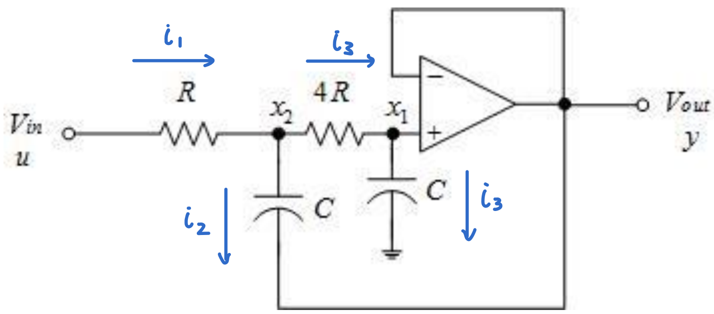
Project Tasks

1. Consider the overdamped second order circuit shown below:



$C = 10\mu F; R = 50k\Omega$

Determine the transfer function $H(s) = \dfrac{Y(s)}{U(s)}$ [1]

Using impedances to solve the problem:

Ideal op amp law: no current flow into op amp

$$\frac{V_{x_2} - V_{x_1}}{4R} = \frac{V_{x_1} - 0}{\frac{1}{Cs}} \Rightarrow V_{x_2} - V_{x_1} = 4RCsV_{x_1} \Rightarrow V_{x_2} = (1+4RCs)V_{x_1}$$

Using ideal op amp law for negative feedback: $V_{x_1} = V_{out}$

$$V_{x_2} = (1+4RCs)V_{out}$$

Using KCL:

$$i_1 = i_2 + i_3$$

$$\frac{V_{in} - V_{x_2}}{R} = \frac{V_{x_2} - V_{out}}{\frac{1}{Cs}} + \frac{V_{x_2} - V_{out}}{4R}$$

$$4V_{in} - 4V_{x_2} = 4RCsV_{x_2} - 4RCsV_{out} + V_{x_2} - V_{out}$$

$$4V_{in} = (5+4RCs)V_{x_2} + (-1-4RCs)V_{out}$$

$$4V_{in} = (5+4RCs)(1+4RCs)V_{out} + (-1-4RCs)V_{out}$$ } substituting for V_{x_2}

$$4V_{in} = V_{out}[(5+4RCs)(1+4RCs) - 1 - 4RCs]$$

$$4V_{in} = V_{out}[5 + 24RCs + 16R^2C^2s^2 - 1 - 4RCs]$$

$$4V_{in} = V_{out}[16R^2C^2s^2 + 20RCs + 4]$$

$$\frac{V_{out}}{V_{in}} = \frac{Y(s)}{U(s)} = \frac{1}{4R^2C^2s^2 + 5RCs + 1} = \boxed{\frac{1}{s^2 + 2.5s + 1} = H(s)}$$

[1] All derivations and design calculations should be inserted in the text boxes provided. Handwritten equations are acceptable. **Please indicate final answers with numerical numbers (instead of R and C).**

Determine the state space representation
$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$
$$y(t) = \mathbf{C}\mathbf{x}(t)$$

input: $V_{in} = u$ state: $\bar{x} = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$ so $\dot{\bar{x}} = \begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix}$

output: $V_{out} = y$

Inverse laplace of equations from previous part to get the differential equations to then put in state space form:

$$\mathcal{L}^{-1}[x_2 - x_1 = 4RCs\,x_1] \Longrightarrow x_2(t) - x_1(t) = 4RC\dot{x}_1(t)$$
$$\tfrac{1}{2}x_2(t) - \tfrac{1}{2}x_1(t) = \dot{x}_1(t)$$

$$\mathcal{L}^{-1}[4V_{in} - 4x_2 = 4RCs\,x_2 - 4RCs\,V_{out} + x_2 - V_{out}]$$ using $V_{out} = x_1$

$$4V_{in}(t) - 4x_2(t) = 2\dot{x}_2(t) - 2\dot{x}_1(t) + x_2(t) - x_1(t)$$
$$\dot{x}_2(t) = 2V_{in}(t) - \tfrac{5}{2}x_2(t) + \dot{x}_1(t) + \tfrac{1}{2}x_1(t)$$
$$\dot{x}_2(t) = 2V_{in}(t) - \tfrac{5}{2}x_2(t) + \tfrac{1}{2}x_2(t) - \tfrac{1}{2}x_1(t) + \tfrac{1}{2}x_1(t)$$
$$\dot{x}_2(t) = 2V_{in}(t) - 2x_2(t)$$

$$\boxed{\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} -1/2 & 1/2 \\ 0 & -2 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 2 \end{bmatrix} V_{in} \quad \text{and} \quad V_{out} = [1 \;\; 0]\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}}$$

Determine an analytical expression for the step response $y_{step}(t)$

The step response $y_{step}(t)$ is the output when the input is the step reference of 2.5 V.

$$V_{out} = \frac{1}{s^2 + 2.5s + 1} V_{in} \Longrightarrow Y_{step}(s) = \frac{1}{s^2 + 2.5s + 1} \cdot \frac{2.5}{s}$$

$\omega_n = 1$ $\quad 2\zeta\omega_n = 2.5$

$\zeta = 1.25 \longrightarrow$ overdamped!

