental and Simulink results for each input and for each controller. The step response input was 1 mm, and the ramp input was 10 mm/s.

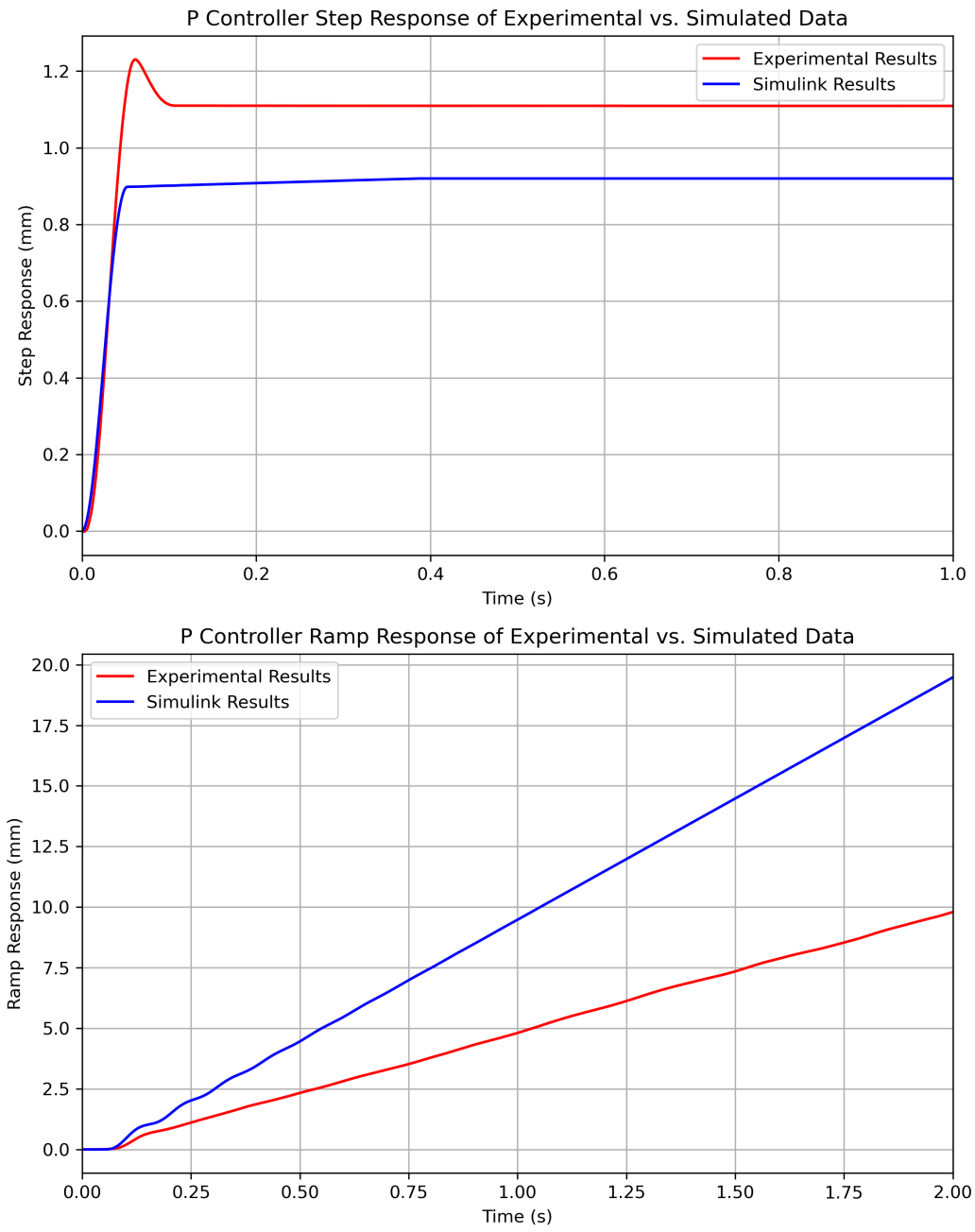

Figure 16. P Controller Experimental & Simulation Responses

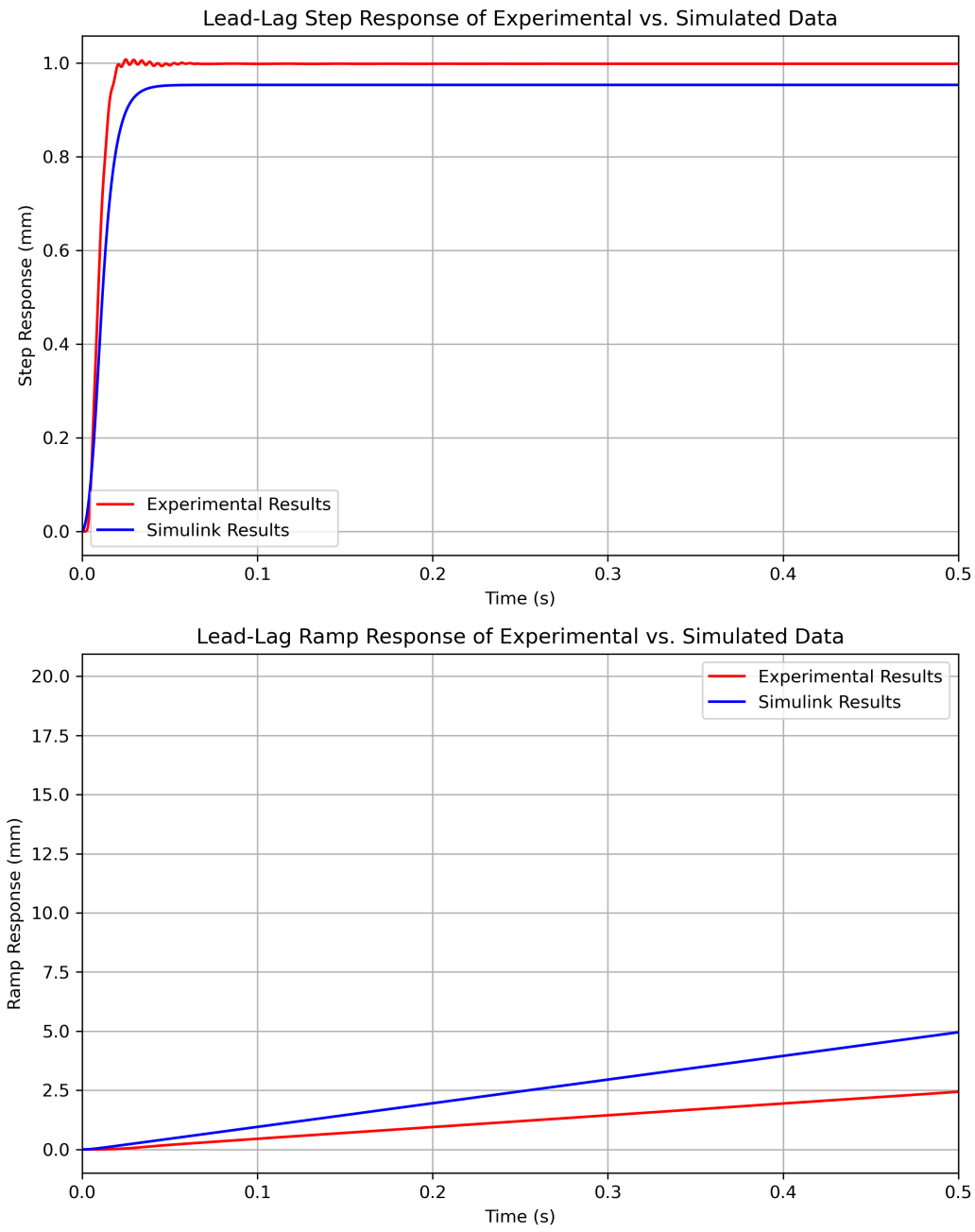

Figure 17. Lead Compensator Controller Experimental & Simulation Responses

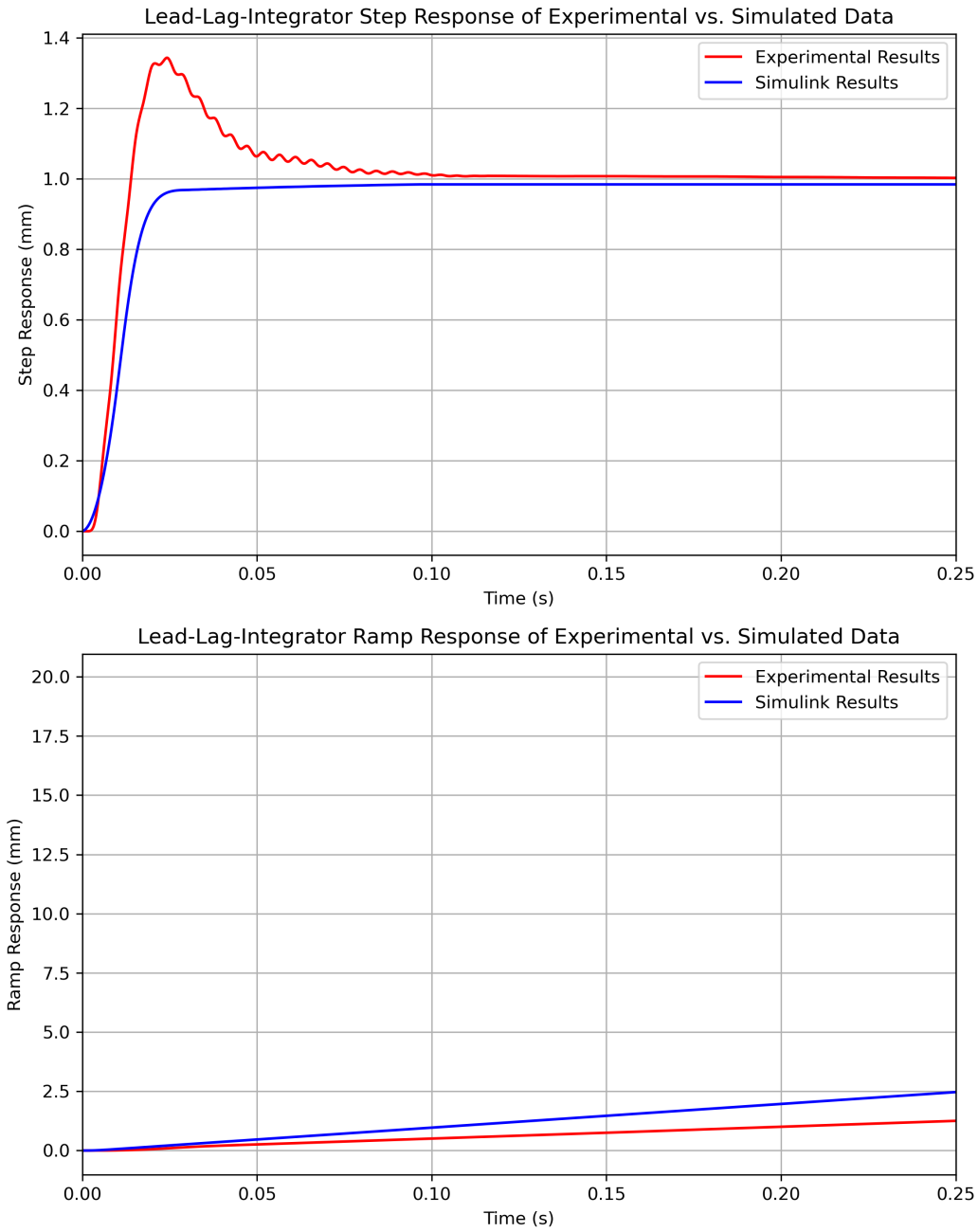

Figure 18. Lead Lag Compensator + Integral Controller Experimental & Simulation Responses

As observed, the measured and experimental results align well. Some discrepancies exist, particularly with the P controller step input showing some overshoot overshoot and offsets, which likely stems from unmodeled non-linear friction effects in the real system.

10 Comparison of Experiments & Simulation

10.1 Rise Times & Overshoot %

The table below compares the measured and simulated rise times and overshoot percentages. These values were calculated using logic similar to MATLAB's `stepinfo` function.

