LAB 4 Report - ECSE 403

Mohamed Reda EL KHILI - 260678513

François-Eliott Rousseau - 260670000

Ismail Faruk - 260663521

1. The open loop transfer function used was [derived from Lab 3 Q 4.2.4]. I used this transfer function by integrating the open loop transfer function of the experimental open loop transfer function for velocity/Voltage.

We get the closed loop transfer function .

For Kp = 5, 𝜁 = 0.055, ⍵n = 77.46 Hz

For Kp = 10, 𝜁 = 0.039, ⍵n = 109.54 Hz

For Kp = 20, 𝜁 = 0.0277, ⍵n = 154.92 Hz

For Kp = 50, 𝜁 = 0.0175, ⍵n = 244.95 Hz

2)

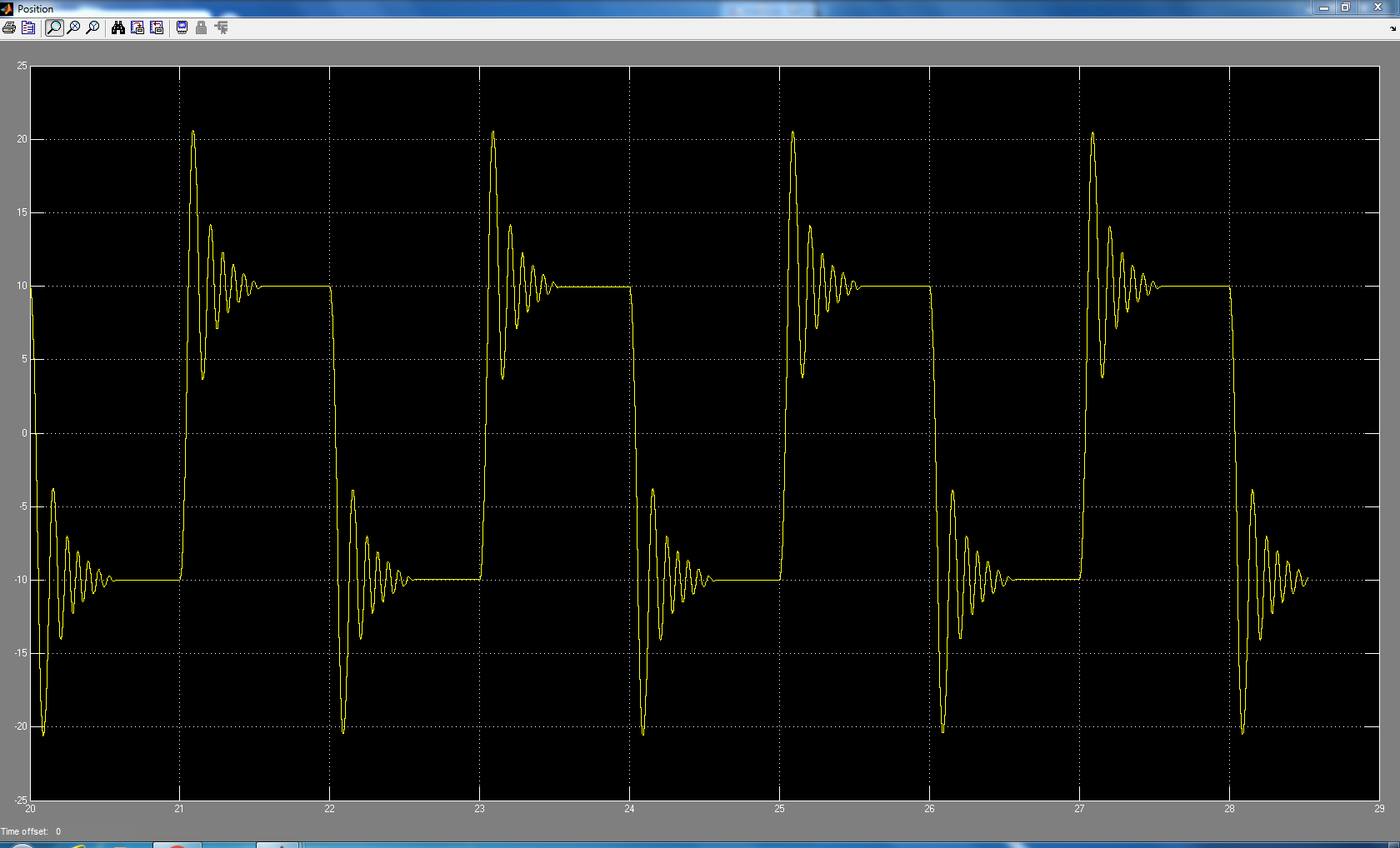


Figure 1: Step response of Proportional Position Controller with KP = 5

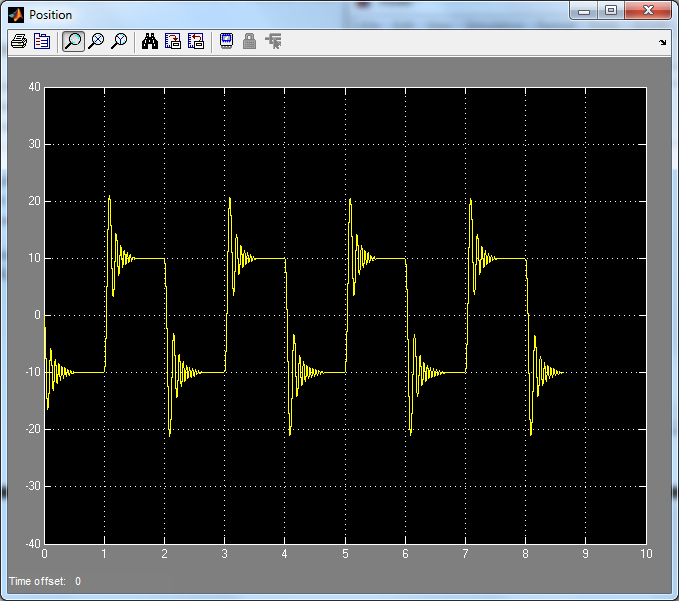