Second order step response for overdamped system:

$$y(t) = \frac{\alpha y_f}{\omega_n^2}\left[1 + Ae^{-(\zeta - \sqrt{\zeta^2 - 1})\omega_n t} + Be^{-(\zeta + \sqrt{\zeta^2 - 1})\omega_n t}\right]$$

$\alpha = 1$, $y_f = 2.5$, $A = \dfrac{-\zeta - \sqrt{\zeta^2 - 1}}{2\sqrt{\zeta^2 - 1}} = \dfrac{-1.25 - \sqrt{1.5625 - 1}}{2\sqrt{1.5625 - 1}} = -1.333 = -\tfrac{4}{3}$

$B = \dfrac{\zeta - \sqrt{\zeta^2 - 1}}{2\sqrt{\zeta^2 - 1}} = \dfrac{1.25 - \sqrt{1.5625 - 1}}{2\sqrt{1.5625 - 1}} = 0.333 = \tfrac{1}{3}$

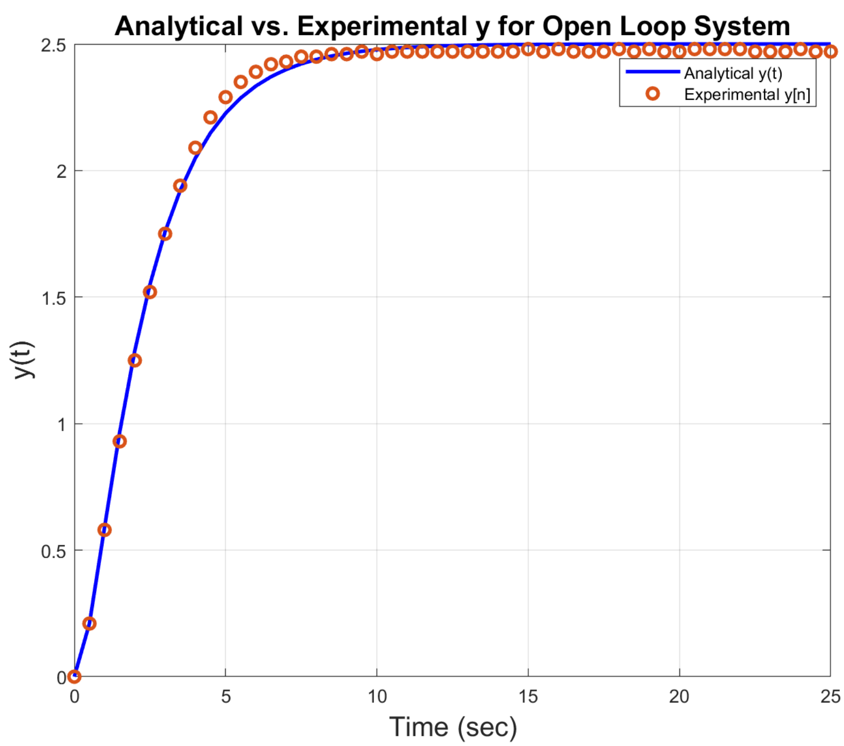
$$y(t) = 2.5\left[1 - \tfrac{4}{3}e^{-(1.25 - \sqrt{1.25^2 - 1})t} + \tfrac{1}{3}e^{-(1.25 + \sqrt{1.25^2 - 1})t}\right]$$

$$\boxed{y(t) = 2.5\left[1 - \tfrac{4}{3}e^{-0.5t} + \tfrac{1}{3}e^{-2t}\right]}$$

2. Build the circuit with the LMC6484 op amp (data sheet in class directory) and measure the open loop step response experimentally by coding the Arduino Uno microprocessor to interface with the circuit.

Insert a fully labeled experimental open loop step response plot for a 2.5V step input[2] with an overlay of the analytical step response from part 1

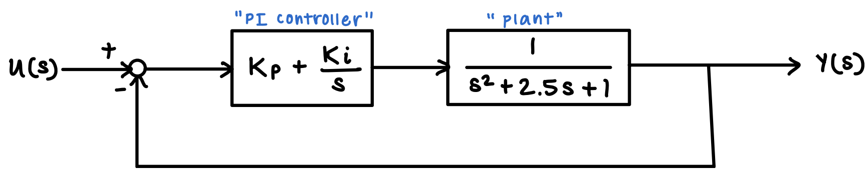
(insert figure here)



[2] All plot figures submitted in this workbook must be fully labeled with title, axis labels, and legends for multiple plots. The quality of the figures will be considered a part of the assessment. See the plot examples in the "Using the Arduino Uno as a digital controller" tutorial document

3. Design a unity feedback continuous-time closed loop system for the circuit (plant) with a PI compensator $C(s) = K_p + \frac{K_i}{s}$ for 70° phase margin and crossover frequency $\omega_{co} = 1\, rad/s$.

