Maglve Project Report
2023 FALL EECS 460

Qinwen QIAN
Tenghao JI
Qunzhuo Feng

I. Controller Design

The plant transfer function is given by $P(s) = -\frac{3734}{s^2-2180}$. For the closed loop system to be asymptotically stable, a PD controller is required for the missing s-term and negative constant free term in the denominator of the function G(s). To satisfy the zero steady state error with respect to a step input, we need an additional I controller. Thus, a PID controller is designed with general equation of $C(s) = \frac{Ki}{s} + Kd * s + Kp$. Then the forward transfer function is

$$G(s) = C(s) * P(s) = -\frac{3734\left(Ki+Kp*s+Kd*s^2\right)}{[s(s^2-2180)]}.$$

By taking Kd out of the parenthesis in the numerator, the forward transfer function becomes

$$G(s) = -3734Kd * \frac{\left(s^2+Kp*\frac{s}{Kd}+Ki*\frac{s}{Kd}\right)}{s(s^2-2180)}.$$

Here we define a new gain K such that

$$G(s) = K * \frac{\left(s^2+Kp*\frac{s}{Kd}+Ki*\frac{s}{Kd}\right)}{s*(s^2-2180)}, \text{ where } K = -3734Kd.$$

As maximum overshoot is 10% and settling time is smaller or equal to 3 seconds, we chose damping, ζ to be 0.6. For 2% tolerance rate, settling time = $4/\zeta\omega_n$. Since theta is chosen to be 0.6 and $\zeta\omega_n$would be greater than 4/3, natural frequency, ω_n would be greater than 2.22. The area of poles we can choose is shown in purple in Figure 1.1. We chose the pair of complex roots at -10±j, then the function

$$G(s) = K * \frac{s^2+20s+101}{[s(s^2-2180)]}, \text{ where } \frac{Kp}{Kd} = 20 \text{ and } \frac{Ki}{Kd} = 101.$$

By plotting the root lotus diagram shown in Figure 1.2, we found out that when the damping ζ = 0.6, the gain K = 278. Recall equation $K = -3734Kd$, Kd = 278/(-3734)=-0.074. Then Kp = -1.489 and Ki = -7.520. The PID controller we designed is $C(s) = -\frac{7.520}{s} - 1.489 - 0.074s$.

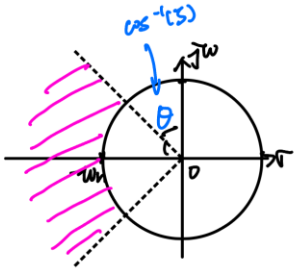

Figure 1.1: Area of complex poles in the s-plane to satisfy the requirements.

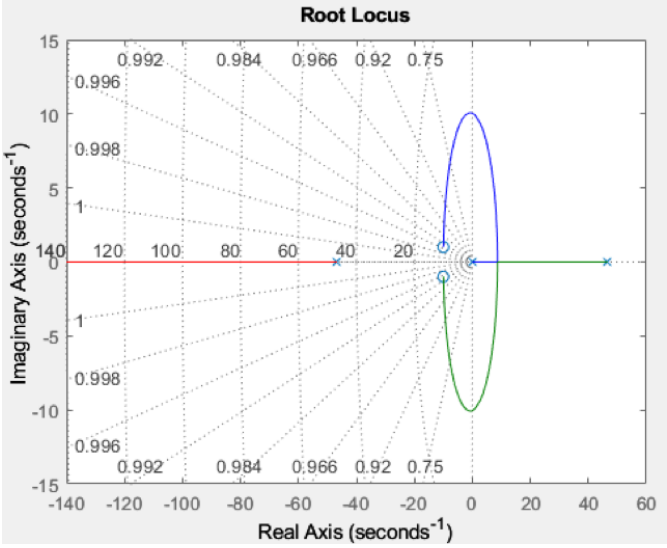

Figure 1.2: Root locus of chosen poles at -10±j.

Because of new poles introduced by the PD controller, a pre-compensator is required in the system to relocate those new poles. We chose our precompensator's transfer function to be $G_{pre}(s) = \frac{101}{s^2+20s+101}$.

II. Closed-loop System Performance Analysis

A. Maglve's linearized system simulation

The linearized plant model is shown in Figure 2.1. The plant transfer function is $P(s) = -\frac{3734}{s^2-2180}$. The PID controller is placed before the plant and a precompensator is used after the input. We defined the input to be a step input with initial value of -0.5 and final value of +0.5 to mimic the reference. From the scope shown in Figure 2.2, the overshoot is 9.770%, which is smaller than the required 10%. The settling time is 0 second.