Figure 2: Step response of Proportional Position Controller with KP = 10

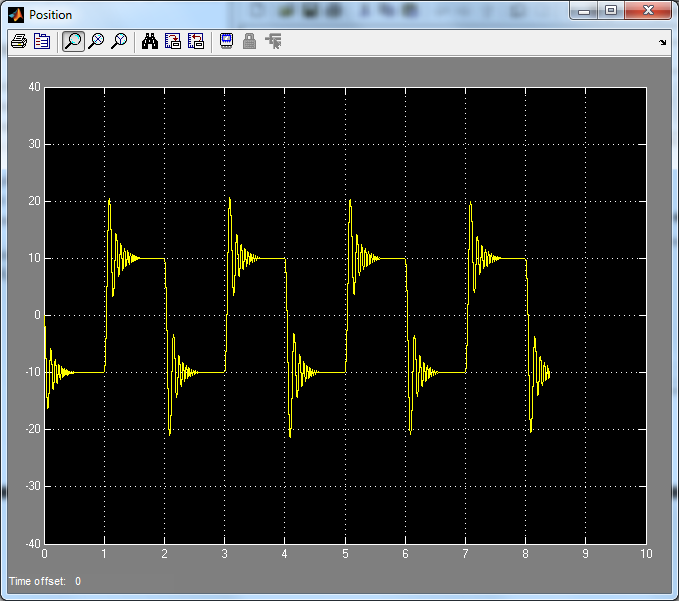


Figure 3: Step response of Proportional Position Controller with KP = 20

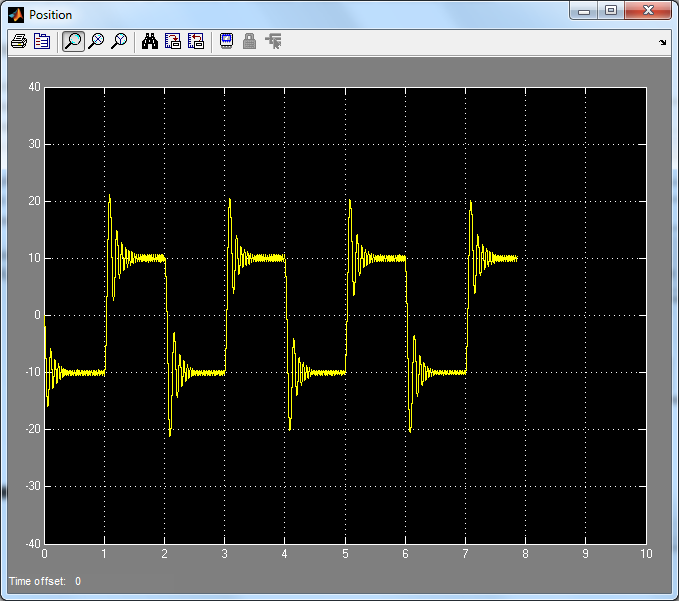


Figure 4: Step response of Proportional Position Controller with KP = 50

As we increase KP from 5 to 50, we first see that the rise time and the overshoot do not change. However, we can see that increasing KP leads to much more oscillations before the response reaches its steady state. We can thus see that the damping ratio decreases as KP increases. Moreover, we can see that, since the step response takes much longer to reach its steady state as KP increases, the steady state error increases.

3)

