Control LAB

Control LAB - HW2

Mohammad Parsa Dini - Std ID: 400101204

May 2025

1 System Identification and Control

1. Linearize the system around the operating point $(u_0, y_0) \approx (0.2, 0.5)$.

$$\dot{y} = -\frac{y}{5} + \frac{4.5}{20}\sqrt{u} = -0.2y + 0.225\sqrt{u}$$

Using Taylor expansion, $\sqrt{u}|_{u_0=0.2} \approx \sqrt{u_0} + \frac{u-u_0}{2\sqrt{u_0}} \approx 0.447 + 1.118(u-0.2) = 1.118u + 0.224$. Thus, around $u = u_0 = 0.2$:

$$\dot{y} = -0.2y + 0.225(1.118u + 0.224) = -0.2y + 0.252u + 0.0504$$

which suggests around the operating point: $\dot{y} \approx 0.002$. Taking the Laplace transform of $\dot{y} \approx -0.2y + 0.25u$ results in:

$$sY(s) = -0.2Y(s) + 0.25U(s) \implies H(s) = \frac{Y(s)}{U(s)} = \frac{0.25}{s+0.2}$$

2. Attempt to directly derive a first-order linear model for the nonlinear system shown in the figure. Apply a small step change around the operating point $(u_0, y_0) = (0.2, 0.5)$ and determine the system's gain and time constant.

$$\dot{y} = -\frac{y}{5} + \frac{4.5}{20}\sqrt{u} \implies \dot{y} = -0.1 + \frac{4.5}{20}\sqrt{0.2} \approx 0.0006$$

Let $u_1 = u_0 + \Delta u = 0.2 + 0.02 = 0.22 \implies 0.0006 = -0.2y_1 + \frac{4.5}{20}\sqrt{0.22} \approx -0.2y_1 + 0.1056$, which gives $y_1 \approx 0.5253$. Thus, $\Delta y = 0.5253 - 0.5 = 0.0253 \implies \text{gain} = \frac{\Delta y}{\Delta u} = \frac{0.0253}{0.02} \approx 1.265$. The Simulink result is depicted below: From Simulink, $K = \frac{0.562-0.5}{0.05} \approx 1.24$. However, the $K = 1.08$ from the .mlx file (available in the GitHub repository) is more reliable. For the time constant, $y_2 = 0.5 + 0.63(0.562 - 0.5) \approx 0.539 \implies t_2 = 34.7 \implies \tau = t_2 - 30 = 4.7$ (from Simulink).

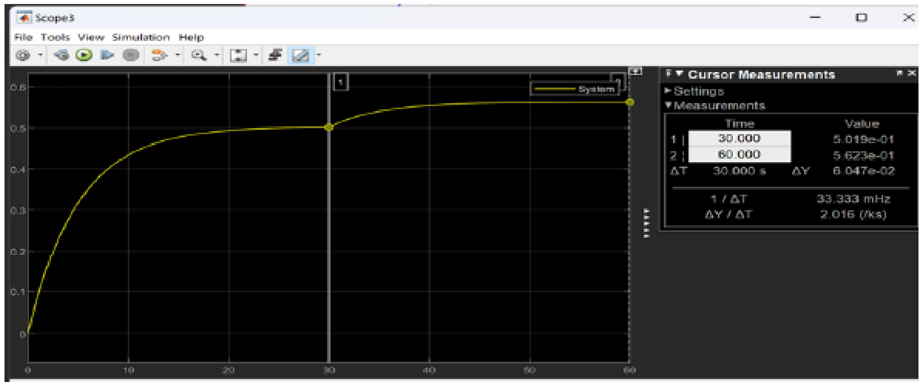


Figure 1: Simulink result for step response around operating point.