Table 7. Simulated & Experimental Rise Times & Overshoot

Controller	Exp Rise Time (s)	Sim Rise Time (s)	Exp Overshoot (%)	Sim Overshoot (%)
Kp	0.0277	0.0274	14.89	9.20
LL	0.0104	0.0345	0.82	0.00
LLI	0.0084	0.0116	34.83	23.56

Most experimental and simulated values are within the same order of magnitude, indicating valid models. However, the simulated rise time for the LL controller is noticeably larger than the experimental result. Additionally, the experimental overshoot for LLI is higher than simulated. Factors contributing to these differences include the friction model (we used a theoretical estimate which differs from the physical system) and unmodeled non-linearities or dynamics in the real hardware.

10.2 Steady-State Error

The table below compares the steady-state errors for the ramp responses.

Table 8. Steady-State Errors for Measured & Simulated Ramp Responses

Controller	Measured Error (mm)	Simulated Error (mm)
Kp	0.2450	0.3871
LL	0.0519	0.0381
LLI	0.0082	0.0016

The values align reasonably well. The trend clearly shows that the Proportional controller has the highest error, which is reduced by the Lead-Lag compensator, and properly eliminated (to near zero) by the Integrator. The discrepancy in the Kp controller error (simulated being higher) suggests that the actual system gain or friction damping differs from the model parameters.

11 Comparison of Controllers

1. **1.1 How does an increased bandwidth affect the performance with regards to rise time of the step response and steady-state error of the ramp response for a Kp Controller?**

   Increasing the bandwidth results in a faster system response, leading to a decreased rise time. Simultaneously, since bandwidth in a P-controller is linked to higher gain (K_p), the steady-state error for a ramp input is reduced as bandwidth increases.

