MECH 304 Dynamic Modelling and Control Term Project

Ball Balancer

Taha Andaç, Arda Oruç, Ali Bora Albayraktar

Prof. Çağatay Başdoğan

04.06.2023

Koç University

College of Engineering, Department of Mechanical Engineering

MECH 304 Dynamic Modelling and Control Term Project
Ball Balancer

Abstract:

This report is the second part of a project which aims to optimize the control system by setting a settling time to less than 4 seconds and finding proper Kp and Kd values for marginally stable, overdamped, and underdamped situations. First, the disturbance signal is added to a ball balancer. Then, it is optimized by using Simulink and finding Kp and Kd values which reduce the settling time to less than 4 seconds. After that, we store *x(t)* and θ(t) data in MATLAB and use them to animate the movement of the ball and the beam. In the end, we find proper Kp and Kd values for marginally stable, overdamped, and underdamped situations and plot pole-zero maps and step responses. In this project, the model is developed by using MATLAB and Simulink programs.

Introduction:

The problem of this project is designing a ball balancer which uses a motor as an input source. A ball balancer is a complex system that uses a motor to balance the ball in a x-y plane by changing the angle of the beam dynamically (1 degree of freedom). Parameters that affect the ball balancer are motor current, motor constant, coefficient of friction of the ball and the beam, inertia of beam and shaft, the mass of the ball, and gravitational acceleration. Our goal is to reach stability and not drop the ball while the settling time is less than 4 seconds. To reach our goal, we design a control system according to the parameters above and add a PD controller to constrain the system's settling time.

Literature Survey:

There is a study about an image-based ball balancer which has 2 degrees-of-freedom. The aim of this study is to create a stable environment for the ball on a plate. The balance is reassured by tilting the plate. In this study, the camera is used to determine the location of the ball and calculates the ideal position to balance the system by image processing algorithm [1]. Two DC motors are used since the degree of freedom of the system is two and 2 PID controllers are used to change the angle of the motors according to the ball's position in x and y directions.

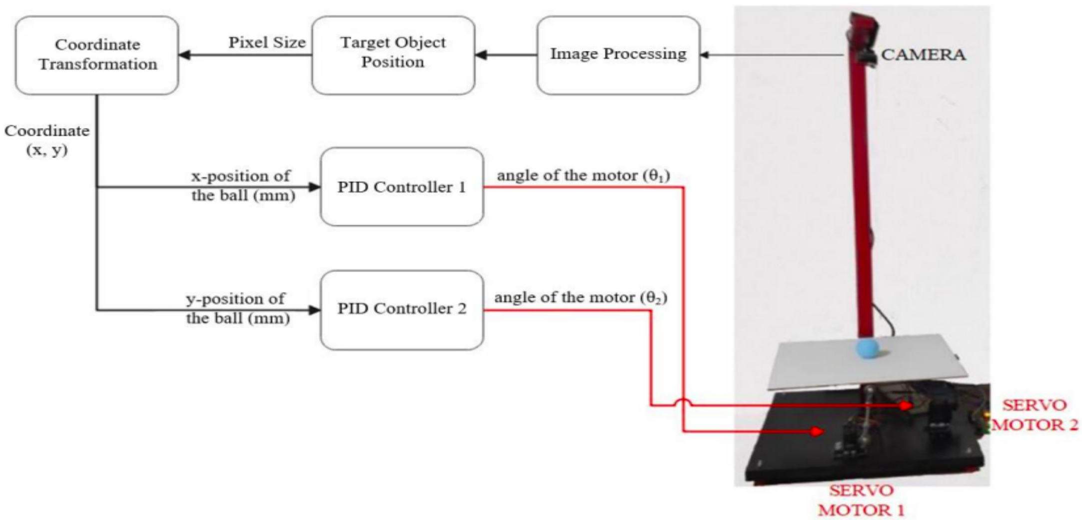

Fig. 1 Explanation of the system by general block diagrams. [1]

Another article is about a real-time 2 degrees-of-freedom ball balancer with a neural integrated fuzzy (NiF) controller. In this system, NiF is implemented as a PID controller and a hybrid controller is created. The transfer function of the plant model is similar to the previous study. The different part is the NiF controller between the plant model and PID controller. The NiF controller is able to solve nonlinear processes. With usage of the NiF controller in the system with the PID controller, steady state error, peak overshoot, and settling time decreased [2]. To sum up, using a NiF controller with a PID controller results in more accurate results.