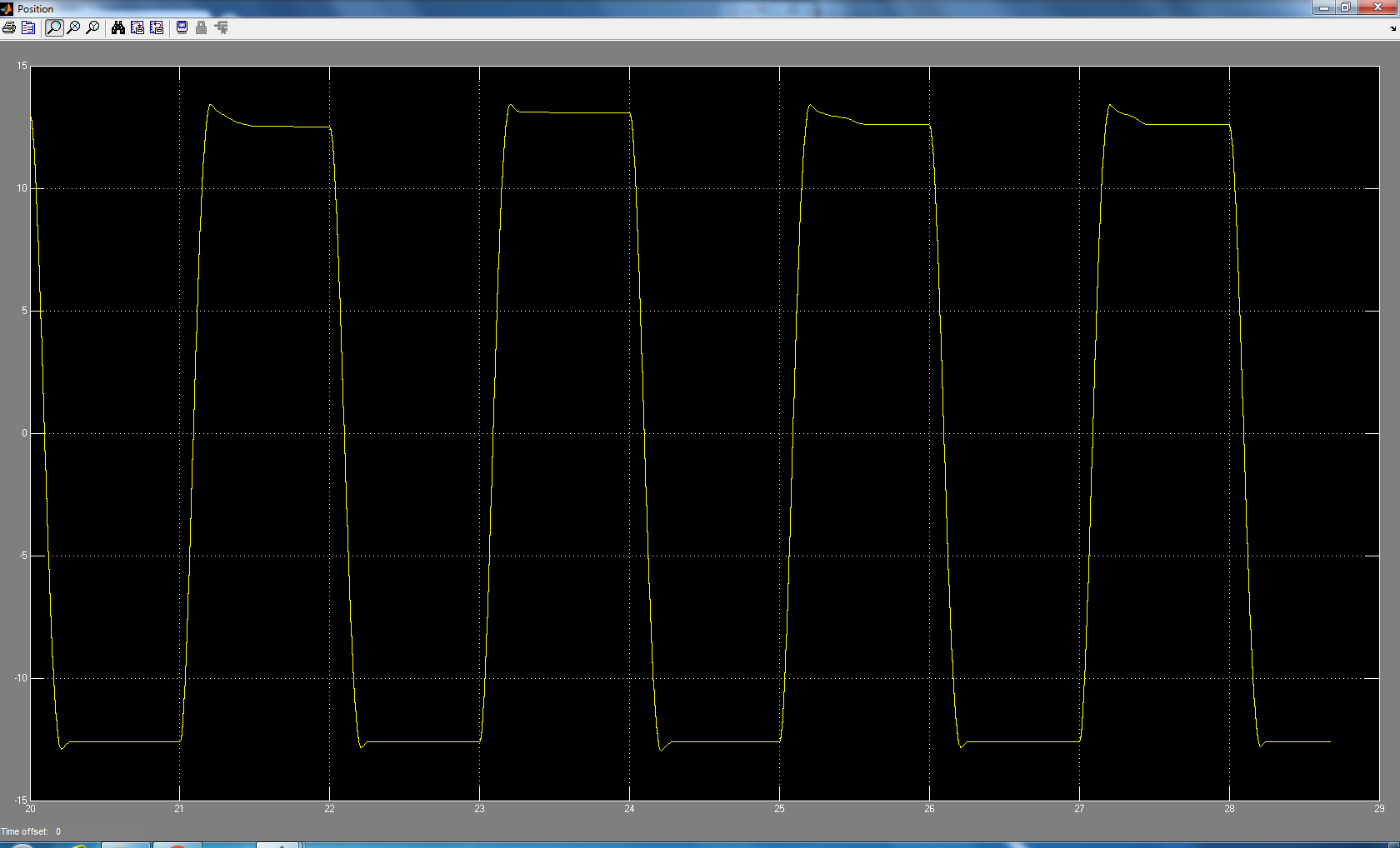


Figure 5: Step response with gain KP = 0.14 (we used a square wave)

With an input square wave of amplitude 10 and frequency 0.5 Hz, we obtain the appropriate gain of KP =0.14. We have a maximum overshoot of 13.432, a steady state of 12.6, a steady state error of 26%, a rise time of 0.116s and a settling time of 0.2s.

4) With Kp = 0.14, the theoretical step plot had a similar rise time. Its overshoot was significantly higher and settling time was relatively high. But there was no steady state error. The difference could be due to experimental error, caused by the apparatus being used.

5)

For KP = 10, we have:

|  |  |
| --- | --- |
| Frequency | Amplitude |
| 1 | 9.93 |
| 2 | 9.975 |
| 5 | 10.068 |
| 10 | 10.341 |
| 20 | 10.864 |
| 50 | 8.682 |

