École Polytechnique Fédérale de Lausanne

Semester Project

Identification and Control of a 2D Ball Balancer

Author:
Omar Hani Mokhtar Ahmed Meebed

Supervisor:
Dr. Alireza Karimi

Assistants:
Vaibhav Gupta
Zhaoming Qin

Automatic Control Laboratory
Mechanical Engineering
Switzerland

15 / 1 / 2023

Abstract

This project tackled the identification and control of the 2D Ball Balancer. Challenges are presented through a mathematical model, and include the non linearity, coupling between inputs and outputs and friction that amplifies both of them. The setup involves a lot of components to get the 2DBB to be functional, and required the design and fabrication of a PCB to connect the device to the microcontroller and dealing with computer vision to acquire ball coordinates. A first attempt at controlling the 2DBB used cascade control, with a proportional controller in the inner loop, and a PID controller at the outer loop that was tuned online. This configuration resulted in a stabilizing controller, but was observed to be not sufficient in tracking reference trajectories. To improve this performance, the outer controller was redesigned as a data driven controller. This controller required a frequency response model, which was obtained through spectral analysis method. Multiple models were computed to account for uncertainty in the ball starting position. The identified model is a MIMO system with cross talk components that were treated as disturbance to decouple both coordinates. Using the outer PID controller as an initial controller, the improved data driven controller is generated, with can track trajectories with a maximum frequency of 1.3 Hz.

Contents

1 Introduction

The 2D Ball Balancer (2DBB) is a multivariable system used to benchmark various control schemes, involving a ball that rolls on a plate. Since this system has non linearities, uncertainties and coupling between inputs and outputs, it is often used to mimic various applications in the robotic field. The goal of this project is to design and implement a data driven controller on the 2DBB, as well as validate its performance by comparing it to the classical PID method. Section 2 summarizes the main equations describing the mathematical model. Section 3 will cover the experimental setup and describe the circuit used to connect various components. Section 4 will cover the design of an initial PID controller through online tuning. Section 5 will cover the design of the data driven controller using a measured frequency response. Section 6 will provide final remarks.

2 Mathematical Model

The goal of this section is to have a better understanding of the physical system, that will be helpful in designing and validating the model that will be later obtained using system identification techniques. The mathematical model will also be used to highlight the challenges when trying to compute a more exact representation of the system.

To derive dynamical equations of the 2DBB, the following assumptions will be taken:

- The ball is always rolling and slipping is not considered.
- The ball is symmetric and homogeneous.
- Friction between ball and plate is negligible.
- The ball and plate are in contact all the time

The dynamical equations will be obtained using the Lagrangian, which are given by:

$$\frac{d}{dt}\frac{\partial L}{\partial \dot{q}_i} - \frac{\partial L}{\partial \dot{q}} = Q_i, \quad L = T - V \tag{1}$$

The kinetic energy of the system is given by the translation and rotation of the ball, as well as the rotation of the plate. The rotation energy of the plate will depend on the ball position on it, since the ball will change the mass distribution on the plate. The kinetic energy is then given as [1]:

$$\begin{aligned} T &= T_{ball} + T_{plate} \\ &= \frac{1}{2}m_b(\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2}I_b(w_x^2 + w_y^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(\dot{\alpha}x_b + \dot{\beta}y_b)^2 \end{aligned}$$

Where m_b is the ball mass, I_b and I_p is the moment of inertia of the ball and plate respectively, (x_b, y_b) the ball position and (α, β) the plate's inclination angles. Replacing the angular velocities

(w_x, w_y) by their expression for a homogeneous ball $w_x = r_b x_b$ and $w_y = r_b y_b$, where r_b is the ball radius gives the following expression for the kinetic energy:

$$T = \frac{1}{2}(m_b + \frac{I_b}{r_b^2})(\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2}(I_p + I_b)(\dot{\alpha}^2 + \dot{\beta}^2) + \frac{1}{2}m_b(\dot{\alpha}x_b + \dot{\beta}y_b)^2 \tag{2}$$

The expression for potential energy will only consider the gravitational force, given as:

$$V = m_b g(x_b \sin\alpha + y_b \sin\beta) \tag{3}$$

Knowing that this system has 4 degrees of motion (2 for ball positions and 2 for plate angles) and a torque is applied on the plate angles, the expression of the dynamical equation is given by [1]:

$$0 = (m_b + \frac{I_b}{r_b^2})\ddot{x}_b - m_b(x_b\dot{\alpha}^2 + y_b\dot{\alpha}\dot{\beta}) + m_b g \sin\alpha \tag{4}$$

$$0 = (m_b + \frac{I_b}{r_b^2})\ddot{y}_b - m_b(y_b\dot{\beta}^2 + x_b\dot{\alpha}\dot{\beta}) + m_b g \sin\beta \tag{5}$$

$$\tau_x = (I_p + I_b + m_b x_b^2)\ddot{\alpha} + 2m_b x_b\dot{x}_b\dot{\alpha} + m_b x_b y_b\ddot{\beta} + m_b\dot{x}_b y_b\dot{\beta} + m_b x_b\dot{y}_b\dot{\beta} + m_b g x_b\cos\alpha \tag{6}$$

$$\tau_y = (I_p + I_b + m_b x_b^2)\ddot{\beta} + 2m_b y_b\dot{y}_b\dot{\beta} + m_b x_b y_b\ddot{\alpha} + m_b\dot{x}_b y_b\dot{\alpha} + m_b x_b\dot{y}_b\dot{\alpha} + m_b g y_b\cos\beta \tag{7}$$