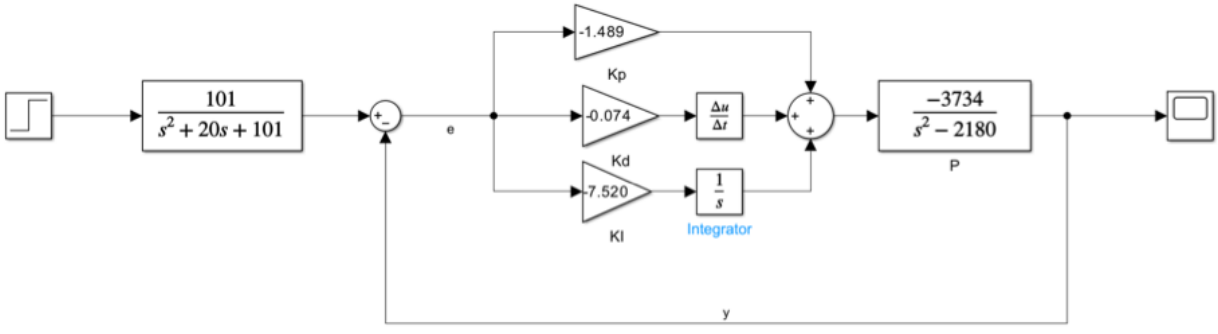

Figure 2.1: SIMULINK diagram of linearized system.

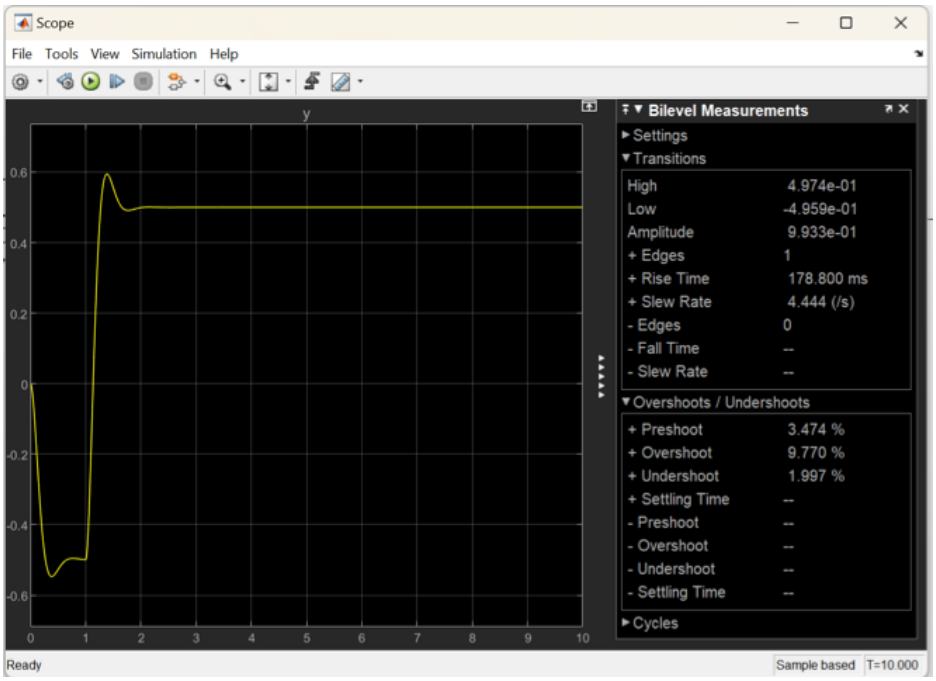

Figure 2.2: Scope of output response with step input from -0.5cm to +0.5cm.

B. Maglve's nonlinear system simulation

When we do the controller design using the Maglev's nonlinear model in SIMULINK, the settling time is 3.5803 seconds and maximum overshoot is 9.4594% by using MATLAB command 'stepinfo'. The maximum overshoot is lower than 10%, satisfying the requirement. Although the settling time is 3.5803 seconds, the system step input jumps at the time of 3 seconds. So, the actual settling time for the system is 0.5803 seconds, satisfying the requirement as well. The output of the Maglev's nonlinear system is shown in Figure 2.3.