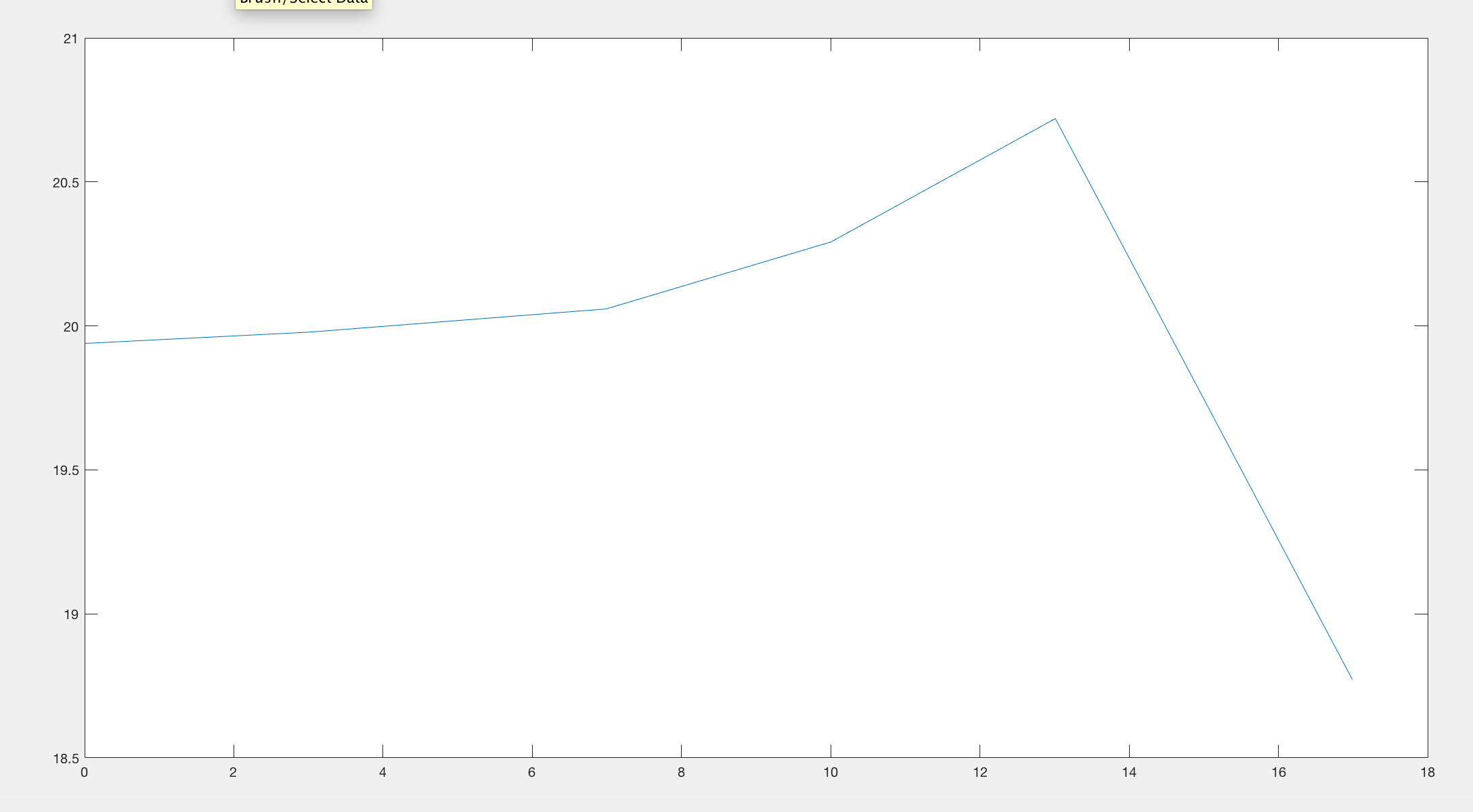


Figure 6: Bode Plot for KP = 10

For KP = 50, we have:

|  |  |
| --- | --- |
| Frequency | Amplitude |
| 1 | 19.5455 |
| 2 | 10.522 |
| 5 | 10.614 |
| 10 | 10.4772 |
| 20 | 10.6364 |
| 50 | 8.318 |