The expression for the torque is approximated by the transfer function of a DC motor relating the torque and the input voltage [2]:

$$\frac{T(s)}{U(s)} = \frac{K_t(Js + B)}{(R_a + L_a s)(Js + B) + K_b K_t} \tag{8}$$

K_t is the torque constant, J the moment of inertia of the motor and load, B the frictional coefficient, R_a the armature resistance, L_a the armature inductance and K_b the back emf constant.

From Equations 4 and 5, it is possible to notice the coupling terms that depend on $\dot{\alpha}$ and $\dot{\beta}$. It is also possible to notice the non linear dependency of x on α and y on β. Equations 6 and 7 show the effect of an input torque on the ball and plate system, which describes complicated non linear dynamics. When considering the effect of ball friction in addition to the existing equations, the ball dynamics become much more chaotic. Assuming that the friction follows a simple Coulomb model given by:

$$\mathbf{F_s} \leq \mu_s \mathbf{F_n} \qquad (\dot{x}, \dot{y}) = 0$$

$$\mathbf{F_d} = -\mu_d \mathbf{F_n}\, sgn(\begin{bmatrix}\dot{x} \\ \dot{y}\end{bmatrix}) \qquad (\dot{x}, \dot{y}) \neq 0$$

This entails that if the ball is moving along the x direction, the friction force follows the dynamic friction model along the y direction as well, which can cause the ball to move in the y direction if $\mu_d \mathbf{F_n} < F_s < \mu_s \mathbf{F_n}$ and thus amplifies the coupling effect already present in the system. This effect should be considered when evaluating the ball dynamics.

3 System Setup

3.1 Main Components

The 2DBB used in this project is shown in Figure 1. It contains:

- A babyfoot ball
- A 2 degree of freedom plate that can swivel around its center point, with maximum angular displacement at $\pm 45°$
- 2 rotary servo base units connected to the plate, each equipped with a DC motor with 6V as nominal voltage, an encoder with 4096 counts per revolution and a potentiometer, which will not be used
- A digital camera with a resolution of 640×480 pixels, mounted above the plate

Figure 1: Ball & Plate System

To actuate the motors and get relevant information on the position of the ball, a myRio 1900 device is used as the microcontroller that transmits measurements to the PC and sends the actuation input to the servo base unit. As the input from the microcontroller has not sufficient current to move the motors, a VoltPAQ-X4 Amplifier is used as a relay between the microcontroller and the actuator. To have a secure connection between the microcontroller, encoders and actuator, a simple printed circuit was designed. This is explained in section 3.2

3.2 PCB Design

The connection between the pins of the microcontroller, encoders and motor is done with a printed circuit board, designed with EAGLE (Figure 2). The components of this board are the following:

- 2 34-pin headers to connect the microcontroller expansion ports (MXP)

- 2 phono connectors (more specifically the FC68371) that will send signals to the amplifier, which is transmitted to the motor
- 2 MAB 5SH connectors directly plugged in to the encoders, which will obtains information on the motor speed