Modelling Calculations:

Part c)

A Simulink model is developed to simulate the response of the disturbance signal. We added a disturbance signal and connected it to a θ(s) signal to a Simulink model we developed for the first part. To generate a signal as mentioned in the project description, we used "Signal Builder". The disturbance signal we created is the same disturbance signal in the description document. After adding the disturbance signal, the only thing left is optimizing the PD controller to have a settling time of fewer than 4 seconds. We did the optimization on the "Design parameter" and obtained the Kd and Kp values for a settling time of fewer than 4 seconds.

The Kd value is 360.5729 and the Kp value is 312.7208. The step for the selected PD controller is shown in Fig. 1. The last version of the Simulink model with the disturbance signal is shown in Fig. 2.

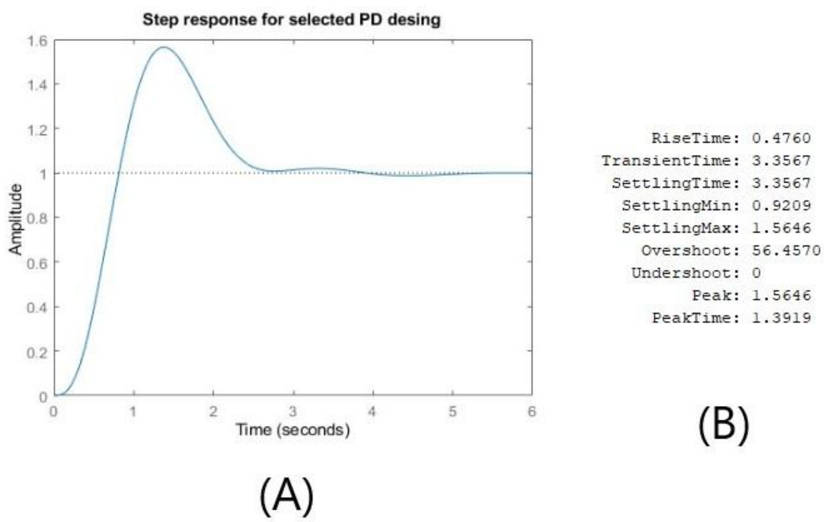

Fig. 1: (A) The step response for the PD controller. (B) Info of the step response.

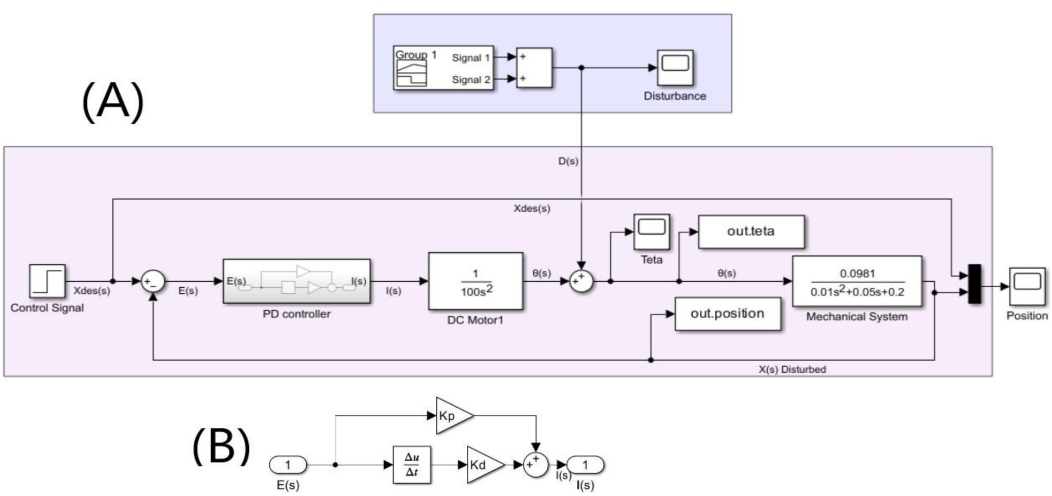

Fig. 2: (A) The last version of the system model. (B) The system of PD controller.