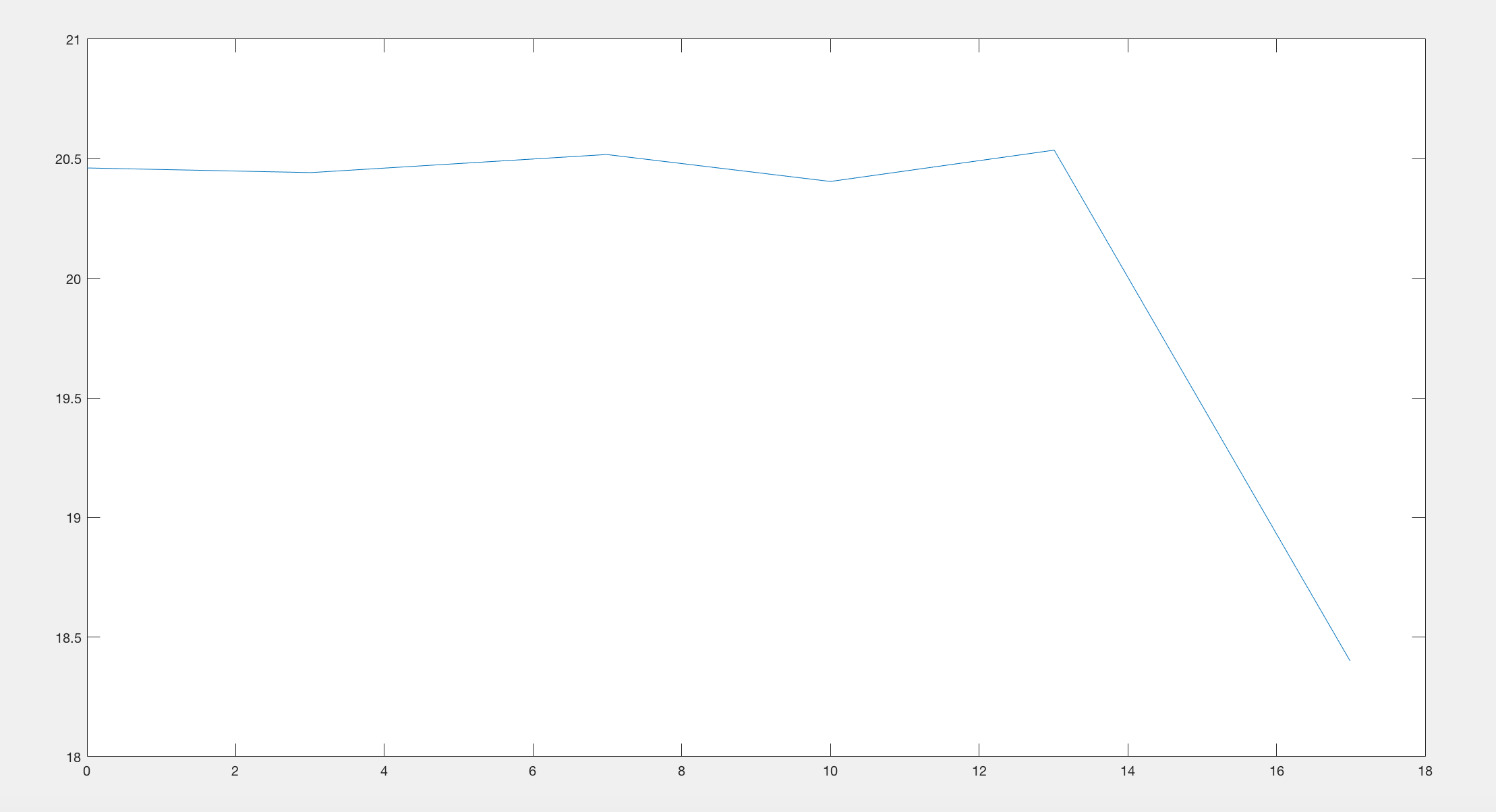


Figure 7: Bode plot for KP = 50

6)

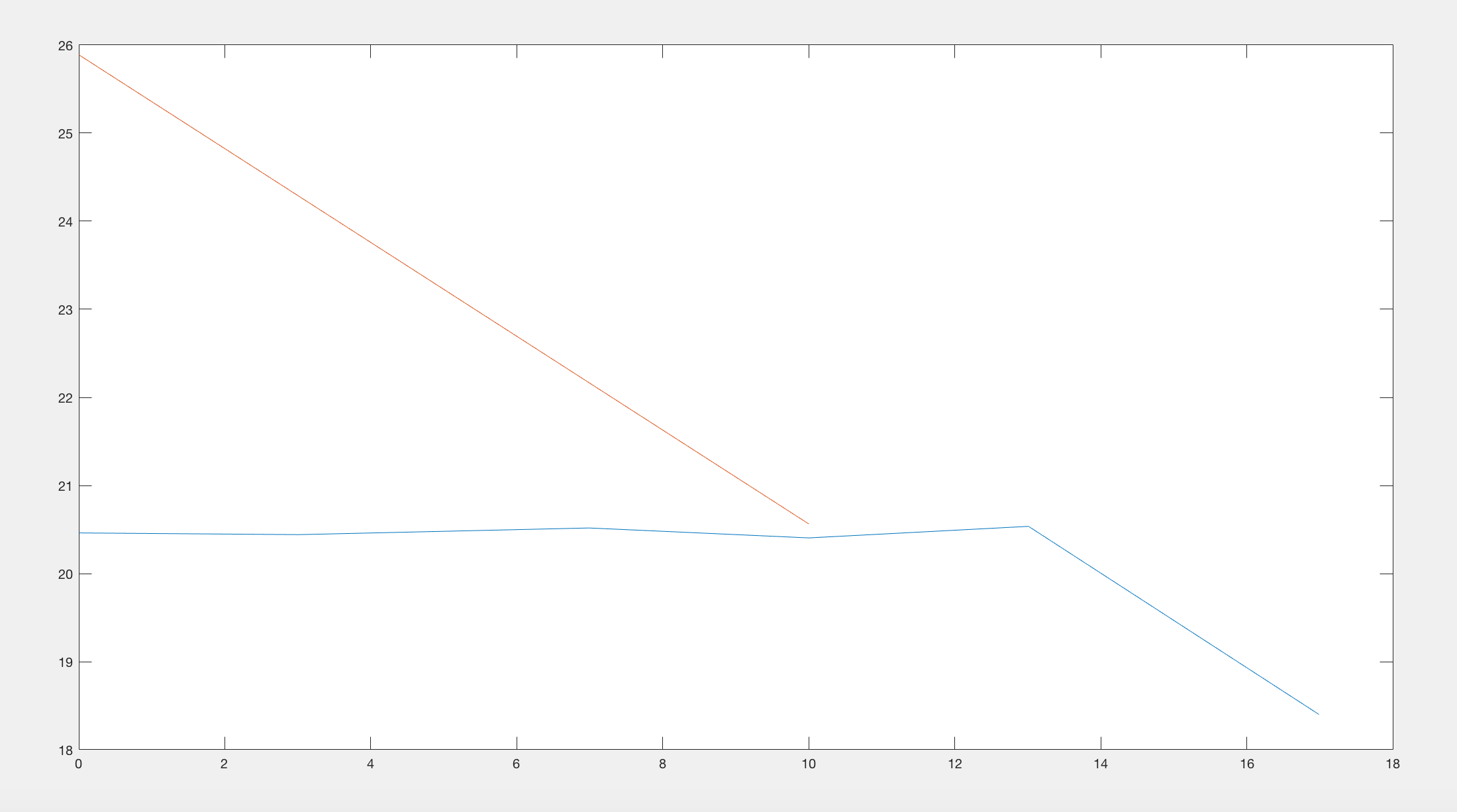


Figure 8: Bode Plot of position output with and without a proportional controller

We can see that applying a proportional controller changes the bode plot for the step response with position as output. Indeed, we can see that we went from an infinite gain margin to a finite one that we could thus control using our Kp.

**NOTE:** Going forward, we used a different value for Kp, because it seemed the machine’s behavior had changed and to continue with the experiment, we found the best fit value for Kp, under the supervision of the Laboratory TA.