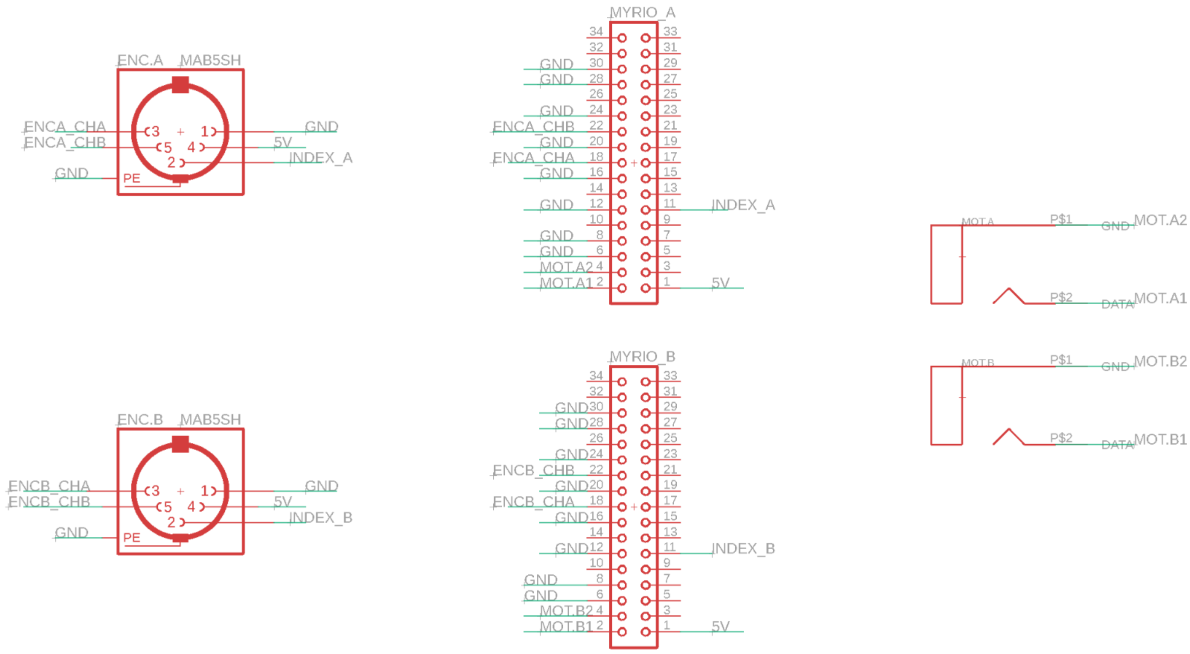


Figure 2: PCB circuit

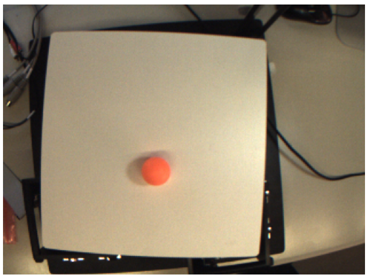
The encoders are connected to the corresponding encoder pins on the microcontroller (pins 18 and 22). Encoders are supplied with 5V and both are connected to the ground. The index is connected on a digital input pin (pin 11) to be used if necessary. To be able to turn motors in both directions, both pins are connected to analog outputs (pins 2 and 4), where it will be possible to set the amplitude by varying the voltage difference and the direction by selecting which pin outputs the higher voltage.

It is important to note that the method used to turn motors in both directions is not optimal, since it limits the input voltage to ± 2.5 V, while the motors have a nominal voltage of 6V. A quick solution is to increase the amplifier gain from 1x to 3x. This will enable the motor to perform on the entirety of its spectrum, but will have consequences in regards to output absolute accuracy, which will become ± 150mV [3]. This should be taken into consideration when interpreting final results. Output amplification will also have an effect on its step size, which will become approximately 3.66 mV. However such a step size is sufficiently small for this case.

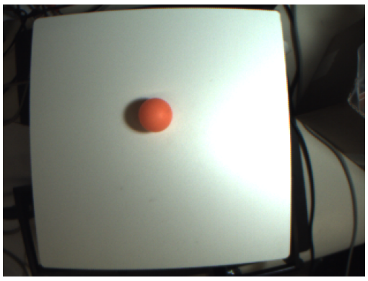
3.3 Image Processing

The goal of the image acquisition is to extract the position of the ball and the plate. This is done using the Vision Assistant module in LabView. The initial image is shown in Figure 3a. From this image, it is possible to observe that the camera is capturing the space surrounding the plate. This space is not useful to get ball and plate position and should be removed. This is done with a physical calibration of the position of the camera, which in this case is just lowering the camera and making sure it is covering the whole plate in all possible orientations. To improve image quality, the lenses were also adapted accordingly to obtain a crisp image that is showing the least amount of pixelation. A light source was also also added next to the system to have consistent lighting at any time of the day. The image obtained after this step is shown in Figure 3b.

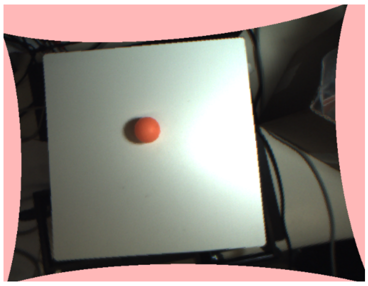
From Figure 3b it is possible to observe that the supposedly flat plate has curved corners. This is actually a consequence of lens distortion. To remove this distortion, image correction is done using a calibration grid to create a distortion model. The result is shown in Figure 3c. Afterwards, the plate coordinates are obtained. More specifically, the bottom left and top right corner coordinates are obtained with edge detection to get plate edges, followed by finding the intersections between them (Figure 3d). The ball is obtained by color matching using a sample ball image, and is shown in Figure 3e with the red square around the ball.



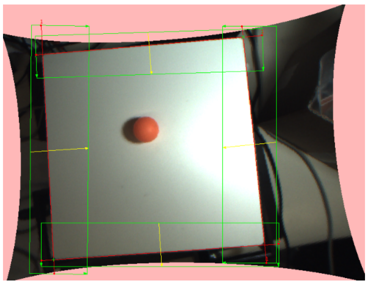
(a) Initial ball and plate image



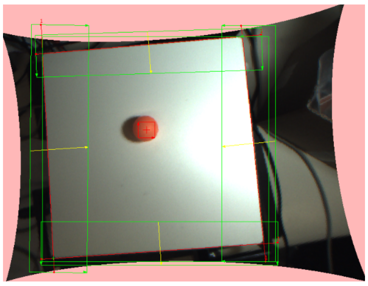
(b) Physical calibration of camera and lenses