Part d)

We stored data (*x(t)* and θ(t)) from Simulink to MATLAB and used it to animate the movement. The position-time graph for the disturbed, non-disturbed, and step signals are shown in Fig. 3. The *x(t)* is calculated by using θ(t) and L. The animation of the ball on the beam is shown in MATLAB code since we cannot add video to the PDF document. We fed the system with an additional step input with -1 magnitude for 0.3 seconds to test it even further and for the simulation to look more appealing.

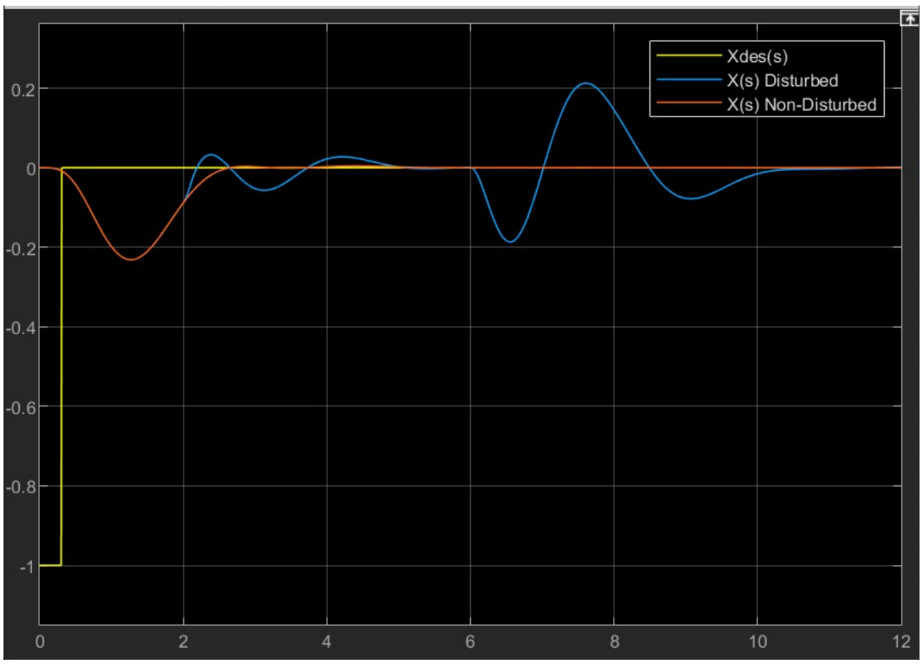

Fig. 3: The position-time graph for the disturbed, non-disturbed, and step signals.

Part e)

We determined the K_p and K_d values for the system graphically with the root-locus method by utilizing the Control System Designer toolbox of MATLAB, which you can see on table 1. You can see the pole-zero maps and step responses for marginal stability, underdamped case and overdamped case in figures 4, 5 and 6 respectively.

Table 1: K_p and K_d values for marginal stability and damping cases

	Marginally Stable	Underdamped	Overdamped
K_p	360.5729	250.689	0.47863
K_d	764.1925	593.9575	264.0243