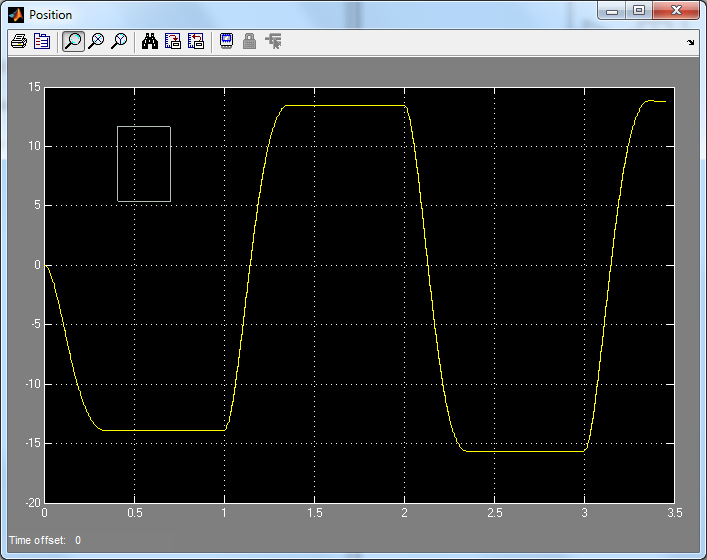


Figure 9: New step response of P, new KP = 0.05

7) Effect of PD controller

1. The differentiation operator was decreasing overshoot and settling time.

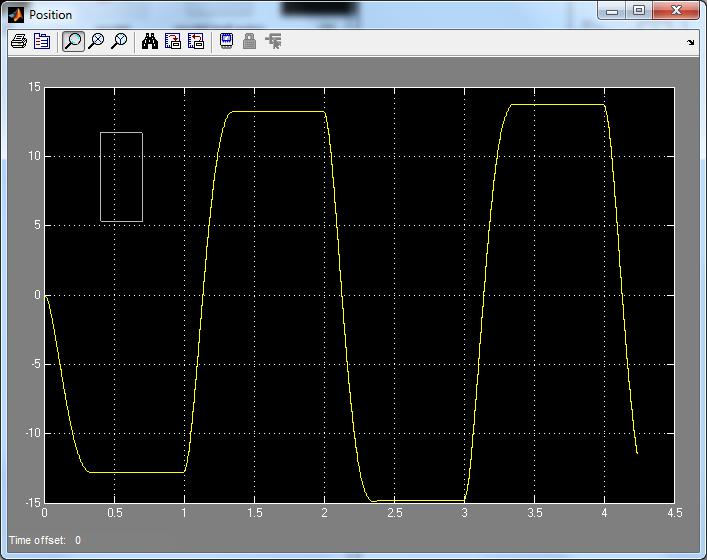


Figure 10: Step response of the system with PD with Kd= 0.00015

For our system, the best Kd is 0.00015.

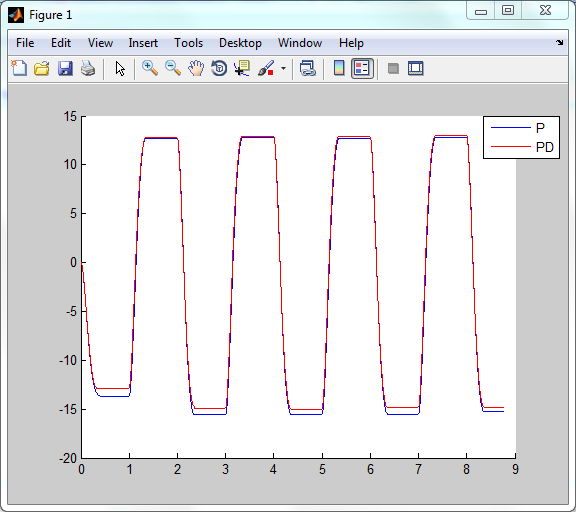


Figure 11: Step response of PD controller with best P controller

b) Tuning both KP and Kd, we get KP=0.051 and Kd=0.00009, we get the following response

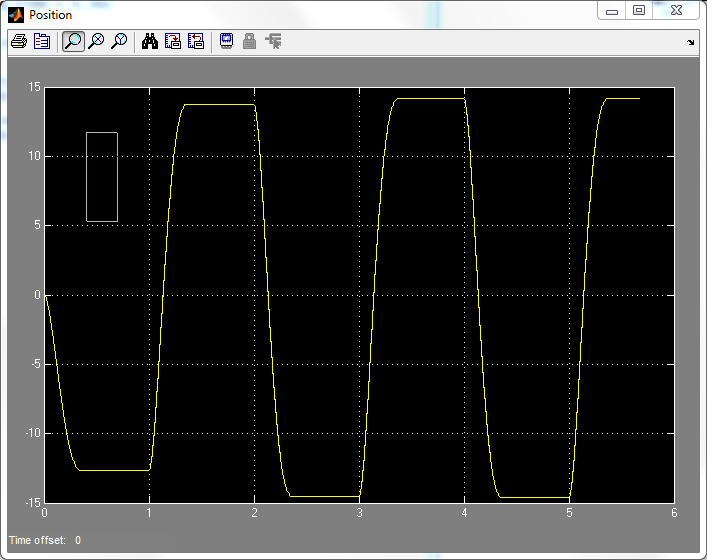


Figure 12: Step response of new PD controller

8) Effect of PI controller

1. The integrating operator was decreasing rise time and almost removed steady-state error. It was increasing overshoot and settling time.