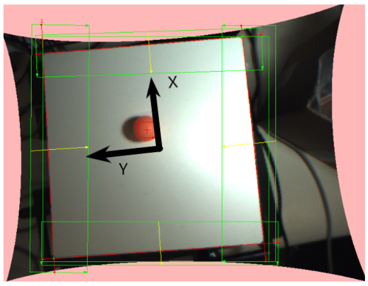
(c) Software calibration of lens distortion



(d) Plate edge detection



(e) Finding ball through sample image



(f) Adjusting ball coordinates using plate position

Figure 3: Image processing steps

Since the plate will be moving when the system is active, absolute coordinates are not the best way to describe the position of the ball. Instead coordinates relative to the plate position will be used. In this case, the origin of the coordinates will be taken at the center of the plate, and the axes will be chosen to align with the angular direction of the motors, as shown in Figure 3f.

3.4 Implementation

Combining all previous elements, the chart in Figure 4 summarizes the workflow of the system. The setup has two programs running in parallel: The image processing portion (Section 3.3) is run on the computer, which allows the heavier calculations to be done quickly and enabling the image acquisition to be done at approximately 17 ms (60 FPS) instead of 34 ms (30 FPS) if it was done on the microcontroller. After computing a moving average to reduce effect of noise, this program transmits the ball coordinates to the microcontroller, which will then use this along with encoder position to compute the control input. The control input is transmitted to the motors through the printed circuit and the amplifier.

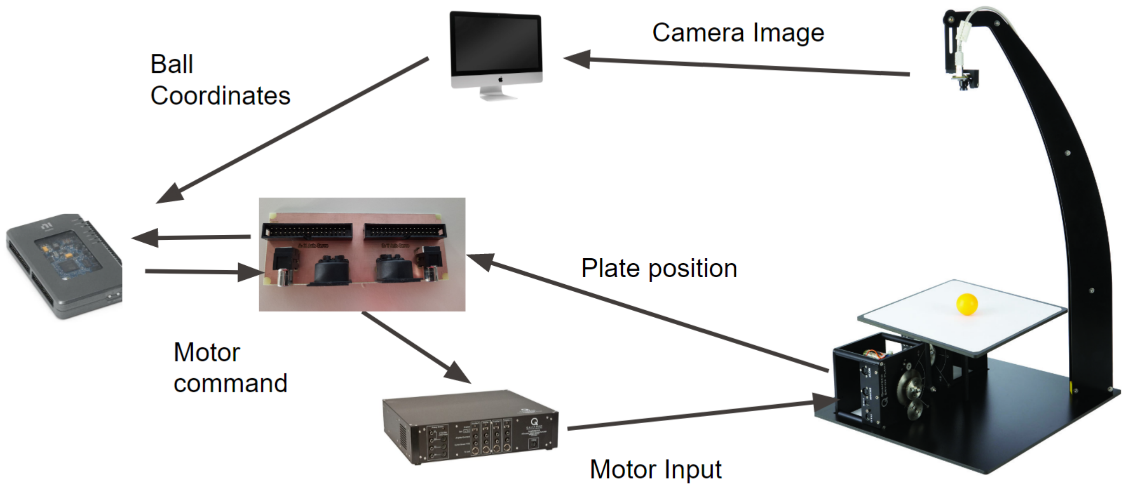


Figure 4: B&P system workflow

More specifically, the second program is working as follows:

- Obtain ball and plate positions. When powering up the microcontroller, it is necessary to have the plate flat as to obtain correct starting position
- Obtain reference or target position that is set before start of experiment (Either for a certain coordinate or a trajectory)
- Compute the control input
- Send the control signal. This signal will be further constrained between -2V and 2V, so that the motor is never beyond the nominal voltage of 6V (knowing that the amplifier has a gain of 3)
- Wait for new ball coordinates

4 Initial Controller

4.1 Controller Objectives

Since the system is inherently unstable, the design of a stabilizing controller is necessary. The goal of the stabilizing controller is to prevent the system from becoming unstable during the identification process. On top of that, it is necessary to tune the controller such that the input remains below saturation limits as to remove potentially nonlinear behavior. The controller should also be able to position the ball at the origin, so as to obtain a similar starting position for each identification test.

4.2 Controller Structure

The chosen control structure for this system is cascade control. This will result in separating the system into two portions: the inner portion is the position control of the plates, while the outer portion is the position control of the ball (Figure 5). This choice is motivated by the complexity reduction of the overall system, especially the inner controller, which has been reduced to the position control of a DC motor and can be easily tuned with a simple controller. Another motivation is the fact that the bandwidth of the encoder is much greater than the bandwidth of the camera. Implementing this inner control loop that will run at a much higher frequency will take advantage of the encoder bandwidth.