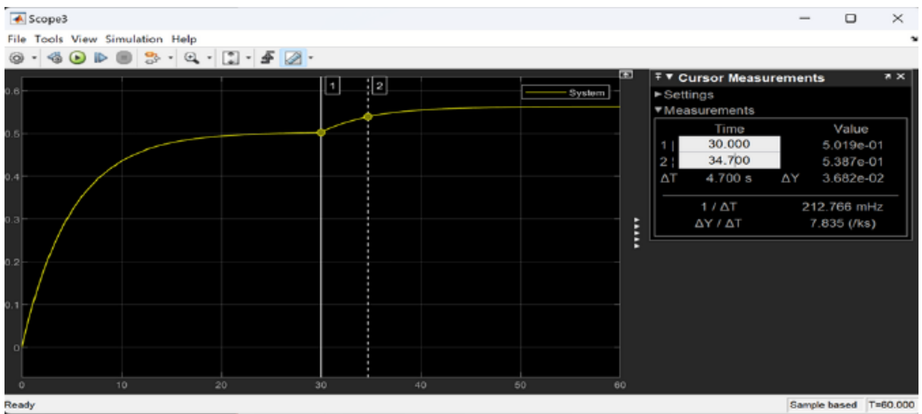


Figure 2: Simulink output for gain calculation.

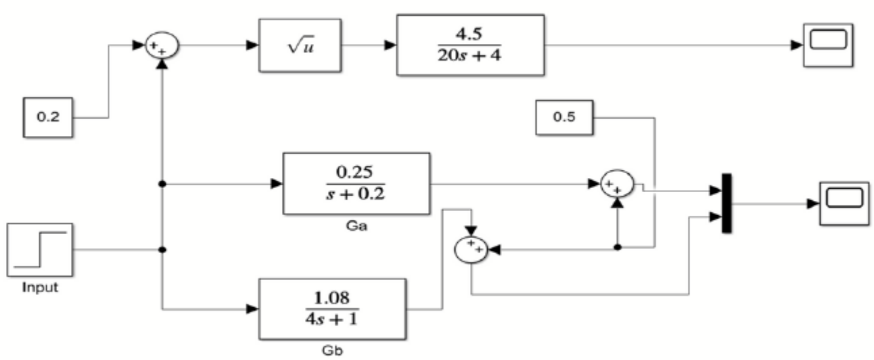


Figure 3: Simulink output for time constant calculation.

3. Compare the linear model derived in parts 1 and 2 with the actual system by applying a step change around the operating point $(u_0, y_0) = (0.2, 0.5)$. Plot the output of models and the real systems on the same axes. The results are shown in the .mlx file attached to the GitHub repository and this report. The Simulink result: As shown, $G_a(s)$ performed better

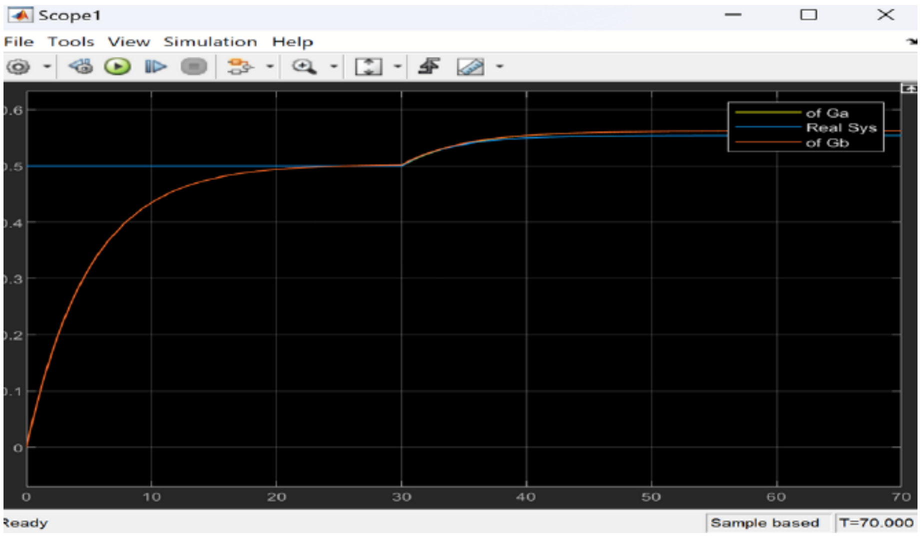


Figure 4: Comparison of linear model and actual system response.

in MATLAB and Simulink.