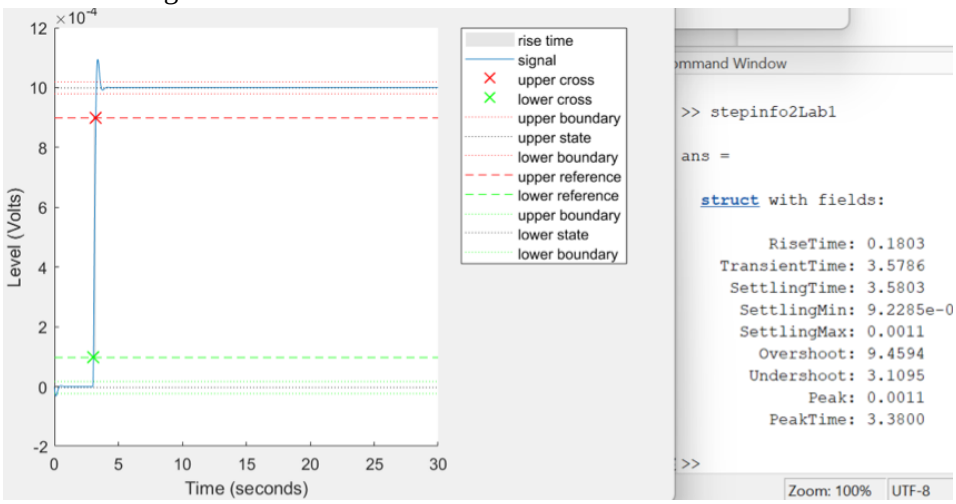

Figure 2.3: Output response plot using Maglev's nonlinear system.

C. Maglve's system simulation

When we implemented the controller on the Maglev computer, the controller did not stabilize the system at first. But after tuning the three gain parameters, it can stabilize the system as Figure 2.4 shows below. The possible reason for the original design not working is that the required input to the plant is relatively large for the initial designed Kd value.

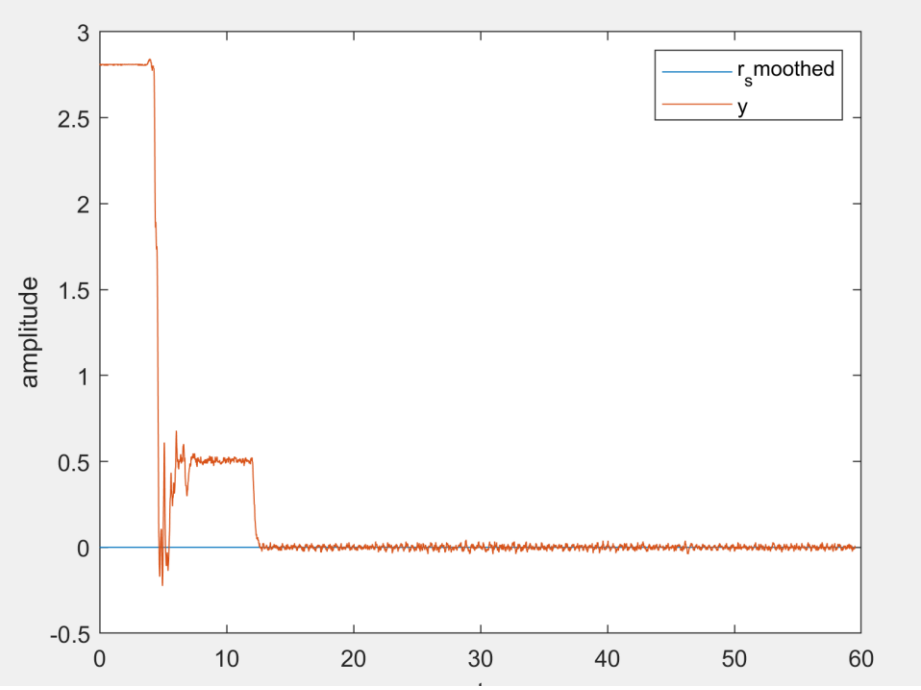

Figure 2.4: Output response with zero input.

By using command 'stepinfo' in MATLAB, we define the range of y and t after 10 seconds, and set the final y number to get maximum overshoot and settling time. The response of step reference with amplitude of 0.4cm, the settling time is 2.3444 seconds, steady state error is 0 and the overshoot is 3.7598%. Therefore, all the specifications were met.