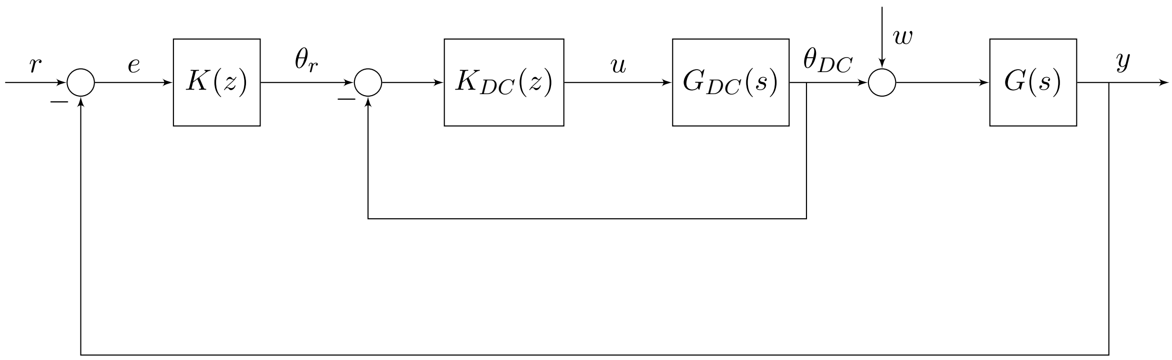


Figure 5: Cascade control structure for 2DBB

To implement this control structure, the second program that is computing the control input is modified. It will now run two loops in parallel, with one running at 500 Hz, corresponding to the inner loop controller K_{DC}, while the outer loop controller K will run at 60 Hz. The input to K will be ball position, and will output plate angles. The input to K_{DC} will be plate angles received from the outer controller, and will output voltage to be sent to the motor. Since the inner loop will be running much faster than the outer loop, the plate angles provided to the inner loop will be kept the same until new coordinates are provided by the outer loop controller.

4.3 Design of K_{DC}

G_{DC} is a closed-loop stable system, which implies that the controller should only be chosen to meet objectives mentioned above. A first approach is to design the controller so that saturation should only occur when the required angle difference is above 35°. This choice is made knowing that a reasonable nominal plate angle should be around 17°. Using the knowledge that the DC motor encoder has 4096 counts/rev, the saturation limit is at 2V and choosing a proportional control structure, K_p is given by:

$$\begin{aligned} K_p \times \Delta\theta &\leq 2\text{V} \\ K_p &= \frac{2}{35 \times 4096/360} \approx 0.005 \end{aligned}$$

K_{DC} is then defined as:

$$K_{DC} = K_p \times e[k] = K_p \times (\theta_r - \theta_{DC}) \tag{9}$$

4.4 Design of K

K was chosen as a discrete PID controller that can be easily implemented and tuned. Since $G(s)$ is unstable, the primary goal of this controller is to stabilize the ball on the plate. This was achieved through a PD controller, where the derivative term was added to compensate for the enormous overshoot in the ball position, which was the main source of instability. However the addition of this derivative term amplified measurement noises, thus introducing a new source of instability. This was solved by adding a low pass filter in the derivative term, with a time constant of $\tau = 0.1$. The derivative term can be then expressed as:

$$D(s) = K_d \times \frac{s}{\tau s + 1}$$

Applying Tustin's method to discretize the derivative portion gives:

$$D(k) = \frac{1}{2\tau + T_s} \times \left[2K_d(e[k] - e[k-1]) + (2\tau - T_s)D[k-1]\right]$$

Where T_s is the sampling time. To obtain a behavior where the ball returns to the origin, an integrator term with a small K_i is added to the other terms. It is given by the following expression:

$$I[k] = \frac{K_i T_s}{2} \times (e[k] + e[k-1]) + I[k-1]$$

The issue with this term is that it is very unlikely that the ball will stabilize at the origin exactly, which will always induce an integrator action that will always be moving the ball in a bounded interval, meaning that the ball will never fall off the plate, but will never stop moving. This is the effect of the friction between the ball and the plate, where if the plate is slightly inclined the ball will not move, but if the plate is inclined a bit more the ball will move quickly, since it has switched from static to dynamic friction.

To be able to have a stationary ball and driving the ball around the origin, an alternative solution is to consider the origin as a bounded set. This will imply that if the ball is within a certain distance from the origin, the integrator action will stop and the ball will attempt to stabilize inside this set. One disadvantage of this method is that in some cases, the ball will become stationary outside the set, which will reactivate the integrator term and moving the ball away from a stationary state, repeating the stabilization process again, which may seem to lead to an unstable result. In the end, it comes down to a tradeoff between oscillation time before the ball stops moving and the radius of the set of points considered to be near the origin. In this particular case, the radius of 1.5 cm was taken as the radius of the uncertainty set.