4. Design a PI controller to raise the output from 0.5 to 0.6 while satisfying:
   - Zero steady-state error for a unit step input.
   - Overshoot less than 10%.
   - Settling time less than 10 seconds.

   The PI controller is $u(t) = K_p(\beta r(t) - y(t)) + K_i \int_0^t (r(t) - y(t))\, dt$. With $\beta = 0.1$, we have $U(s) = -K_p Y(s) + K_i \frac{R(s)-Y(s)}{s}$. Using $Y(s) = G(s)U(s)$, the closed-loop transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{G(s)K_i/s}{1 + G(s)(K_p + K_i/s)} = \frac{K_i G(s)}{s + G(s)(K_p s + K_i)}$$

   Let $G(s) = \frac{K}{\tau s+1}$, so:

$$\frac{Y(s)}{R(s)} = \frac{\frac{KK_i}{\tau}}{s^2 + \frac{KK_p+1}{\tau}s + \frac{KK_i}{\tau}} = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

   Zero steady-state error is met since:

$$\lim_{s\to 0} \frac{Y(s)}{R(s)} = 1 \implies e_{ss} = \lim_{t\to\infty} e(t) = \lim_{t\to\infty} y(t) - r(t) = 0$$

For overshoot:

$$\text{Overshoot} = e^{-\frac{\pi\zeta}{\sqrt{1-\zeta^2}}} \le 0.1 \implies \zeta \ge \sqrt{\frac{\ln(0.1)^2}{\ln(0.1)^2+\pi^2}} \approx 0.591$$

Choosing $\zeta = 0.65$ satisfies $\zeta \ge 0.591$. For settling time:

$$T_s = \frac{4}{\zeta\omega_n} \le 10 \implies 2\zeta\omega_n \ge \frac{8}{10} \implies \frac{KK_p+1}{\tau} \ge 0.8$$

Using $G(s) = \frac{0.25}{s+0.2} = \frac{1.25}{5s+1}$, we have $K = 1.25$, $\tau = 5$. Thus:

$$K_p = 3.4 \ge \frac{\frac{8\tau}{\beta}-1}{K} = 2.4$$

Also, $\omega_n^2 \ge \frac{16}{\beta^2\zeta^2} \implies \frac{KK_i}{\tau} \ge \frac{16}{\beta^2\zeta^2}$, so:

$$K_i = 1.6 \ge \frac{16}{K\zeta^2\beta^2} \approx 1.51$$

In MATLAB and Simulink, all criteria are met.

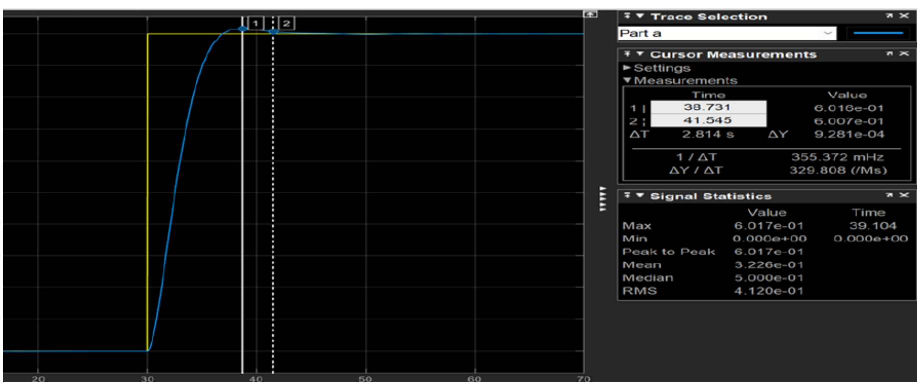


Figure 5: Simulink output for PI controller performance.

2 Controller Design & Actuator Dynamics

Consider the system with transfer function $G(s) = \frac{1.5}{0.5s+1}$. Design a controller to satisfy:

- Zero steady-state error.
- 0% overshoot.
- Settling time less than 0.15 seconds.

1. What type of controller would you recommend to best satisfy the design criteria?
   Since zero steady-state error requires an integrator, a PI controller is appropriate, as P, PD, lag, or lead controllers lack the $\frac{1}{s}$ term. We proceed with a PI controller.