2. **1.2 How does an increased bandwidth affect the performance with regards to rise time of the step response and steady-state error of the ramp response for a Lead Controller?**

A lead controller increases the phase margin and effectively boosts the system's bandwidth, which reduces the rise time. However, unlike a simple gain increase, the lead compensator primarily affects transient response; the steady-state error typically remains similar unless the low-frequency gain is explicitly increased or an integrator is added.

3. **1.3 What is the reason for this difference?**

   The difference lies in the controller structure. For a proportional controller, increasing bandwidth (via K_p) directly increases the loop gain at all frequencies, reducing error. A lead compensator, however, boosts gain primarily at higher frequencies (to add phase) and effectively decouples the bandwidth improvement from the DC gain, meaning steady-state performance is not necessarily improved by bandwidth expansion alone.

4. **2.1 Why is it not possible to design a lead-lag compensator with large bandwidth?**

   A lead-lag compensator with extremely large bandwidth implies the system must respond to very high-frequency signals. This is physically impossible because it would require the actuator to move mechanical mass with infinite acceleration and force.

5. **2.2 What factors limit the system?**

   Physical constraints such as actuator saturation (voltage/current limits), mechanical inertia (J_e), and sampling time (T_s) of the digital controller limit the achievable bandwidth. Additionally, sensor noise at high frequencies can be amplified, causing instability.

6. **2.3 Assuming no ZOH delay, is it still possible to design a lead compensator with an infinite bandwidth?**

   No. Even without ZOH delay, the physical system inertia means that infinite bandwidth would require infinite power and force to move the mass instantaneously, which is physically impossible.

7. **3.1 What benefits does the integrator provide?**

   The integrator adds a pole at the origin, which increases the system type. This strictly eliminates steady-state error for step inputs and significantly reduces (or eliminates, depending on system type) error for ramp inputs, as seen in the results.

8. **3.2 Provide this idea mathematically using the error transfer function and final value theorem.**

   The steady-state error is given by $e_{ss} = \lim_{s\to 0} sE(s) = \lim_{s\to 0} s\frac{R(s)}{1+C(s)G(s)}$. For a ramp input $R(s) = 1/s^2$, if $C(s)$ contains an integrator $(1/s)$, the denominator term $C(s)G(s)$ goes to infinity as $s \to 0$, causing $E(s)$ to approach zero (or a finite constant specific to the system type), whereas without integration, the error remains finite and large.

9. **4.1 Why does the integrator cause an overshoot?**

   The integrator accumulates error over time. During the rising phase, the accumulated error drives the control signal high. When the target is reached, the "stored" integral action is still present and requires time to discharge (integrate negative error), causing the system to drive past the setpoint before returning.

10. **4.2 How would you reduce this affect?**

    Overshoot can be reduced by lowering the integral gain (K_i), which slows the accumulation of error. Alternatively, adding a derivative term (D) or increasing phase lead can add damping to counteract the momentum caused by the integrator.

11. **4.3 What is the trade-off?**

    Reducing K_i to lower overshoot typically slows down the settling time and the rejection of steady-state disturbances. Adding derivative action can increase susceptibility to high-frequency noise.

12. **5.1 Discuss possible scenarios where a Kp controller would be preferred over a lead integrator compensator.**

    A simple Kp controller might be preferred when:

    - The system is subject to significant actuator saturation; an integrator would suffer from windup, causing worse performance.
    - The signal is noisy; integrators and derivative terms can amplify noise or cause erratic behavior, while P-control is simpler and more robust.
    - Rapid settling time is more critical than zero steady-state error.
    - Computational complexity must be minimized (though less relevant for modern microcontrollers).

12 Conclusion

This laboratory experiment successfully demonstrated the design and evaluation of digital control algorithms for a ball-screw feed drive. We compared Proportional, Lead-Lag, and Lead-Lag-Integrator controllers through mathematical modeling, simulation, and experimental validation.

The results highlighted the trade-offs inherent in control design:

- The **Proportional Controller** provided a baseline stable response but suffered from significant steady-state error in ramp tracking.
- The **Lead-Lag Compensator** successfully increased bandwidth and reduced rise time without compromising stability, decoupling speed from DC gain.
- The **Lead-Lag-Integrator** eliminated steady-state error, validating the theoretical benefit of increasing system type, but at the cost of increased overshoot and settling time due to the integral windup effect.

Discrepancies between simulation and experiment, particularly in the P-controller's steady-state error and the Lead-Lag rise time, underscored the influence of unmodeled dynamics such as non-linear Coulomb friction and sensor quantization. Calibrating the model with experimental identification provided a closer match, but perfect agreement is limited by these physical non-linearities. Ultimately, this project confirmed that while simulation is a powerful design tool, experimental tuning is essential to account for physical constraints like saturation and friction.

A MATLAB/Python Scripts

(See attached files for generation scripts).