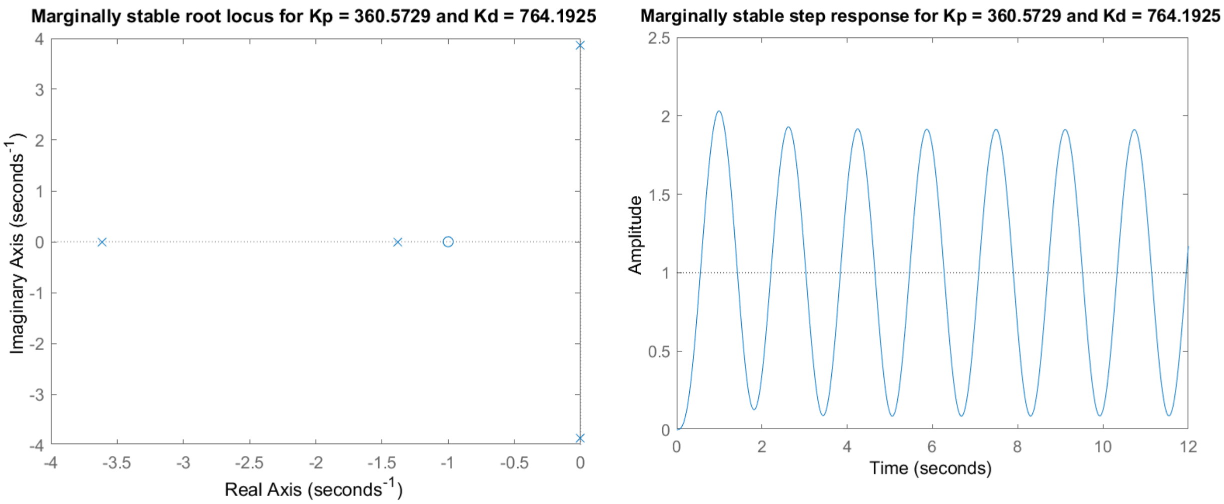

Fig. 4: Pole-zero map and the step response for marginal stability

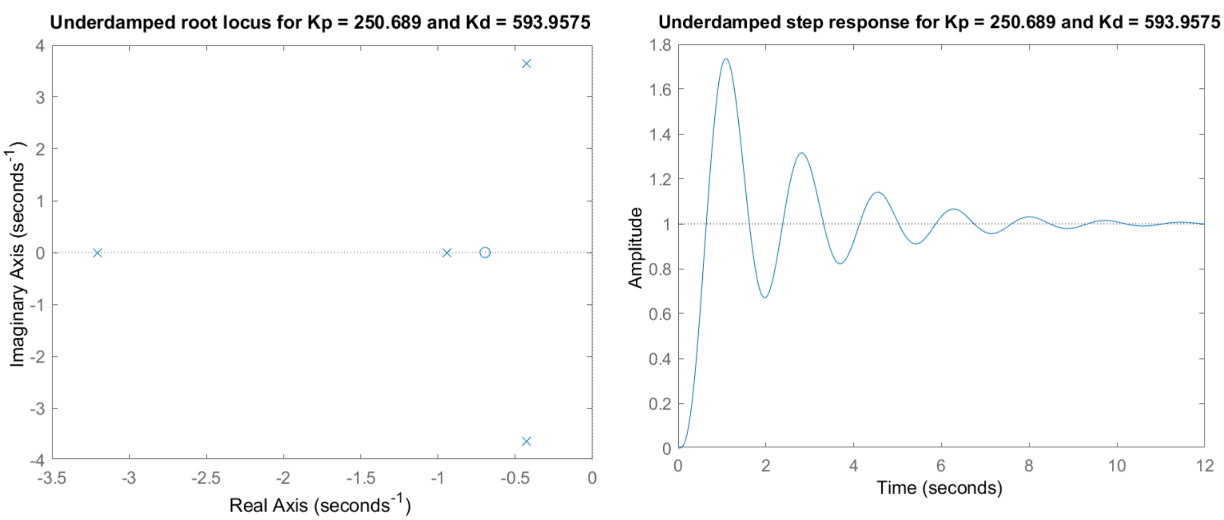

Fig. 5: Pole-zero map and the step response for underdamped system

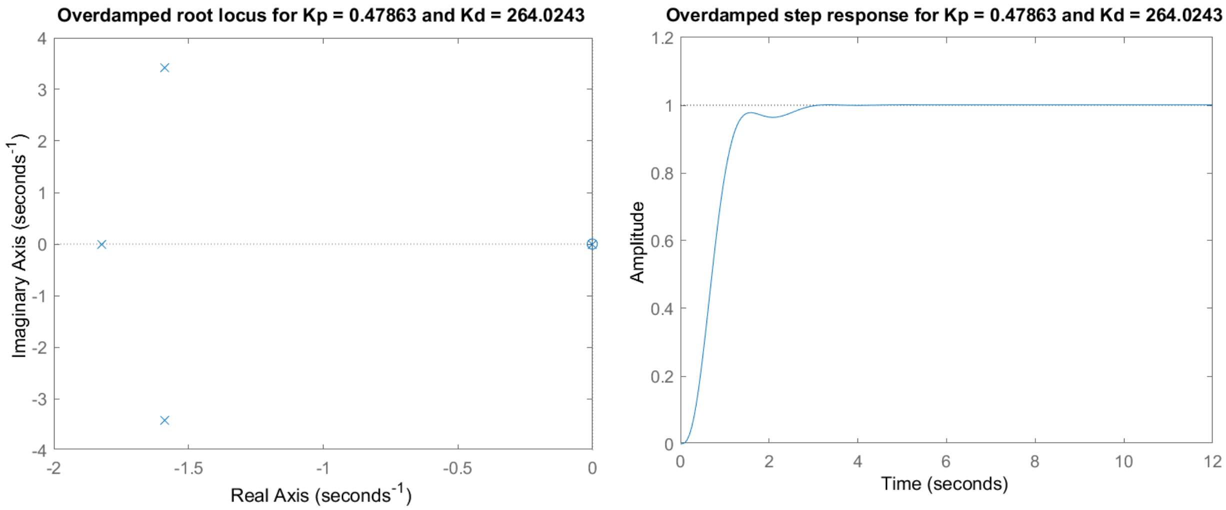

Fig. 6: Pole-zero map and the step response for overdamped system

Discussion and Conclusion:

Our main objective and the main criteria for success for this project is for the ball to not fall off from the beam. To evaluate whether the ball fell off from the beam or not, the animation code is not sufficient as it assumes that the ball stays on the beam and if the ball fell off the beam, the ball in the animation goes beyond the length of the beam but it stays on the alignment of the beam. This is problematic for cases in which the ball falls off with a very small offset. Instead, we need to look at the position versus time plots for each case. Our position zero is the center of the beam of 60cm, on which the ball initially stands at. Therefore, the position in the position versus time plots should always remain between 0.3m and -0.3m. If the position exceeds these boundaries, it means that the ball falls off.

For the K_p and K_d values we have in part c, we see from figure 3 that the position plot doesn't go out of the $\pm\ 0.3m$ interval. Therefore, the ball does not fall off the end of the beam. Therefore, we can say that our controller has been successful in dealing with the initial step input of 0.3 seconds with -1 magnitude and with the disturbance whose specifications have been given in the project description. If one were to look at the animation close enough, which is a bit hard because of the thickness of the beam and the ball, it can be seen that the ball's position doesn't exceed the length of the beam.

For the K_p and K_d values we have for marginal stability, available on table 1, we see from figure 7 that the position plot doesn't go out of the $\pm\ 0.3m$ interval. The system is challenged by the disturbance that starts at $t = 6s$ as the position value approaches the margins, but it doesn't go out of the interval. Therefore, the ball does not fall off the end of the beam. We can say that our controller has been successful in dealing with the disturbance whose specifications have been given in the project description, when the system is marginally stable. However, it is worth noting that the position data and the angle data in figure 7 is highly oscillatory, which means that it will take a long time for the system to settle, which is not ideal.

