HW5

Andy Tsai

April 30, 2025

a) Transfer Function for the plant

$$G(s) = \left(\frac{3}{s} + \frac{1}{2s^2+3}\right) \cdot \frac{1}{4s+1}$$

$$\Longrightarrow G(s) = \frac{6s^2+s+9}{s(4s+1)(2s^2+3)}$$

b) Plant's response to the input signal

As shown in Figure 1, the output $y(t)$ lags behind the input and has a larger amplitude. It doesn't track the input well, which shows that the open-loop system struggles with time-varying signals and needs feedback control.

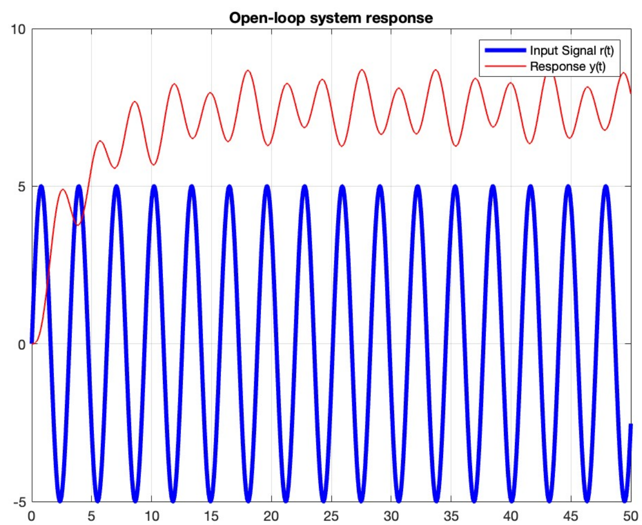
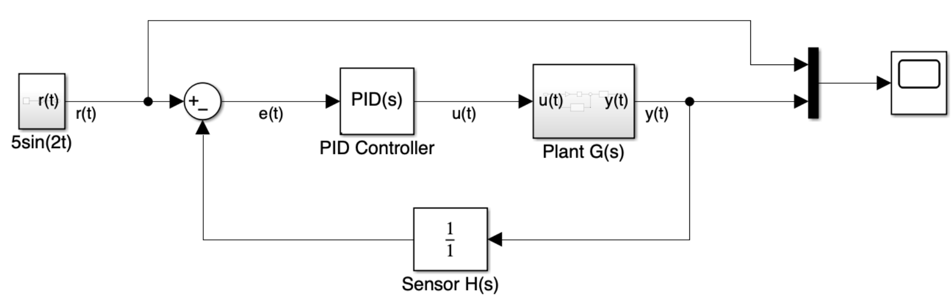


Figure 1: Open-loop response to input $r(t) = 5\sin(2t)$.

c) Block diagram of the close-loop system



d) PID Parameter Selection and Justification

To improve the system's performance, I tested three sets of PID parameters and observed their effects on the closed-loop response.

First, I used a simple proportional controller with $K_p = 1$, $K_i = 0$, $K_d = 0$. As shown in Figure 2, the system responded quickly but with large steady-state error and persistent oscillations, showing that K_p alone wasn't enough for good tracking.

Next, I tried a PI controller with $K_p = 0.5$, $K_i = 3$, $K_d = 0$ (Figure 3). Adding the integral term helped reduce steady-state error, but the system became unstable due to integrator windup, especially near the end of the simulation.

Finally, I used a full PID controller with $K_p = 0.5$, $K_i = 5$, $K_d = 15$ (Figure 4). This setup gave the best result: the output closely tracked the input with low error and minimal delay. The derivative term helped dampen oscillations and reduce overshoot.

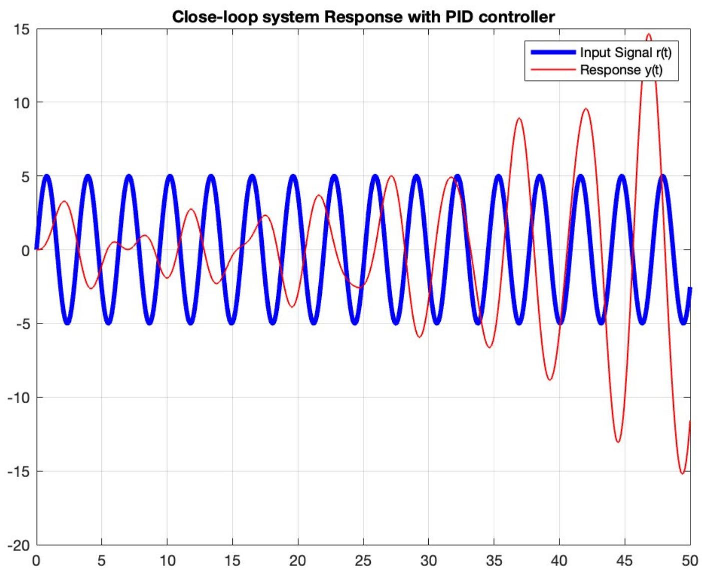


Figure 2: Closed-loop response with $K_p = 1$, $K_i = 0$, $K_d = 0$ (Proportional control).