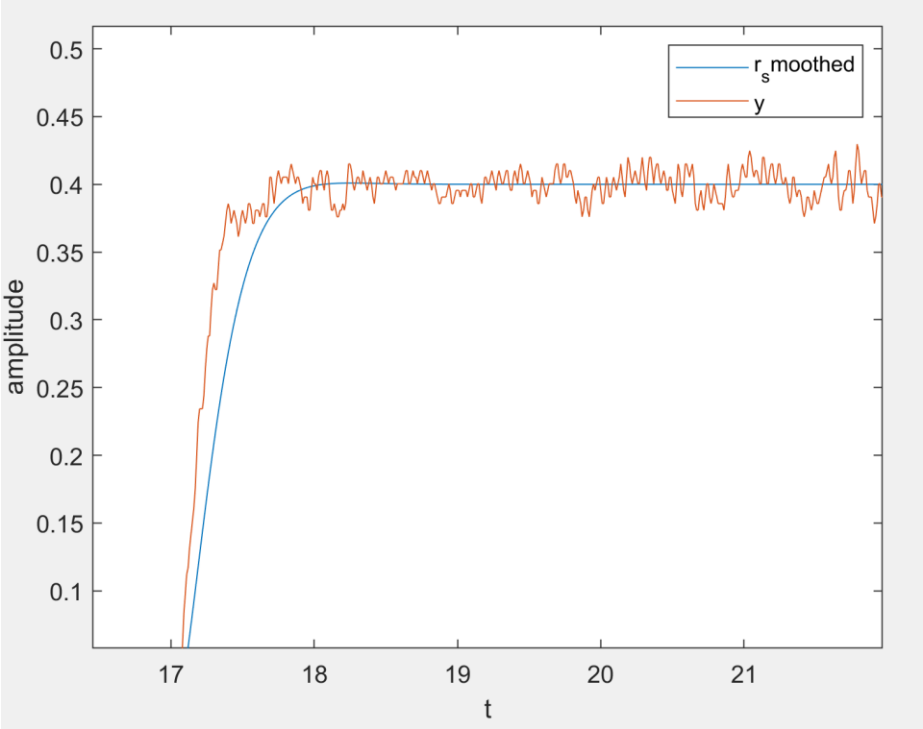

Figure 2.5: Output response with a step input of 0.4cm.

To get the overshoot for square wavewith amplitude of 0.4, we still use MATLAB command ‘stepinfo’, with -0.4 set as y_initial value and +0.4 set as y_final. We neglected the first square wave cycle since the y value did not start from -0.4. The overshoot for square waves is 17.1387%, 11.0352%, 9.8145%, 9.8145%, and 8.5938% respectively corresponding to the square waves starting from the second to the sixth wave. The settling times are 3.6164 seconds, 2.8932 seconds, 2.9732 seconds, 2.2764 seconds and 2.9639 seconds. The specification for overshoot was not fully met.