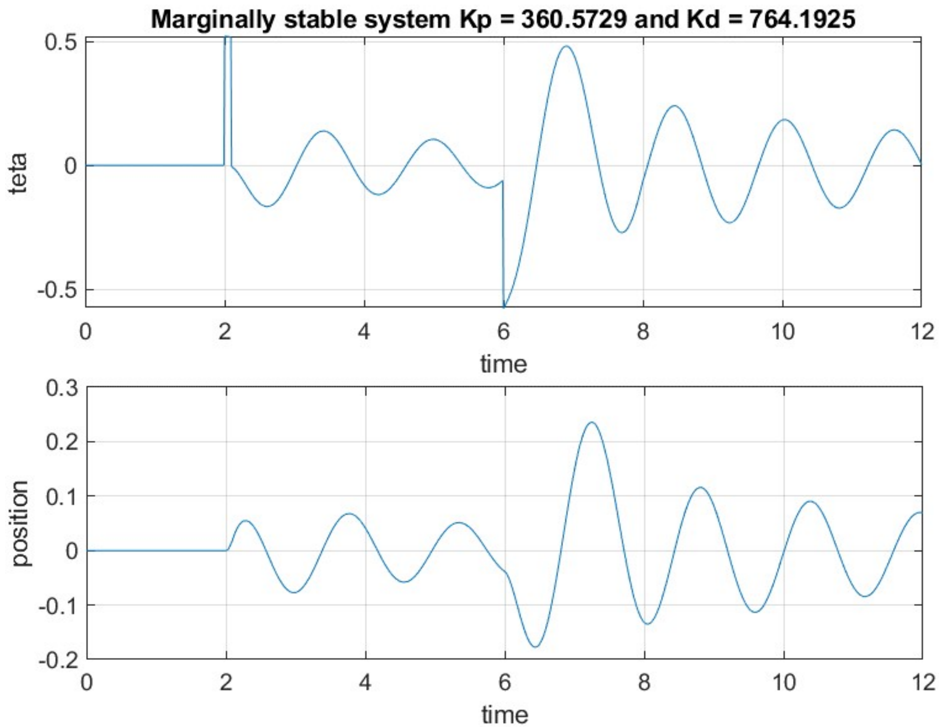

Fig. 7: Beam's angular position and ball's position in time for marginally stable case

For the K_p and K_d values we have for the underdamped system, available on table 1, we see from figure 8 that the position plot doesn't go out of the $\pm\ 0.3m$ interval. This time, the disturbance that starts at $t\ =\ 6s$ is less disruptive of the system as the magnitude of oscillation is less than the marginally stable case and doesn't get as close to the margins. Therefore, the ball does not fall off the end of the beam. We can say that our controller has been successful in dealing with the disturbance whose specifications have been given in the project description, when the system is underdamped. The position and the angle data in figure 8 are still a bit oscillatory but the magnitude of oscillations decrease more rapidly, meaning that the system will also settle sooner than the marginally stable case.

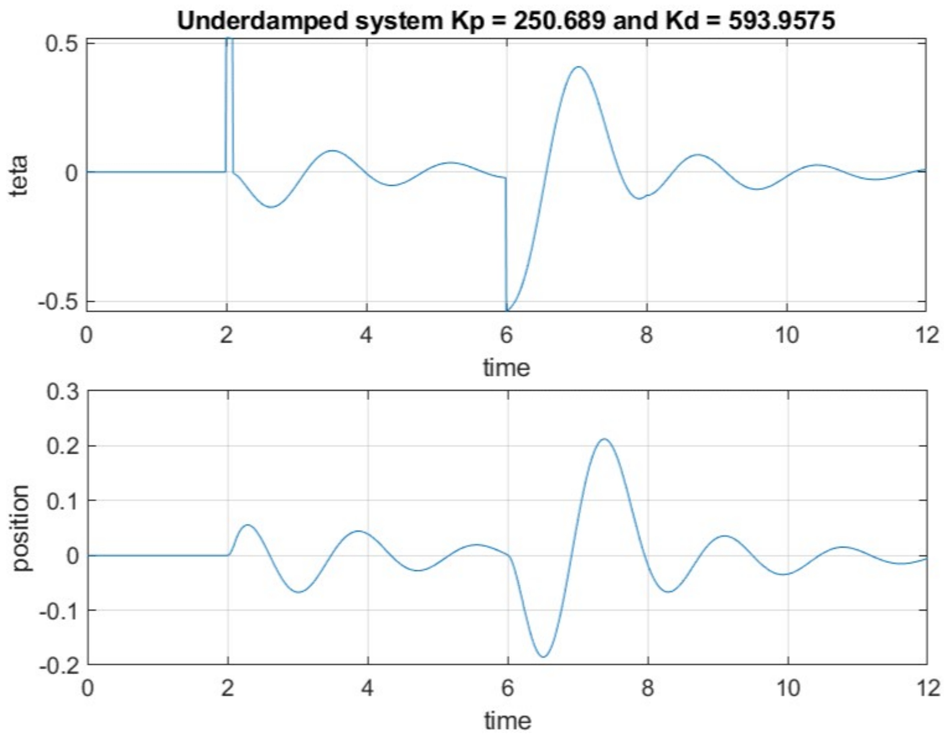

Fig. 8: Beam's angular position and ball's position in time for underdamped case

For the K_p and K_d values we have for the overdamped system, available on table 1, we see from figure 9 that the position plot doesn't go out of the $\pm\ 0.3m$ interval. This time, the disturbance that starts at $t = 6s$ is even less disruptive of the system as the magnitude of oscillation is even less than the underdamped case and doesn't even exceed $0.1m$ and $-\ 0.2m$ much. Therefore, the ball does not fall off the end of the beam. We can say that our controller has been successful in dealing with the disturbance whose specifications have been given in the project description, when the system is overdamped. The position and the angle data in figure 9 are only slightly oscillatory as they settle down to zero nearly right away. Therefore, we can say that the overdamped case is the fastest to stabilize the position of the ball on the beam.