Unity feedback closed loop system with PI compensator:



$$T_{OL}(s) = \left(K_p + \frac{K_i}{s}\right)\left(\frac{1}{s^2+2.5s+1}\right) = \frac{K_p + \frac{K_i}{s}}{s^2+2.5s+1} = \frac{K_p s + K_i}{s^3 + 2.5s^2 + s}$$

$$T_{OL}(j\omega) = \frac{K_p j\omega + K_i}{-j\omega^3 - 2.5\omega^2 + j\omega}$$

Using these to find K_p and K_i: $|T_{OL}(j\omega_c)| = 1$ & $\measuredangle T_{OL}(j\omega_c) = -180° + \phi_m$

$$|T_{OL}(j\omega_c)| = \frac{\sqrt{K_i^2 + K_p^2\omega_c^2}}{\sqrt{(-2.5\omega_c^2)^2 + (\omega_c - \omega_c^3)^2}} = 1$$

$$\sqrt{K_i^2 + K_p^2} = \sqrt{(-2.5)^2}$$

$$K_i^2 + K_p^2 = 6.25$$

$$K_i^2 + (2.747K_i)^2 - 6.25 = 0$$

$$K_i = \pm 0.855$$

$$\measuredangle T_{OL}(j\omega_c) = \tan^{-1}\left(\frac{K_p\omega_c}{K_i}\right) - \tan^{-1}\left(\frac{\omega_c - \omega_c^3}{-2.5\omega_c^2}\right)$$

$$\tan^{-1}\left(\frac{K_p}{K_i}\right) - \tan^{-1}\left(\frac{1-1}{-2.5}\right) = -180° + 70°$$

$$\tan^{-1}\left(\frac{K_p}{K_i}\right) - 180° = -180° + 70°$$

$$\frac{K_p}{K_i} = 2.747$$

$$K_p = 2.747K_i$$

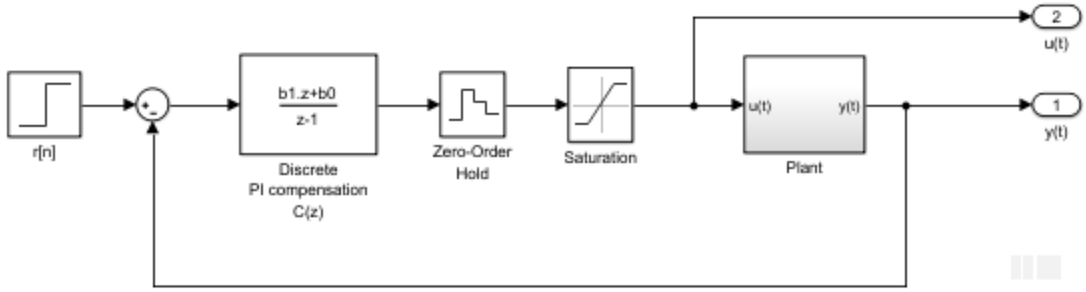
Choosing positive K_i to limit design space:

$K_i = 0.855$

$K_p = 2.747(0.855) = 2.349$

$$\boxed{\text{PI compensator: } C(s) = 2.349 + \frac{0.855}{s}}$$

4. Build a SIMULINK model for digital PI control of the circuit with a saturated control $0 \le u \le 5V$ and sample time $T = 0.1s$



Simulink =>Discrete =>Discrete Transfer Fcn. Sample time=> T
Simulink =>Discrete =>Zero-Order Hold. Sample time=> T

Emulate (discretize) the PI compensator from part 3 using the backward differencing method.

$$C(s) = \frac{U(s)}{E(s)} = 2.349 + \frac{0.855}{s}$$

$$sU(s) = 2.349\,s\,E(s) + 0.855\,E(s)$$

$$u'(t) = 2.349\,e'(t) + 0.855\,e(t)$$ inverse laplace

Backwards differencing method: using numerical approximation of the derivative:

$$\frac{u_n - u_{n-1}}{T} = 2.349\left(\frac{e_n - e_{n-1}}{T}\right) + 0.855\,e_n$$