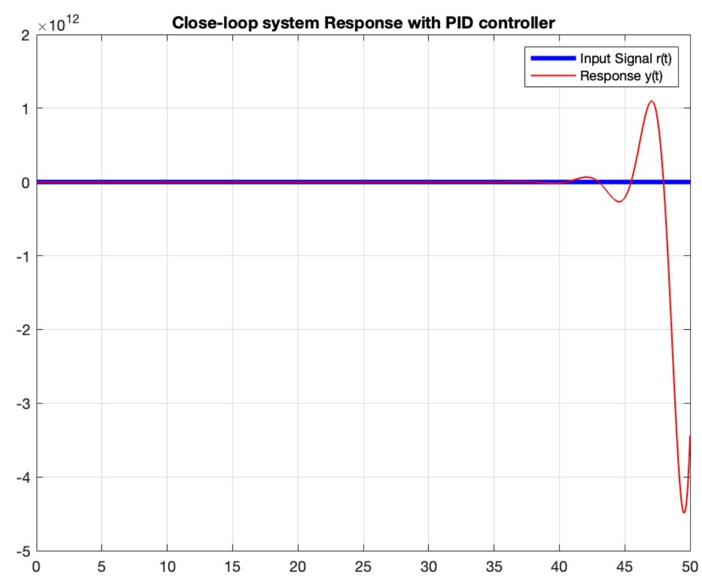


Figure 3: Closed-loop response with $K_p = 0.5$, $K_i = 3$, $K_d = 0$ (PI control).

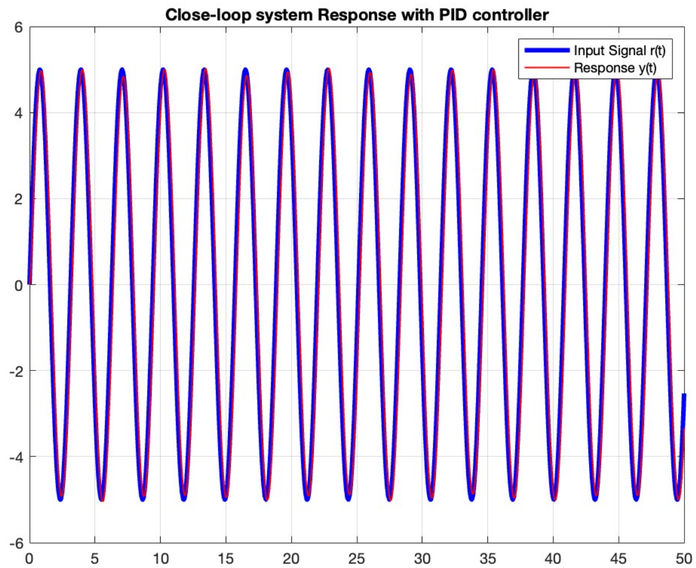
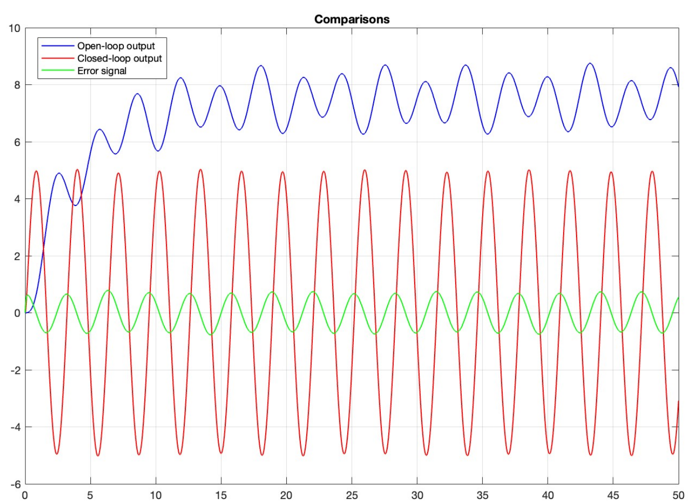


Figure 4: Closed-loop response with $K_p = 0.5$, $K_i = 5$, $K_d = 15$ (PID control).

e) System Response Comparison



f) Discussion of PID Controller Effects

The final PID controller significantly improved tracking performance. Compared to the open-loop system, the closed-loop response showed much better alignment with the input, with reduced phase lag and steady-state error. Increasing K_p helped speed up the response but introduced some overshoot. The integral term (K_i) eliminated steady-state error, but too much of it caused instability, as seen in the PI-only case. Adding derivative control (K_d) helped stabilize the system by damping oscillations and reducing overshoot. Overall, the PID controller provided a good balance between responsiveness and stability.