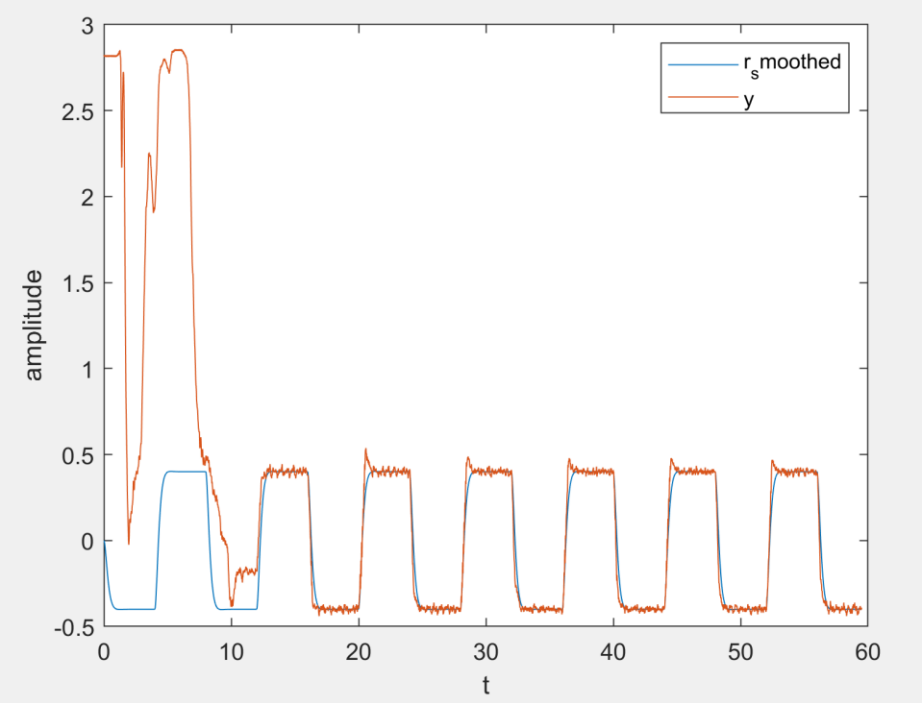

Figure 2.6: Output response to square wave input with amplitude of 0.4cm.

Again we do not take the first square wave into consideration when calculating the overshoot. For the response to square wave with amplitude of 0.45cm, the overshoot was 23.7847%, 16.7318%, 14.5616%, 17.8168% and 9.1363%, respectively corresponding to the square waves starting from the second to the sixth wave. The settling times are 3.37 seconds, 4.12 seconds, 2.8284 seconds, 2.5212 seconds, 2.6316 seconds, and 2.0115 seconds respectively. The time taken for overshoot to satisfy the specification is longer than square wave with amplitude of 0.4. Reason that overshoot decreases as time passed is that larger input may introduce a larger error and the I controller will work to eliminate this error over time.

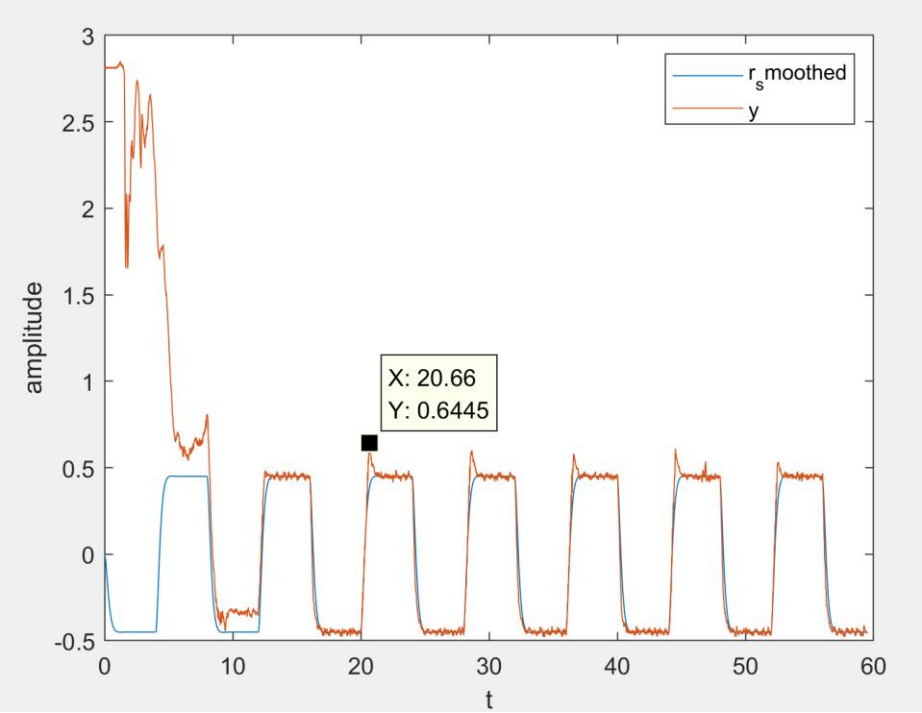

Figure 2.7: Output response to square wave input with amplitude of 0.45.

For the response to square wave reference with amplitude of 0.5cm, the overshoot was 20.8008%, 15.4297%, 18.8477%, 12.0117%, and 13.4766% respectively corresponding to the square waves starting from the second to the sixth wave. The settling times are 2.693 seconds, 2.535 seconds, 2.2141 seconds, 2.8017 seconds, and 2.1308 seconds respectively.