$$u_n - u_{n-1} = 2.349 e_n - 2.349 e_{n-1} + 0.855 T e_n$$

$$u_n = u_{n-1} + (2.349 + 0.855T) e_n - 2.349 e_{n-1}$$

$$u_n = u_{n-1} + 2.435 e_n - 2.349 e_{n-1}$$ using T= 0.1s

$$U(z) = z^{-1} U(z) + 2.435 E(z) - 2.349 z^{-1} E(z)$$ taking z-transform

$$U(z)(1 - z^{-1}) = E(z)(2.435 - 2.349 z^{-1})$$

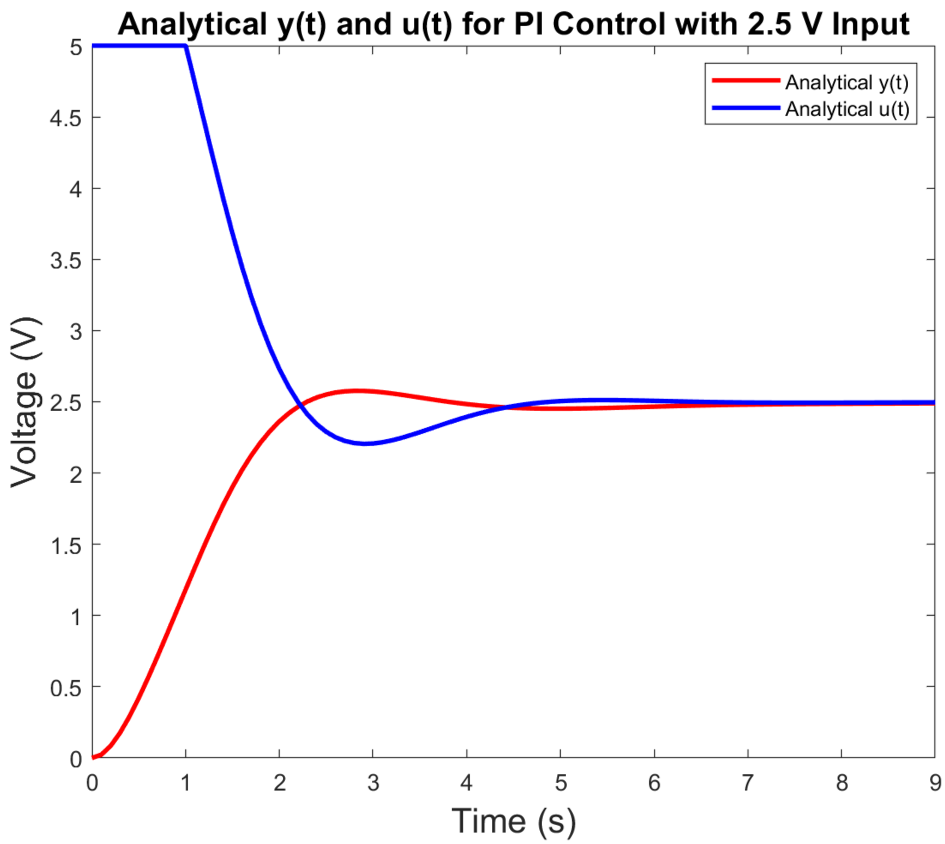
$$\frac{U(z)}{E(z)} = \frac{2.435 - 2.349 z^{-1}}{1 - z^{-1}}$$

$$\boxed{C(z) = \frac{2.435 - 2.349 z^{-1}}{1 - z^{-1}}}$$

$$C(z) = \frac{2.435 z - 2.349}{z - 1}$$ ⟶ put this into discrete PI compensation C(z)

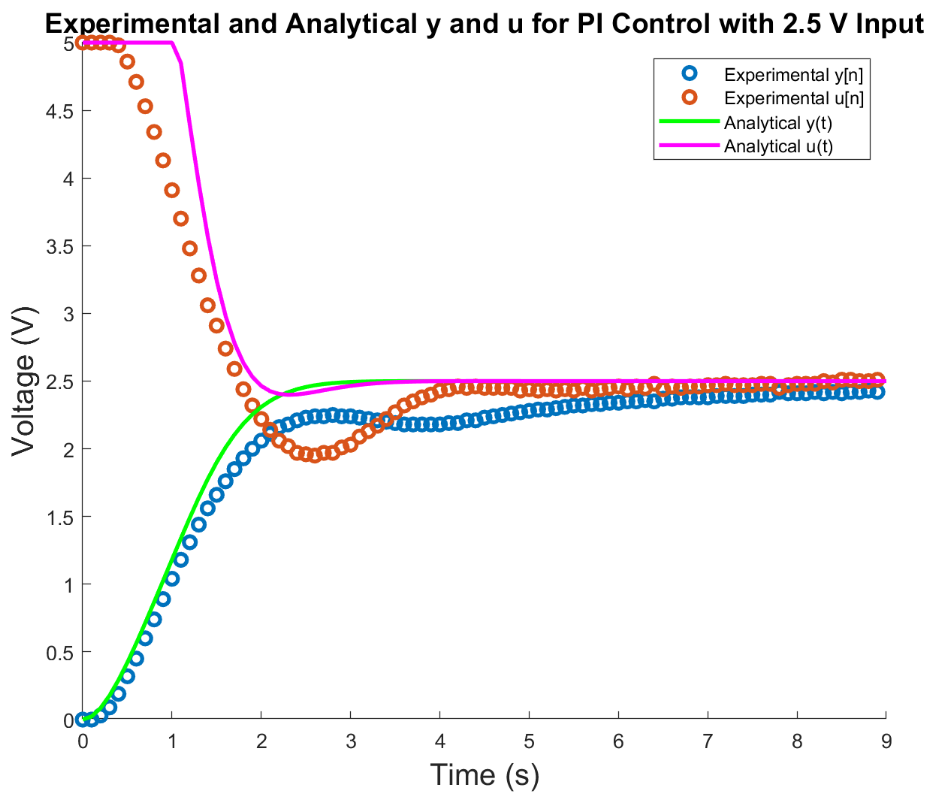
Insert a fully labeled simulated closed loop step response plot of the digital control system for a 2.5V step reference input $r[n]$. Design for a phase margin $\phi_m = 70^o$ at crossover frequency $\omega_{co} = 1\, rad/s$ Plot both the response $y(t)$ and the control input $u(t)$ on the same figure.

(insert figure here)



5. Program the discrete PI closed loop control for the LMC6484 op amp circuit Insert a fully labeled experimental closed loop step response plot for a 2.5V step reference input, with an overlay of the simulated step response from part 4. Plot both the response $y[n]$ and the control input $u[n]$ on the same figure.