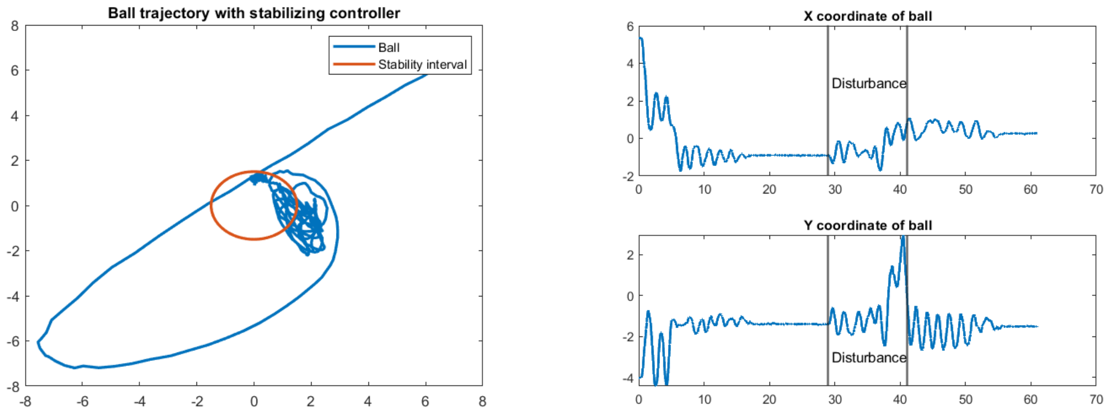
Rearranging terms, the controller K is given by:

$$K(z) = \begin{cases} \frac{55.43z^2 - 110z + 54.47}{0.217z^2 - 0.4z + 0.183} & r > 1.5 \\ \frac{55.42z - 54.58}{0.217z - 0.183} & r \leq 1.5 \end{cases} \tag{10}$$

Where r is the distance from the origin, $K_p = K_d = 25$, $K_i = 2$.

4.5 Initial Controller Performance

Applying the control structure described in this section gives the following result when trying to bring the ball to the origin and disturbance rejection once it is inside the stability zone



(a) Ball trajectory when going to the origin (b) Ball trajectory when rejecting disturbance

Figure 6: Performance of stabilizing controller

It is possible to observe the overall predicted behavior from Figure 6a. The ball goes to the proximity around the origin, with some attempt to stop the ball movement inside the target set. The system is also robust to external disturbance as seen from Figure 6b, where after 30 seconds, a variable disturbance is introduced for 10 seconds. The ball is able to reenter the stable set after being forced out by an external disturbance. The settling time in this case is 20 seconds, which is a good first attempt for regulation, but is not suited when trying to follow a trajectory. The next step would be to design another controller that gives a much smaller settling time.

5 Data Driven Controller Design

To obtain a better tracking performance, the controller $K(z)$ will be redesigned as a data driven controller. This choice was made knowing that the outer loop is the one having all the nonlinear and uncertain behavior. To achieve this, a more exact model of the system is necessary, which will be determined through system identification methods, that will identify the transfer function between θ_r and y. Using the controller determined in the previous section as the initial stabilizing controller, a more advanced controller will be determined through H_∞ objectives.

5.1 Identification Process

The 2DBB is a MIMO system with 2 input and 2 output. It is considered as a black box since crucial information on friction and saturations are neglected in the mathematical model, and the goal is to calculate the frequency response of G. The chosen procedure is to use a single channel excitation for each input separately while the other input is set at 0. By observing the output of this excitation input, this allows to then compute each of the four transfer functions of the entire MIMO system.

The excitation input is chosen to be a PRBS signal. To obtain all relevant information on the system, two PRBS signals will be applied: The first PRBS has 31 samples, with an amplitude of ± 0.5 cm, applied for 8 periods. This enables to obtain an estimate on high frequency behavior. Since the tracking objectives lie at low frequency, the estimate does not have to be precise. The second PRBS has a frequency divider of 5 applied to the first PRBS input. This enables to sample lower frequency signals with a large number of samples, and completely discarding higher frequency information, where the first PRBS has a richer density in this area.

To account for variable starting ball positions (Figure 7), they should be incorporated as a model uncertainty. More specifically, a total of three tests will be done for different ball positions. Each test consists of a first step of moving the ball to the stability zone around the origin, then the excitation signal is applied after the ball is stable. This method eliminates some transient effects occurring at the start, and the first two periods are discarded from calculations of the frequency response to eliminate all remaining transient effects

The model uncertainty will be given as a multi-modal uncertainty, where each model is computed with a different starting ball position. Other unknown uncertainties values include the plate starting position, which will depend on the device startup. Plate lighting will also vary slightly between daytime or nighttime. These uncertainties were unfortunately not considered in the identification process, but they were controlled as to remain constant throughout the testing duration.

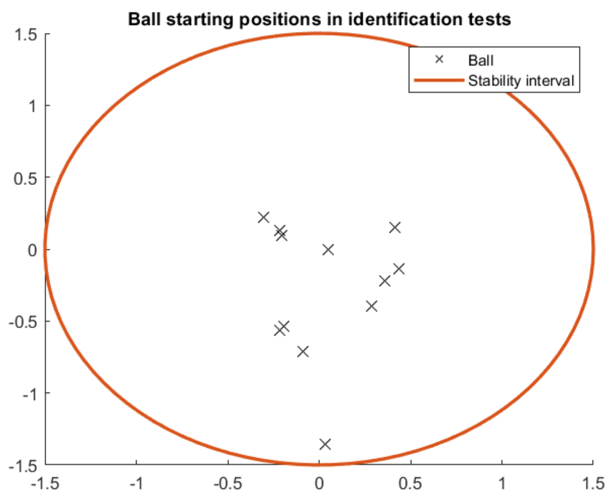


Figure 7: Ball Starting position

As the stabilizing controller is inherently nonlinear with the addition of the dead zone of the integrator around the origin, the response of the system when excited with a periodic signal is aperiodic, which made it impossible to completely remove truncation error. Instead a Hann window of size 31 and 155 (for the first and second excitation signal respectively) were applied on the response data. The frequency response was calculated with the spectral analysis method. Results are shown in Figure 8