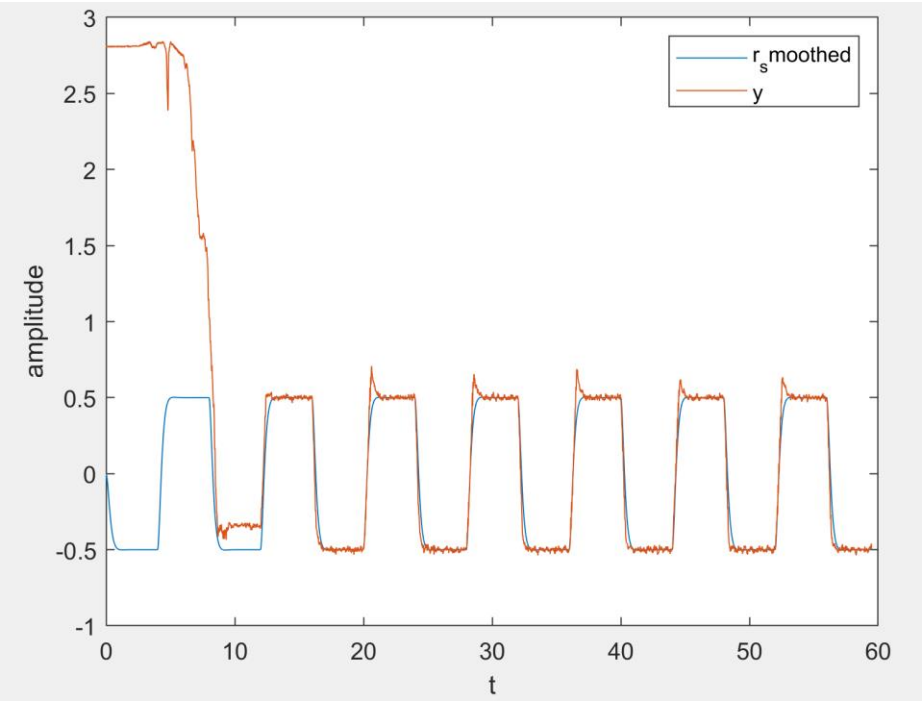

Figure 2.8: Output response to square wave input with amplitude of 0.5.

For the response to square wave reference with amplitude of 0.6cm, the overshoot was 8.1868%, 8.1868%, 6.1523%, 4.1178%, and 2.8971% respectively corresponding to the square waves starting from the second to the sixth wave. The settling times are 1.9407 seconds, 2.041 second, 2.854 second, 2.2928 second, and 1.7532 second respectively. Previously assume $\pm 0.5cm$ range, but as it gets larger it exceeds this range. Surprisingly the overshoot and settling time still satisfy the specifications. But it might be we only consider the rising edge of the output to compute overshoot. If we include the part where the falling edge, number of overshoot increases shown in Figure 2.9.

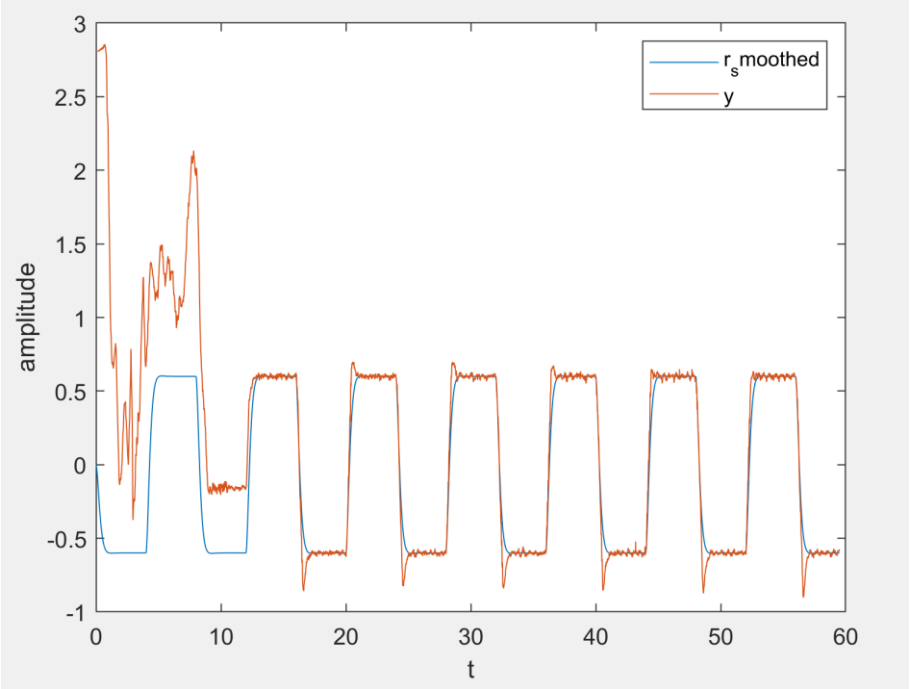

Figure 2.9: Output response of square wave with amplitude of 0.6.