(insert figure here)



Complete a table of trials of values of design parameters ϕ_m, ω_{co} with the resulting experimental values of 1% settling time $t_{settling}$; peak overshoot M_p; steady state error e_{ss} and the summation of the absolute deviations of the control input from the final steady state control input $U = \sum_{n=0}^{\infty} |u[n] - u_{ss}|$ where $u_{ss} = \lim_{n \to \infty} u[n]$.

ϕ_m	ω_{co}	$t_{settling}$ [sec]	M_p	e_{ss}	$U = \sum_{n=0}^{\infty} \lvert u[n] - u_{ss} \rvert$
70	1	10.8	0.256%	0	14.96
60	1	7.35	4.4%	0	29.2682
60	0.9	5.45	10.48%	0	34.3436
50	0.9	6.92	21.4%	0	41.78
70	0.8	5.93	5.28%	0	30.75
→ 65	0.9	5.85	4.00%	0	377.35

- **less than 1% peak overshoot**
- **zero steady state error**
- **minimize the control input keeping 1% settling time under 4 s**
- **sample at 10 Hz**
- **control input saturates at 5V**

Note: we tried all possible combinations and variations of ϕ_m and ω_{co} but were not able to get a trial that met the required specifications. We spent around 10 hours trying to do this, but were still unsuccessful. We noted the closest trial to meeting the specification above.

6. Design an observer-based discrete-time state feedback control of the circuit (plant)

- Determine the equivalent discrete-time state space description of the plant for a sample period $T = 0.1s$ Use MATLAB **c2d** command.

$$\mathbf{x}[n+1] = \mathbf{A_d}\mathbf{x}[n] + \mathbf{B_d}u[n]$$
$$y[n] = \mathbf{C_d}\mathbf{x}[n] + D_d u[n]$$

Record $\mathbf{A_d}, \mathbf{B_d}, \mathbf{C_d}, D_d$

Used continuous state space and c2d command on Matlab to get:

$$A_d = \begin{bmatrix} 0.9512 & 0.0442 \\ 0 & 0.8187 \end{bmatrix}$$

$$B_d = \begin{bmatrix} 0.0046 \\ 0.1813 \end{bmatrix}$$

$$C_d = [1 \quad 0]$$

$$D_d = [0]$$

- Design a full state feedback controller to minimize the objective function:

$$J = \frac{1}{2}\sum_{n=0}^{\infty}\left(Q_{11}x_1[n]^2 + Q_{22}x_2[n]^2 + u[n]^2\right) \qquad \text{with } Q_{11} = 100; Q_{22} = 1;$$

Use MATLAB **dlqr** command . Record the full state feedback gain vector ***K*** and the discrete-time closed loop poles

Using matlab's dlqr command: dlqr(Ad, Bd, Q, R)

$$A_d = \begin{bmatrix} 0.9512 & 0.0442 \\ 0 & 0.8187 \end{bmatrix} \quad B_d = \begin{bmatrix} 0.0046 \\ 0.1813 \end{bmatrix} \quad Q = \begin{bmatrix} 100 & 0 \\ 0 & 1 \end{bmatrix} \quad R = 1$$

We got the following:

$K = [6.5003 \quad 1.2703]$

discrete time
closed loop poles $P_{1,2} = 0.7548 \pm 0.1323j$

- Select the reference gain K_r for zero steady state error to a constant reference input. Record K_r

calculated Kr on matlab using the following equation:

$$K_r = \frac{1}{(C_d - D_d\bar{K})(I - A_d + B_d\bar{K})^{-1} B_d + D_d}$$

$\boxed{K_r = 8.7706}$

- Design an observer with pole locations at twice the speed of the closed loop poles Record **L** and the observer poles

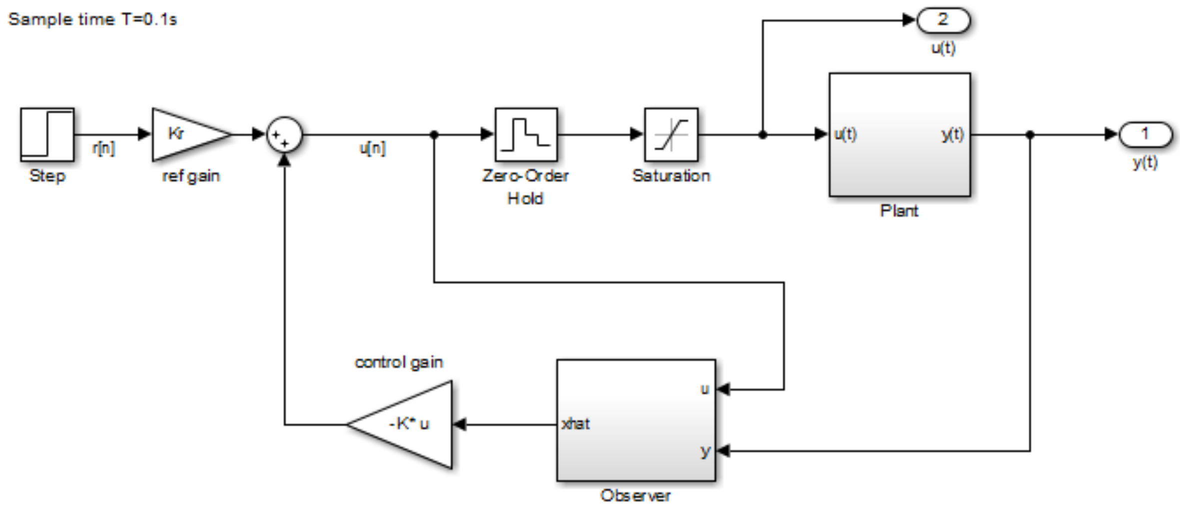
chose observer poles to be twice as fast as discrete time closed loop poles:

