# Fall 2024 – 16-642 Manipulation, Estimation, and Control
# Problem Set 2

Due: Monday 7 October 2024

GUIDELINES:

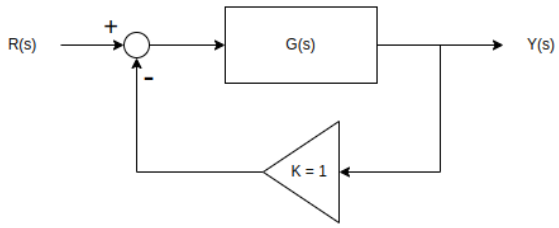
- You must *neatly* write up your solutions and submit all required material electronically via Canvas by **11:59 pm of the posted due date.**
- Include any MATLAB scripts that were used in finding the solutions. Also include any plots generated with MATLAB as a part of the question.
- You are encouraged to work with other students in the class, however you must turn in your own *unique* solution.
- Late Policy: You are given **in total 72 hours** of grace period for **ALL** homework across the semester, which you can use to give yourself extra time without penalty. Late work handed in when you have run out of grace will not be accepted.

1. (40 points) Picking up from Problem 2(h) in Problem Set 1, build an observer for the cart-pendulum system and repeat your most successful tracking controller simulation replacing the state feedback you used in Problem Set 1 with the state estimate from your observer. Simulate both the full nonlinear dynamics and the dynamics with an observer using a time step of $T = 0.01$ seconds, a final time of $t_f = 20$ seconds, with the actual starting state $x_0 = [0,\ 0,\ 0,\ 0]^T$ and the estimated starting state $\hat{x}_0 = [0.01,\ 0.01,\ -0.03,\ 0.01]^T$ (This is a randomly guessed estimated starting state to observe the difference between the estimated and actual states. You may experiment with different initial values, but in your submitted answer, please use the provided values). Plot the state vs. time (four plots in total). On each plot, overlay the estimated and actual states for comparison. If your implementation is correct, you will notice a slight difference at the beginning, but the estimated states will quickly converge to the actual states.

2. (30 points) Given the following plant model

   $G : u(t) \rightarrow y(t)$

   $$\dddot{y}(t) + 13\ddot{y}(t) + 78\dot{y}(t) = \dddot{u}(t) + 4\ddot{u}(t) + 80\dot{u}(t)$$




   Answer the following questions (5 points each):

   (a) Write down the model equation after applying the Laplace (L) transformation and determine the open-loop transfer function $G(s)$ for the system.

(b) Determine the closed-loop transfer function $T(\mathrm{s}) = \frac{Y(\mathrm{s})}{R(\mathrm{s})}$ with unity negative feedback.

(c) Determine poles and zeros of T (s) using MATLAB's $zero$ and $pole$ functions.

(d) Qualitatively describe the closed-loop system response to a unit step input using the poles and zeros of the closed-loop transfer function obtained in (c) - will it oscillate and why? will it be stable and why? How do the zeros affect the system performance?

(e) Plot the well-labelled (title, axes labels) step-response of the system using MATLAB's $step$ function.

(f) Find the steady state value using Final Value Theorem.

3. (30 points) Implement a PID controller in MATLAB and use it to control the plant with transfer function

$$G(s) = \frac{20s + 17}{s^4 + 9s^3 + 231s^2 + 400s + 60}.$$

Your design objectives are to have a rise time of 1 seconds, a maximum percent overshoot of less than 5%, and a steady state error of zero. Tune the gains manually to achieve these objectives. List the final gains you choose and provide a plot of the resulting closed loop step response.

Here are the step-by-step instructions for tuning PID parameters. We recommend following these steps to earn partial credit for solving parts (a) and (b):

(a) Increase the proportional gain ($K_p$) until you get a fast response.

(b) If there is steady-state error, increase the integral gain ($K_i$) to eliminate it, but be careful not to make the system oscillate too much.

(c) If the system is oscillating too much that exceeds the maximum percent overshoot, increase the derivative gain ($K_d$) to add damping.

If the current PID parameters fail to meet the design objectives, you may need to restart the tuning process by selecting a new $K_p$. Additionally, it is not always necessary to use all three parameters ($K_p$, $K_i$, and $K_d$). In some cases, the system can achieve the desired performance with just two parameters.