Our controller design did not function well with a square wave. As the square wave's amplitude gets larger, the performance gets better. We did need to tune the parameters to meet the design specifications for smaller amplitude square wave. We decreased the value of Kp for smaller overshoot and solved the problem of the overshoot. With the new designed controller before tuning, we can stable the system with 0.6 square wave. We made the assumption that the new poles introduced by the PID controller will be a pair of complex poles and another real pole which is insignificant. We did not check whether the third closed loop pole is insignificant when we did the design. The maglev can be operated on a 'shaky' base, but the output response will receive more disturbance than 'non-shaky' bases. The result also depends on how large the 'shaky' is. The maglev will break down if it is too 'shaky'.

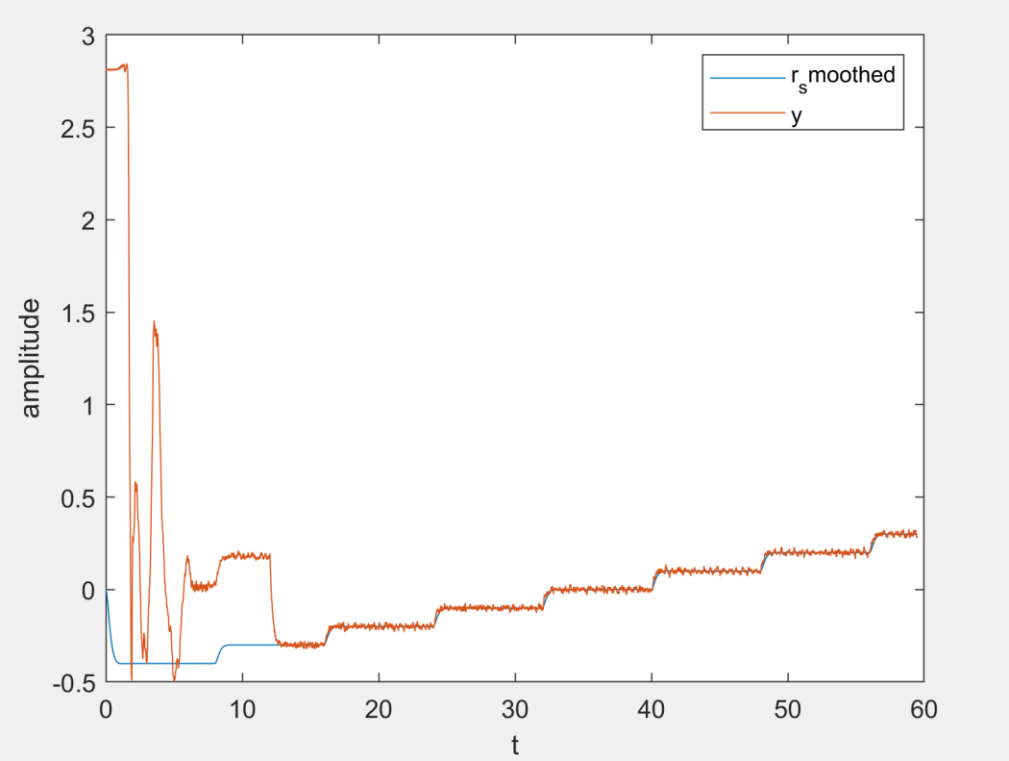

Figure 2.10: Output response with a stair wave.

We simulated the disturbance by manually pushing the ball down or up a bit. The disturbance is shown in Figure 2.11 as the two impulses at the time of 18 seconds and 28 seconds. The system became stable as it returned to the steady state error after each impulse was applied.