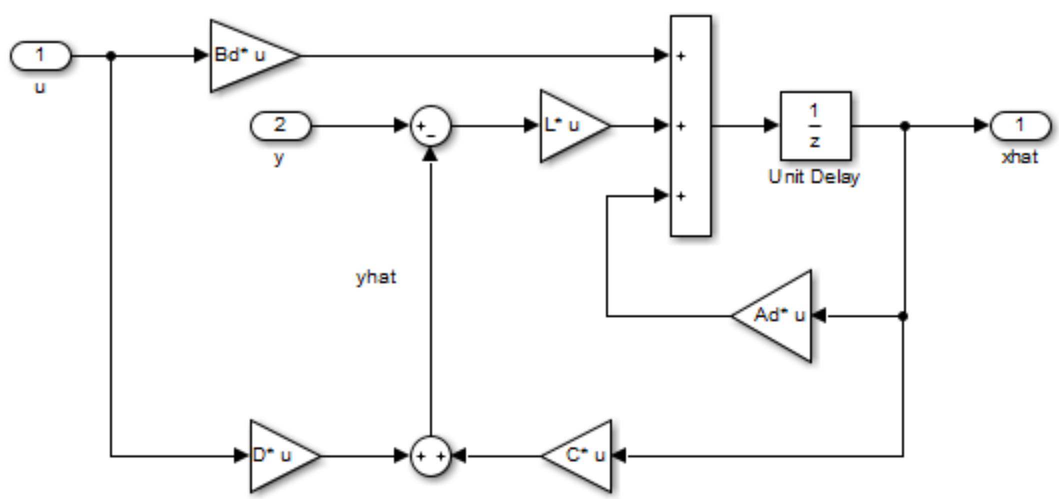
$\boxed{\text{observer poles: } 0.3774 \pm 0.0661j}$

using place command in Matlab to find L:

$$\boxed{L = \begin{bmatrix} 1.0151 \\ 4.5045 \end{bmatrix}}$$

7. Build a SIMULINK model for observer-based state feedback control of the circuit with a saturated control $0 \le u \le 5V$



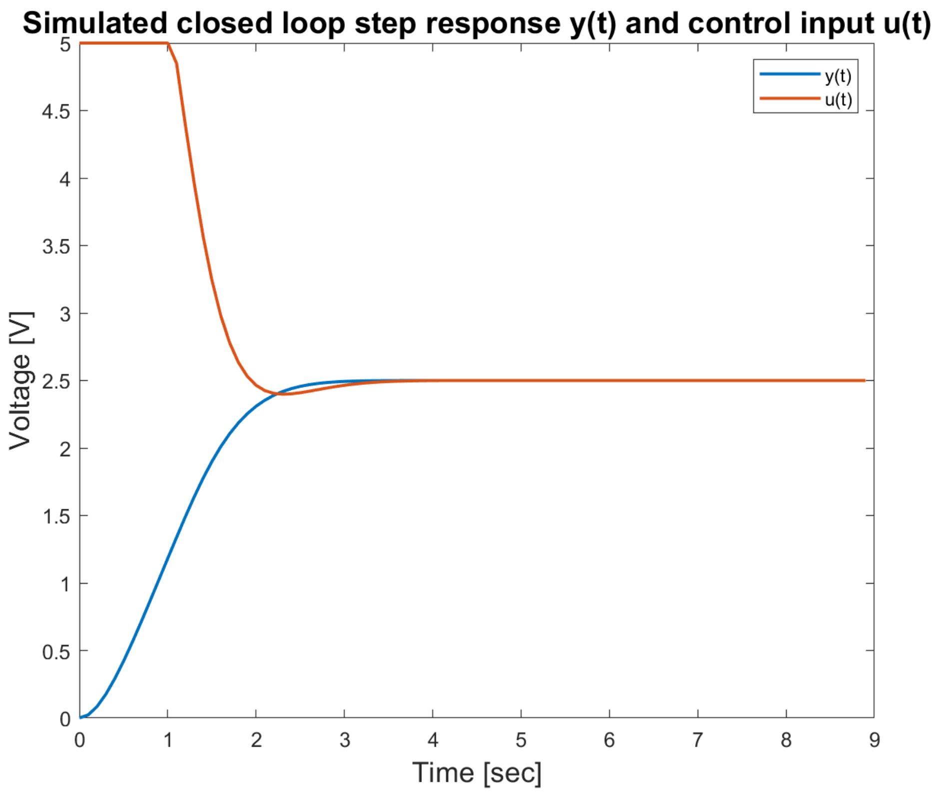


Discrete Observer Subsystem

*Simulink =>Gain =>*Multiplication=>Matrix(K*u)
Simulink =>Discrete =>Unit Delay.   Sample time=> T

Insert a fully labeled simulated closed loop step response plot of the digital control system for a 2.5V step reference input $r[n]$. Plot both the response $y(t)$ and the control input $u(t)$ on the same figure.