2. Design an appropriate controller and assess its performance for an input step change from 50 to 60. Provide a detailed explanation of your design process.

$$G(s) = \frac{1.5}{0.5s+1} \implies K = 1.5, \tau = 0.5$$

   The PI controller is $u(t) = K_p(\beta r(t) - y(t)) + K_i \int_0^t (r(t) - y(t))\, dt$. With $\beta = 0.15$, the Laplace transform gives:

$$U(s) = -K_p Y(s) + \frac{K_i(R(s) - Y(s))}{s}$$

   Using $Y(s) = G(s)U(s)$, the closed-loop transfer function is:

$$\frac{Y(s)}{R(s)} = \frac{G(s)K_i/s}{1 + G(s)(K_p + K_i/s)} = \frac{\frac{KK_i}{\tau}}{s^2 + \frac{KK_p+1}{\tau}s + \frac{KK_i}{\tau}}$$

   Zero steady-state error is satisfied since:

$$\lim_{s\to 0} \frac{Y(s)}{R(s)} = 1 \implies e_{ss} = 0$$

   Zero overshoot requires $\zeta \geq 1$. For settling time, $T_s = \frac{4}{\zeta\omega_n} \leq 0.15 \implies 2\zeta\omega_n \geq \frac{8}{0.15} \approx 53.33$, so:

$$\frac{KK_p + 1}{\tau} \geq 53.33 \implies K_p \geq \frac{\frac{8\tau}{0.15} - 1}{K} \approx 17.1$$

   Choosing $K_p = 100$ satisfies this. For $\zeta \geq 1$, we need:

$$\left(\frac{KK_p + 1}{\tau}\right)^2 \geq 4\frac{KK_i}{\tau} \implies K_i \leq \frac{(KK_p+1)^2}{4K\tau} \approx 13467$$

   Choosing $K_i = 8000$ satisfies this. The Simulink result: The .mlx file confirms all criteria are met.

3. Now consider the block diagram where the actuator dynamics are given by $A(s) = \frac{0.99}{0.1s+a}$. Implement your designed controller in this new open-loop structure and evaluate its performance for $a \in \{1, 2, 5, 10\}$.

   (a) Case $a = 1$: Unstable, as shown in Figure 7a.
   (b) Case $a = 2$: Unstable, as shown in Figure 7b.
   (c) Case $a = 5$: Underdamped, as shown in Figure 7c.
   (d) Case $a = 10$: Underdamped, as shown in Figure 7d.

   For $a \in \{1, 2\}$, the time constant $\tau = 0.1/a$ is large, making the actuator slow, causing closed-loop poles to lie in the right half-plane, leading to instability. For $a \in \{5, 10\}$, τ is smaller, making the actuator faster, with poles in the left half-plane, resulting in stability. As a increases, ζ increases, approaching critical or overdamped behavior.