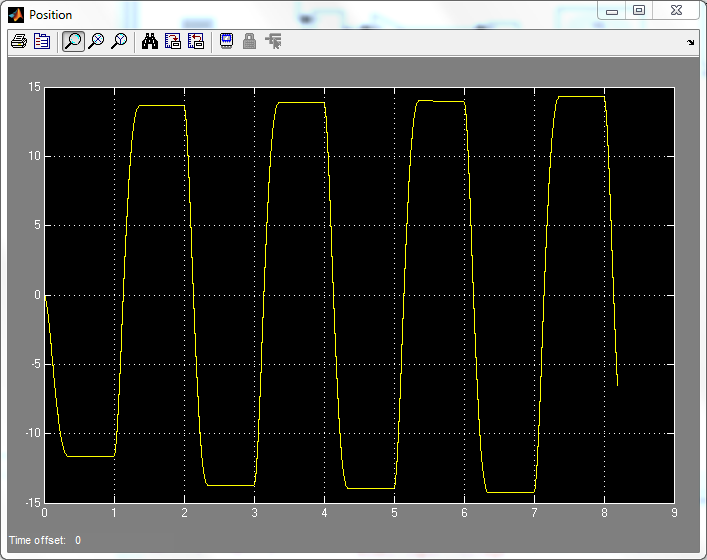


Figure 13 : Step response of the system with PI with KI= 0.003

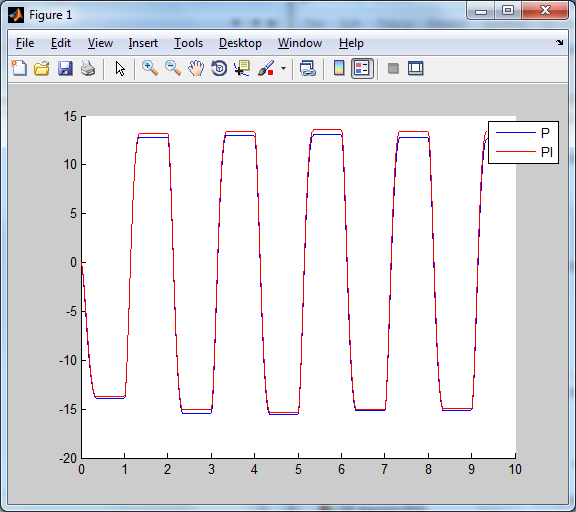


Figure 14: Step response of PI controller with best P controller

b) Tuning both KP and Ki, we get KP=0.051 and Kd=0.0035, we get the following response

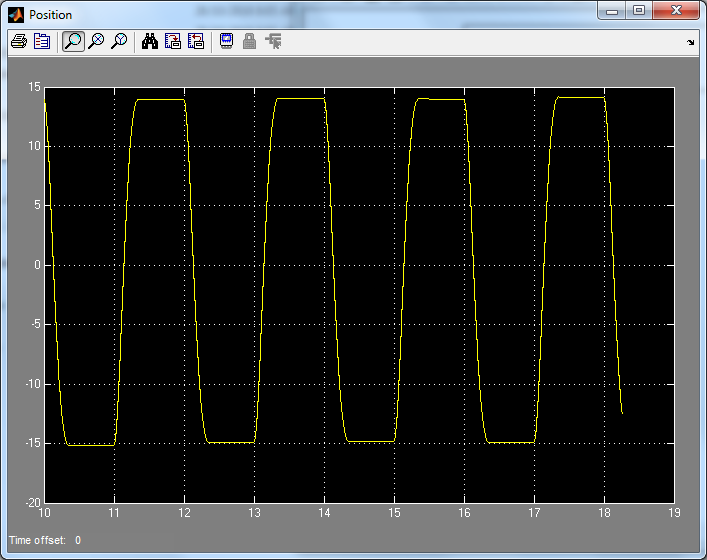


Figure 15: Step response of the system with PI with KI= 0.0035

9)

1. The integrating operator was just decreasing rise time and steady state error. For KP = 0.051 and Kd = 0.00009, best Ki = 0.00098.

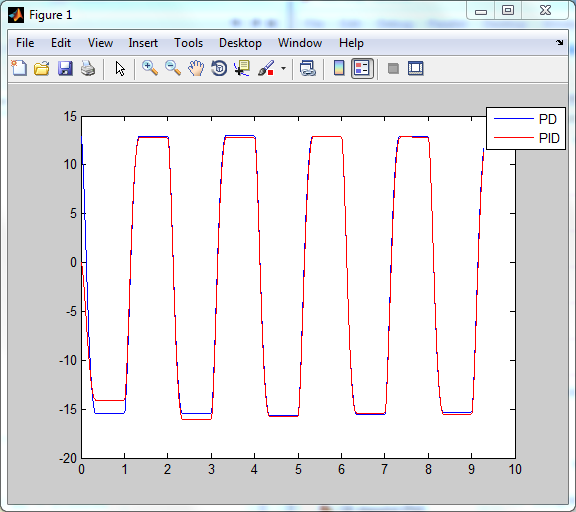


Figure 16: Step response of PI controller with best P controller

b) The differentiating operator was decreasing the settling time and overshoot. For KP = 0.051 and Ki = 0.0035, best Kd = 0.00015.