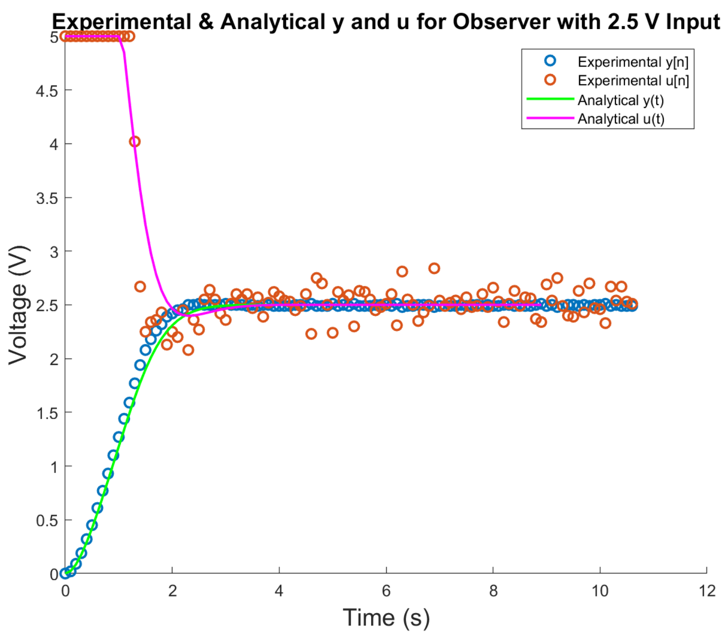
(insert figure here)

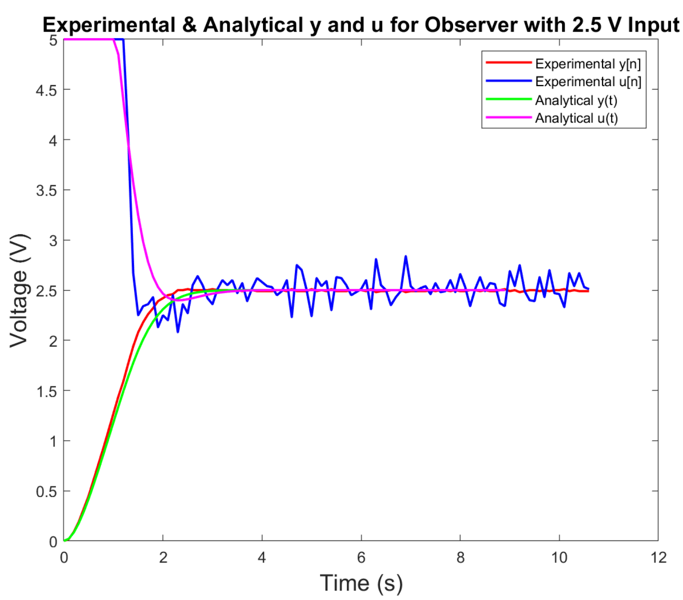


8. Program the observer-based state feedback control for the LMC6484 op amp circuit
Insert a fully labeled experimental closed loop step response plot for a 2.5V step reference input, with an overlay of the simulated step response from part 7
Plot both the response $y[n]$ and the control input $u[n]$ on the same figure.

(insert figure here)

Note: both are plots of experimental y[n] and u[n] overlayed with the simulated y(t) and u(t). We included a line plot to better visualize the oscillations in experimental u[n].





Complete a table of trials of values of design control parameters Q_{11}, Q_{22} with the resulting experimental values of 1% settling time $t_{settling}$; peak overshoot M_p; steady state error e_{ss} and the summation of the absolute deviations of the control input from the final steady state control input $U = \sum_{n=0}^{\infty} |u[n] - u_{ss}|$ where $u_{ss} = \lim_{n \to \infty} u[n]$.

Q_{11}	Q_{22}	$t_{settling}$	M_p	e_{ss}	$U = \sum_{n=0}^{\infty} \lvert u[n] - u_{ss} \rvert$
100	1	2.2833	1.0204%	0	32.8636
→ 200	1	2.2375	0.5462%	0	30.6973
→ 200	10	2.2725	0.6195%	0	31.5345
→ 100	5	2.75	0.6562%	0	29.4814
100	100	7.225	0.438%	0	8.2836
1	100	13.15	0.4383%	0.01	7.0455
50	50	7.025	0.4016%	0.01	16.5991
→ 300	5	2.35	0.9468%	0	31.23

→ These trials met the required specifications of < 1% peak overshoot, zero steady state error, and < 4 sec settling time.