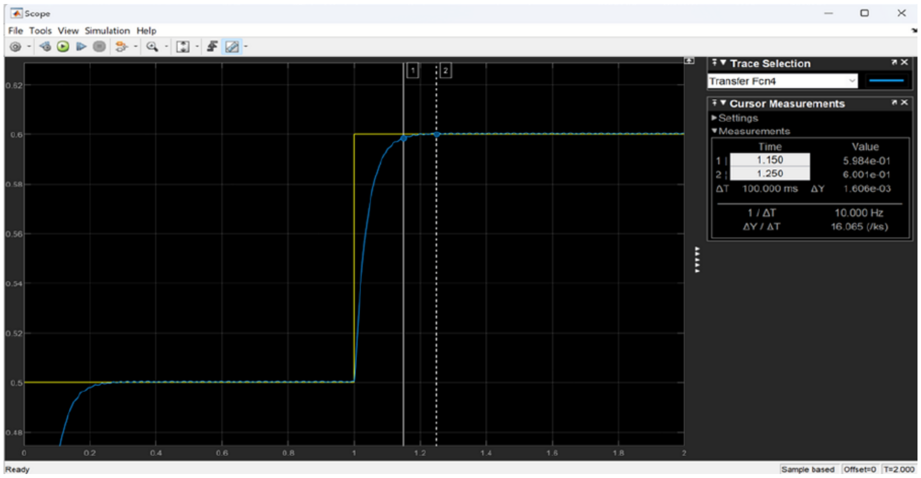
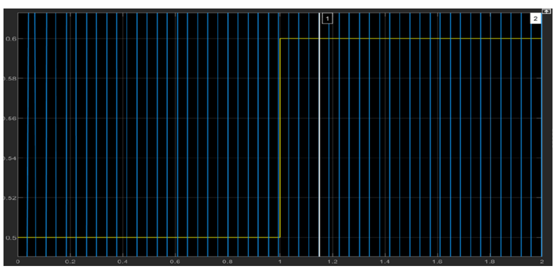
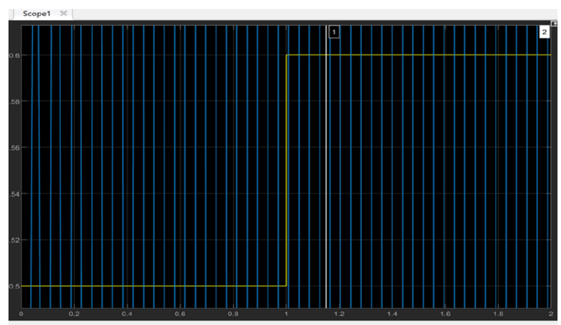


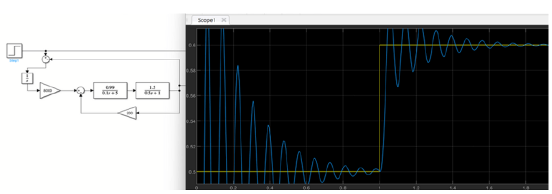
Figure 6: Simulink output for PI controller performance (step change from 50 to 60).



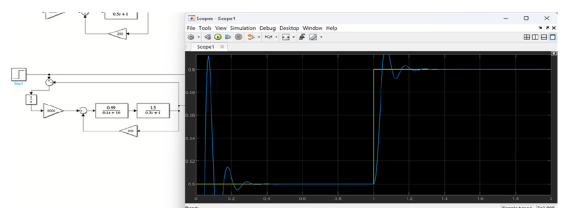
(a) Response for $a = 1$ (unstable).



(b) Response for $a = 2$ (unstable).



(c) Response for $a = 5$ (underdamped).



(d) Response for $a = 10$ (underdamped).

Figure 7: System responses for different values of a in actuator dynamics $A(s) = \frac{0.99}{0.1s+a}$.

Conclusion

This lab demonstrated system identification and control design for a nonlinear system. By linearizing the system around an operating point, deriving a first-order model, and comparing it with the actual system, we validated the linear approximation. The PI controller for the first system met

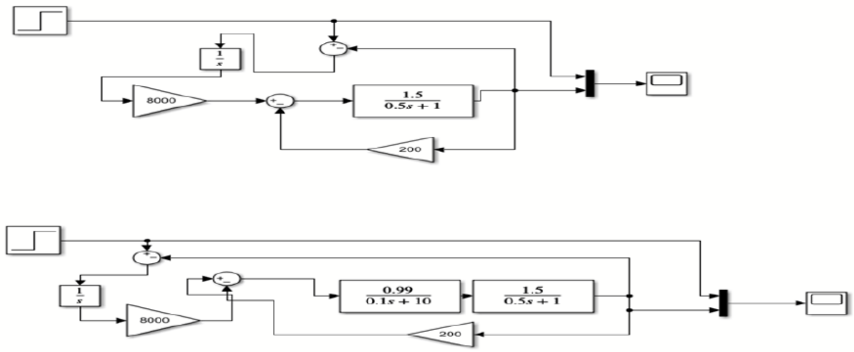


Figure 8: System response with and without actuator dynamics.

the requirements of zero steady-state error, overshoot less than 10%, and settling time less than 10 seconds. For the second system, the PI controller achieved zero steady-state error, zero overshoot, and settling time less than 0.15 seconds, with performance varying based on actuator dynamics, as verified through Simulink simulations.