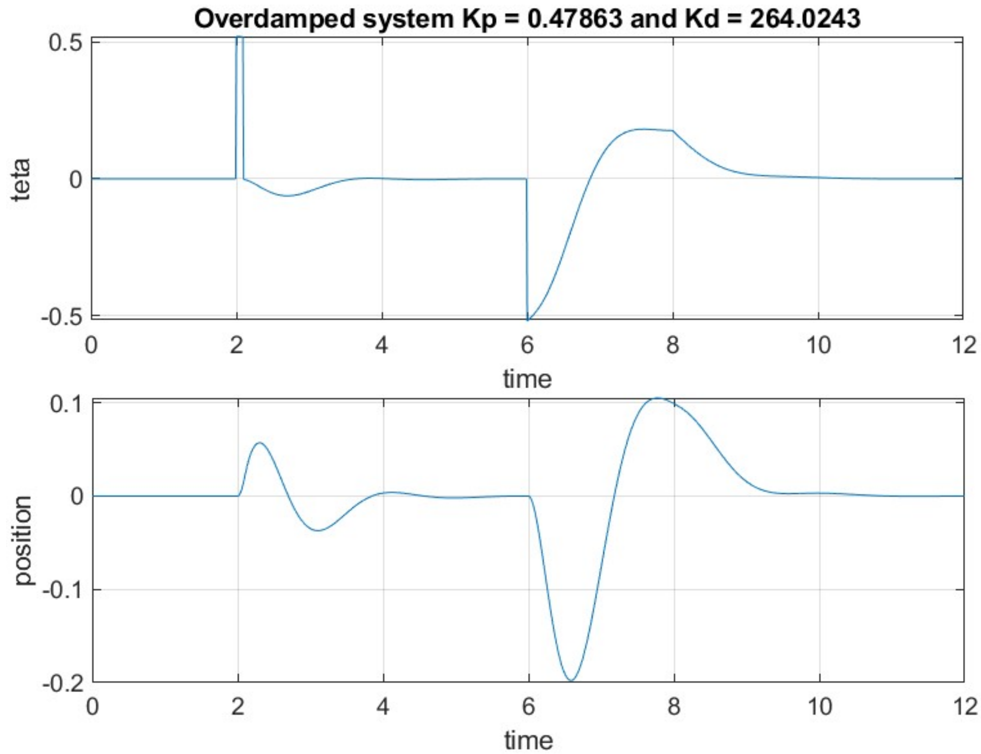

Fig. 9: Beam's angular position and ball's position in time for overdamped case

Overall, we think that we have succeeded in designing a capable control system for the ball balancer as we have succeeded in balancing the ball on the beam for all required cases. We even tested it out with an additional step input at part c, which was successful in testament to the success of our control system. This taught us not only how the control systems are designed in scope of a project task, but also how to test the limits of our designed controller by experimenting with our controller and its inputs, and how to optimize the parameters of our control system to meet desired control performance. We may improve the Simulink model by constructing a separate model but this time with a PID block to see whether the model without a PID block differs from the model with a PID block, and if they do, how much? In more complex projects, using the PID block in Simulink may be more time efficient. Maybe as a suggestion, a step of experimentation with the controller can also be added to the projects of the following terms.

Reference List:

[1] Gürsoy, H. C. and Adar, N. G., "Image Based Control of 2-Dof Ball Balancing System," Journal of Innovative Science and Engineering (JISE), Jun. 2022, doi: https://doi.org/10.38088/jise.1091154.

[2] Singh, R. and Bhushan, B., "Real-time control of ball balancer using neural integrated fuzzy controller," *Artificial Intelligence Review*, Sep. 2018, doi: https://doi.org/10.1007/s10462-018-9658-7.