9. Insert a short (1 page) discussion of your rationale for the choice of the control parameters and the effect of your choices on the experimental results

To begin, we tested "extreme" cases to figure out how the control parameters were related to each other and to output parameters ts (settling time), Mp (max percent overshoot), ess (steady state error), and U (summation of absolute deviations). In table 1, we tuned our PI controller by changing ϕm and ωco, which altered Kp and Ki. We first altered each parameter independently: from ϕm = 70 and ωco = 1, we decreased ϕm to 60 while keeping ωco constant, then kept ϕm constant at 60 while decreasing ωco to 0.9. We then tested increasing ϕm to 70 while decreasing ωco to 0.8, to explore the effect of larger separation between the parameters. These tests revealed that a relatively high ϕm, near 70, resulted in a relatively low Mp (anywhere between 0.256% and 10%), while a ωco around 0.8 to 0.9 resulted in a relatively low ts (anywhere between 5 and 6 seconds). We then attempted to find a middle ground between these tests that combined our desired low Mp and low ts with 0 steady state error. While we were unsuccessful in meeting the exact desired specifications after hundreds of rounds of trial and error, our best-performing system had a ϕm = 65 and ωco = 0.9, resulting in ts = 5.85 and Mp = 4% (with ess = 0). We believe that continuing to tune ϕm and ωco according to the findings in this table will eventually result in behavior within specifications.

In table 2, we tuned our observer system by changing Q11 and Q22, which changed Kr, K, and L. Beginning with Q11=100 and Q22=1, we first altered each parameter independently to gauge the effect parameter-specific changes had on our output ts, Mp, ess, and U. While holding Q22 constant first, we increased Q11 to 200; we then increased Q22 to 10 while keeping Q11 constant at 200. We then decreased both Q11 and Q22 but kept the ratio between Q11 and Q22 constant to see if this had any effect on output parameters (from Q11=200 and Q22=10 to Q11=100 and Q22=5). We tested setting the Q values to the same value 100, then swapped values of Q11 and Q22 such that Q22 was greater than Q11 to examine whether an opposite input had an opposite effect on output. From this series of tests, we determined that smaller changes to Q22 impacted output parameters more greatly than changes of similar magnitude to Q11. Additionally, we found that a relatively greater Q11 (around 200-300) resulted in a lower ts (anywhere between 2.2 and 2.4), while higher Q22 (around 50-100) resulted in a lower Mp (anywhere between 0.4% and 0.44%). Seeing as Q11 and Q22 weigh the importance of minimizing deviation in state variables x1 and x2, respectively, our observed experimental changes are as expected. Since x2 governs the internal dynamics that influence the convergence of the output, changing Q22 has a greater impact on the output parameters. By increasing Q11, we are penalizing variations in x1 (output voltage), which causes the output to reach steady state faster and thus lowers settling time. Since Q22 penalizes deviations in x2 (internal state), it constrains how much momentum can build up in those internal dynamics, which lowers the overshoot. Because the second entry in this table (Q11=100 and Q22=1) satisfied design specifications for our system, we did not need to further tune the Q values.

10. Insert a short (1 page) discussion of the sources of error between the simulated and experimental step response results

Hardware limitations are one of the primary sources of error that contributed to discrepancies between the simulated and experimental step response results. We used 100kΩ resistors in our circuit, placing them in parallel and series to reach the required 50kΩ and 200kΩ, respectively. Real resistors have manufacturing tolerances, causing deviations between their expected and true values. Since we were using two resistors to represent the R and 4R each, these deviations compound to increase the variability between expected and actual resistance. This component-level variability also exists for capacitors. These deviations affect the time constant of the system, which leads to system dynamics such as settling time and max percent overshoot to have deviations from the simulations. In simulating the circuit, we used ideal op-amp laws to derive the transfer function of the system. However, op amp non idealities such as temperature dependence, having a finite input and output impedance, and a finite open-loop gain could lead to the observed discrepancies.

Other hardware and environmental factors possibly causing deviation include breadboard contact resistance, capacitance from wires in the circuit, environmental changes, and error in the experimental zero-order hold. We worked on the circuit over a span of two weeks. While the system is assumed to be an LTI model, the behavior of the components drifting over this two week period, due to temperature changes from the circuit being in a backpack for example, could lead to the output behavior deviating from the simulation. Also, while the zero-order hold maintains a constant output between samples, delays in the Arduino and variation in the digital-to-analog converter's implementation of the zero-order hold could lead to inconsistencies.

The simulated and experimental step response results for y[n] for the PI controller had greater variation than y(t) for the open loop system. One source of error to explain this is the noise present in the Arduino's analog-to-digital converter (ADC). In comparison, there was less deviation between the simulated and experimental y[n] using the observer-based state feedback control. This is because the PI controller, unlike the observer, does not estimate internal states; it is based purely on the difference between the reference and the output. This makes it more sensitive to hardware discrepancies and Arduino ADC noise. In contrast, the observer uses internal state estimation to adjust the control, which can smooth out some of these disturbances and better compensate for deviations between model and reality.

As shown in our plots, while y[n] remained relatively similar between the simulations and experiment, significant deviations are visible between the simulated and experimental u[n] in the observer. These can be explained by the observer gain L being designed to place poles at twice the speed of the closed-loop poles, causing the error to converge quickly. While fast observer poles reduce lag in state estimation, they also make the system more sensitive to measurement noise and model inaccuracies. Therefore, small disturbances in y[n] are amplified in the estimation process and propagate directly into u[n], leading to larger deviations in experimental u[n] compared to simulation.