It is observed that all three transfer functions have a similar behavior, which makes it possible to consider a nominal model out of the three and use it to compute the controller. Another observation is that the main components have an amplitude comparable to the coupled components, which indicates that cross talks components cannot be ignored when designing the controller. The fact that the system starts at -40dB is a result of the units of the input angles, which were

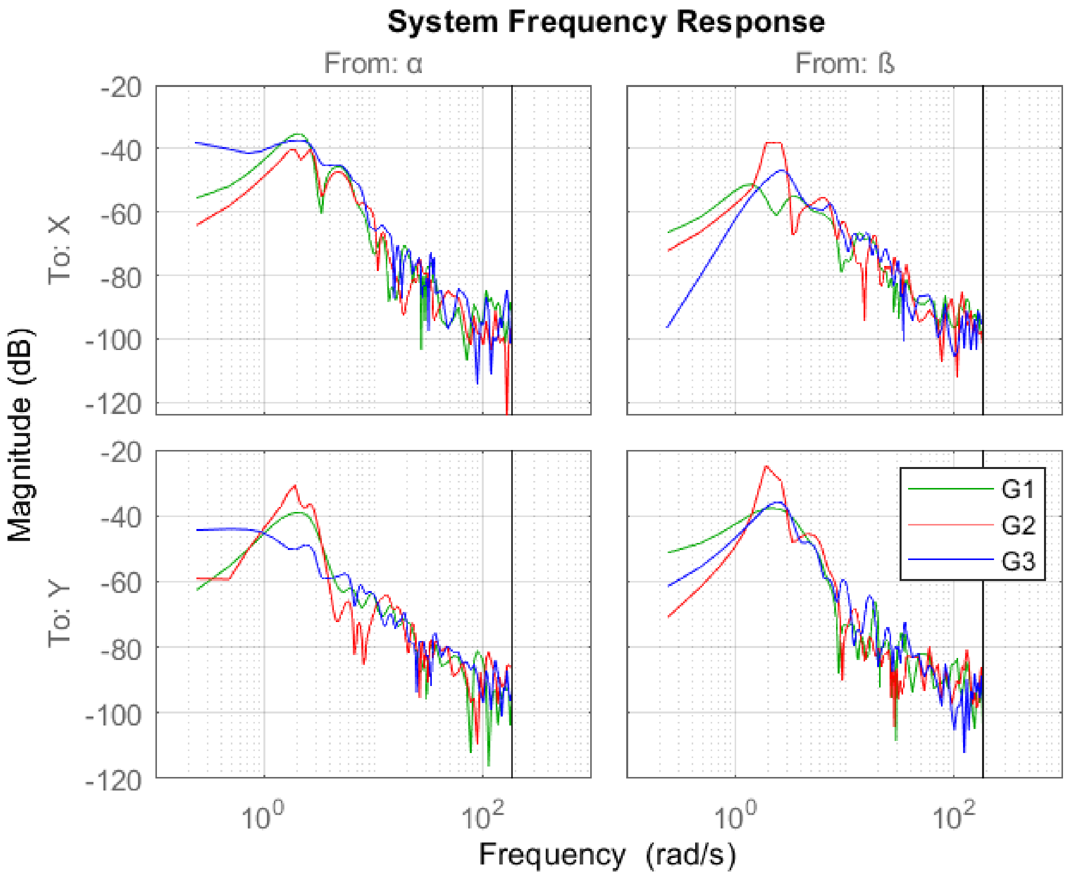


Figure 8: Measured frequency response of G

left as the encoder output varying between ± 300.

5.2 Controller Design

Based on the frequency responses obtained from identification, a multivariable controller should be designed, since cross talk components cannot be ignored. In this project however two independent SISO controllers were designed for each component, and cross talk components will be treated as disturbance to be rejected in order to have a decoupled system. Another objective is to have good tracking performance of reference trajectories at low frequency, and have decreased gain at high frequency. A third objective is to limit the input magnitude as to prevent saturation. This point is treated as an objective instead of a constraint to allow the input to saturate at low frequency such that the tracking performance can be achieved. These objectives are associated with weighting filters as follows:

$$\min \left\| \begin{bmatrix} W_1\mathcal{S} \\ W_3\mathcal{U} \\ W_4\mathcal{V} \end{bmatrix} \right\|_\infty$$

In order to satisfy stability for all models, the multimodal uncertainty is converted to a multiplicative uncertainty and the filter W_2 is computed using the `ucover` function in Matlab, and the constraint is set as:

$$\|W_2\mathcal{T}\|_\infty < 1$$

Other elements of the data driven controller are determined as follows:

- The stabilizing controller is chosen as the PD controller with parameters identical to the previous section:

  $$K_{init}(z) = \frac{55.42z - 54.58}{0.217z - 0.183}$$

  The integrator term was removed since it involves non linear terms such as dead zones that are not essential to have stability.