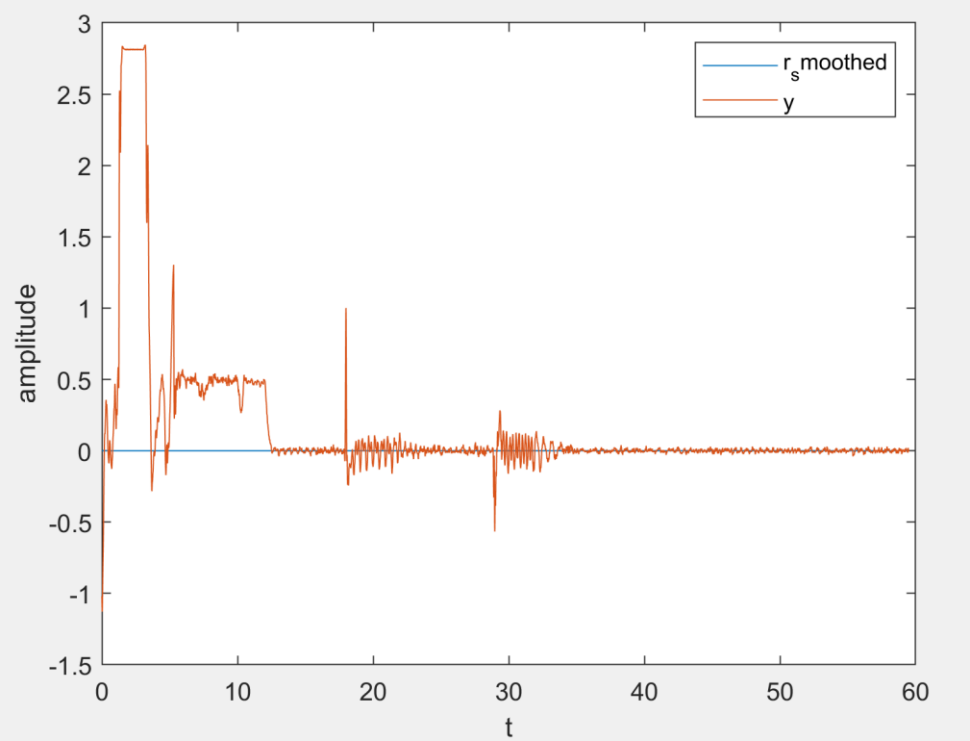

Figure 2.11: Output response with simulated disturbance

In the Maglev project, we designed a controller for the metal ball to float in the air despite the disturbance. In practice, magnetic levitation can be used in those popular levitated plant holders. The technique we have used in the Maglev project can be applied to keep the plant levitated. Our design is a straightforward extension to what we have performed, only for levitated plant holders, the magnet is at the bottom. There might be challenges when the levitated plant holder is plugged in. When we tested our design, we needed to balance the ball after turning on the system. Lifting the plant holder, especially with dirt and plant inside, long enough to find the equilibrium position might be hard for customers. If the device is plugged off, the plant holder would directly drop off and break itself. One of the advantages of our design is that customers may directly add water to the plant without taking the plant holder off the device, as weight of the water acts as disturbance.

III. Conclusions

With the gain parameters designed in Lab Assignment part-A, the simulation in SIMULINK using both linearized and nonlinear models works fine. But when we set the gain parameters in the lab, we were unable to find the equilibrium point for the ball to float for the test to begin with. Since we set the value of Kd so small, we need the input to the plant relatively large to achieve the required system output. However, the input to the plant cannot be as large as we want, we need a larger Kd value. We chose another pair of complex poles at $-5 \pm j\sqrt{5}$ and recalculated the gains. Our new controller's equation is $2.6 - 0.267s - \frac{8}{s}$. Not only can it satisfy the request about the stability, steady state error, setting time and overshoot, it can also stabilize the system with different inputs and even with existence of disturbance.

The performance of the controller is good in general. First of all, the controller can stabilize the system well as the ball can stay in a specific position and would not fall or rise dramatically when we put the ball in the system. The controller can also stabilize the system well with the different inputs, such as step input and square wave input, and the ball can maintain in the specific position despite the amplitude of the input. Moreover, the performance of the controller with respect to impulsive disturbance is also good. The ball would return to the steady state position after we apply an upward or downward force to mimic disturbance. However, our design does not satisfy the specifications with a square wave input. We did need to tune the gain parameters to achieve the overshoot requirement.

We spent most of the time trying to make our initial design work, but the system just did not work even with a zero input. GSI said the lab station layout was different from what he set up last time. Someone might move something during the Thanksgiving break (we had our lab right after the Thanksgiving break). He could not figure out the reason causing the trouble. Since we were running out of time, we chose another pair of complex poles to re-design the new controller.

The most challenging part is that the ball is supposed to be placed at the equilibrium position first even before the integrator finishes its initialization. If we put the ball too low or too low, it would fall into the ground because of gravity or be held to the magnet because of the high magnetic force respectively. We should put the ball in a suitable position and we also spent a lot of time adjusting the position so that the system can stabilize.