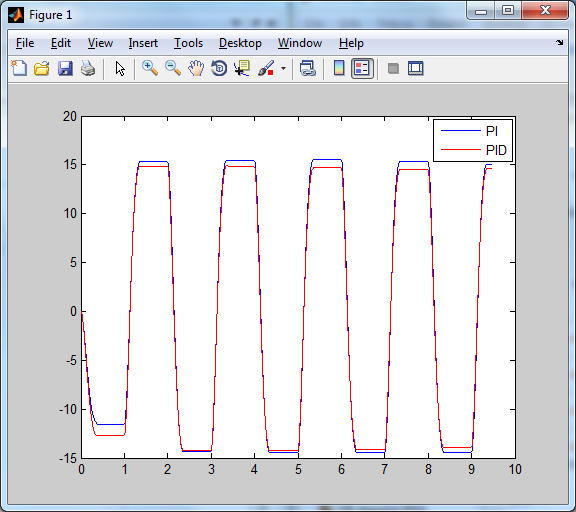


Figure 17 : Step response of PI controller with best P controller

c) Tuning KP, Kd and Ki, we get KP=0.052, Ki = 0.0038 Kd=0.0002, we get the following response

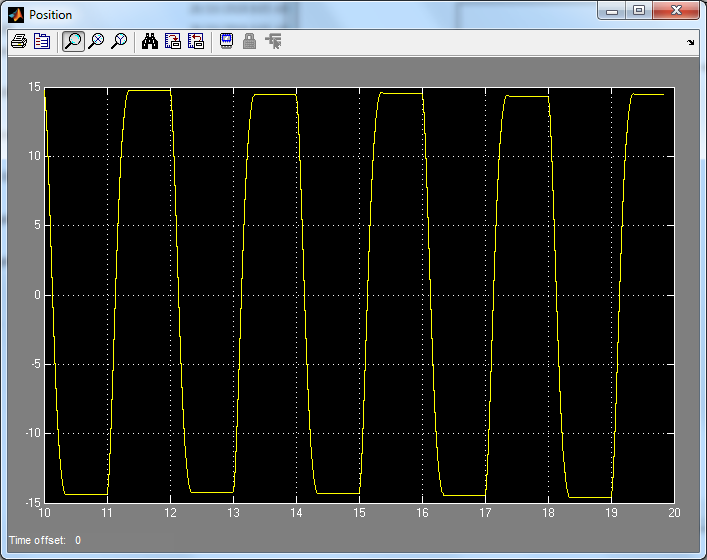


Figure 18: Step response of the system with PD with Kd= 0.00015

d)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Gain | Rise Time | Overshoot | Settling time | Steady state error |
| Kp | Decrease | Increase | Small Change | Decrease |
| Kd | Small Change | Decrease | Decrease | Small Change |
| Ki | Decrease | Increase | Increase | Decrease |

Appendix

Matlab script code used for Q7-Q9

%% Q7 a)

% time1 = (simout1.time);

H1 = (simout1.signals.values); %kp = 0.05

% H2 = (simout1.signals.values); %kp = 0.05, kd = 0.00015

% used to find length

% length(H1); length 4394

% length(H2); length 4837

% length(time1); length 4394

% H\_t = length(H1); %4394

%

% H3 = H1(1:H\_t); %extract data of set size H\_t

% H4 = H2(1:H\_t); %extract data of set size H\_t

% t = time1(1:H\_t);

%

% plot(t,H1,t,H4,'r');

% legend('P','PD');

%% Q8 a)

% time1 = (simout1.time);

% H1 = (simout1.signals.values); %kp = 0.05

% H2 = (simout1.signals.values); %kp = 0.05, ki = 0.003

% used to find length

% length(H1); length 4993

% length(H2); length 4681

% length(time1); length 4681

% H\_t = length(H2); %4681

% H3 = H1(1:H\_t); %extract data of set size H\_t

% H4 = H2(1:H\_t); %extract data of set size H\_t

% t = time1(1:H\_t);

%

% plot(time1,H3,time1,H2,'r');

% legend('P','PI');

%% Q9 a)

% time1 = (simout1.time);

% H1 = (simout1.signals.values); %kp = 0.051; kd = 0.00009

% H2 = (simout1.signals.values); %kp = 0.051; kd = 0.00009; ki = 0.00098

% used to find length

% length(H1); length 4910

% length(H2); length 4643

% length(time1); length 4643

% H\_t = length(H2); %4643

% H3 = H1(1:H\_t); %extract data of set size H\_t

% H4 = H2(1:H\_t); %extract data of set size H\_t

% t = time1(1:H\_t);

%

% plot(time1,H3,time1,H2,'r');

% legend('PD','PID');

%% Q9 b)

% time1 = (simout1.time);

% H1 = (simout1.signals.values); %kp=0.051, ki = 0.0035

% H2 = (simout1.signals.values); %kp=0.051, ki = 0.0035, kd = 0.00015

% used to find length

% length(H1); length 4744

% length(H2); length 4852

% length(time1); length 4744

% H\_t = length(H1); %2602

% H3 = H1(1:H\_t); %extract data of set size H\_t

% H4 = H2(1:H\_t); %extract data of set size H\_t

% t = time1(1:H\_t);

%

% plot(time1,H1,time1,H4,'r');

% legend('PI','PID')