- The controller structure is a 5th order controller with no fixed part, as no poles in the initial controller lie on the unit circle [4]. This means that it is not possible to have an integrator in the controller, since stability is not guaranteed.

- The optimization problem is sampled using a frequency grid of points equivalent to those identified in the frequency response, since other points will only be an estimation by linear approximation using these known points.

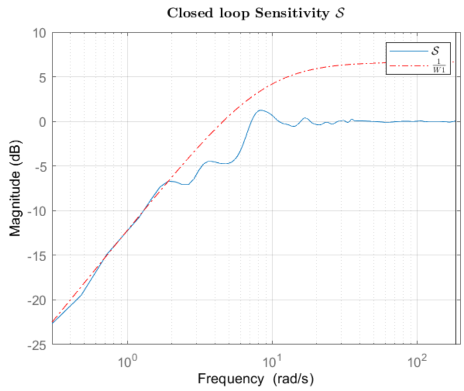
5.3 Performance in Simulation

Due to time constraints, it was not possible to test the improved controller on hardware. Nonetheless, the predicted performance is shown for both controllers, along with the choice of weighting filter for each of them.

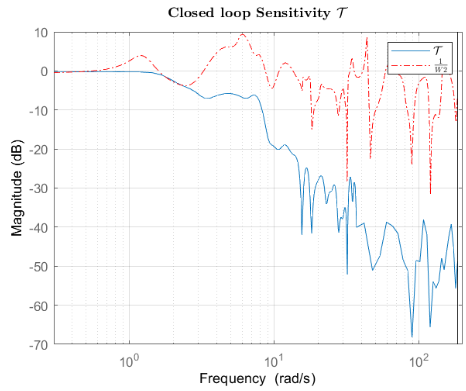
5.3.1 X coordinate

The results of the data driven control optimization problem described above and the weighting filters used for each component is shown in Figure 9. The optimization was done using the G_3 as the nominal model, and the algorithm converged to a solution in 16 iterations. Looking at Figure 9b, the system is able to guarantee a good tracking performance up to $w = 1.3$Hz, and beyond that stability constraints forces the system to reduce its input and can no longer achieve a good tracking performance. Disturbance rejection and input limitations are respected, as seen from Figures 9d and 9c and the expression of this improved controller is given by:

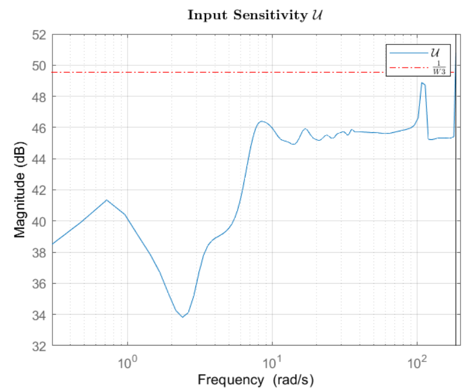
$$K_{X,improved} = \frac{-197.9z^5 + 70.74z^4 + 137.8z^3 + 103.2z^2 + 59.08z - 174.2}{z^5 - 0.3541z^4 - 0.6143z^3 - 0.6051z^2 - 0.384z + 0.9584} \tag{11}$$



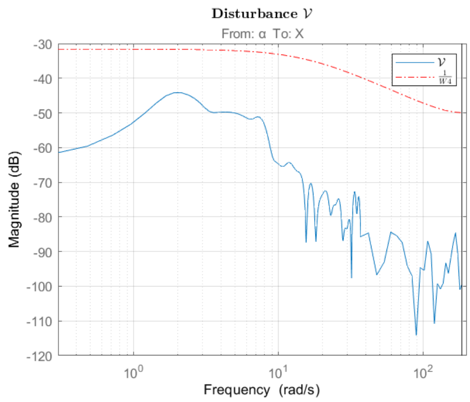
(a) Closed loop Sensitivity S for improved controller on X coordinate



(b) Closed loop Sensitivity T for improved controller on X coordinate



(c) Input Sensitivity U for improved controller on X coordinate



(d) Disturbance V for improved controller on X coordinate

Figure 9: Closed loop functions and weighting filter for the improved controller on X coordinate

5.3.2 Y coordinate

In the Y coordinate, it was unfortunately not possible to achieve robust performance since the weighting filter W_2 is forcing the system to be below 0dB at low frequencies. This is the result of a large error between the identified models, as shown in Figure 10. An alternative is to compute the weighting filter using only two transfer functions and results are shown in Figure 11. This guarantees similar results as in the X coordinates, with a maximum tracking frequency of $w = 3$Hz and the nominal model G_1, but stability is not guaranteed for G_2. The controller is given by:

$$K_{Y,improved} = \frac{-270.3z^5 + 162.2z^4 + 73.03z^3 + 78.35z^2 + 193.2z - 244.6}{z^5 - 0.7057z^4 - 0.2974z^3 - 0.2913z^2 - 0.7012z + 0.9958} \tag{12}$$

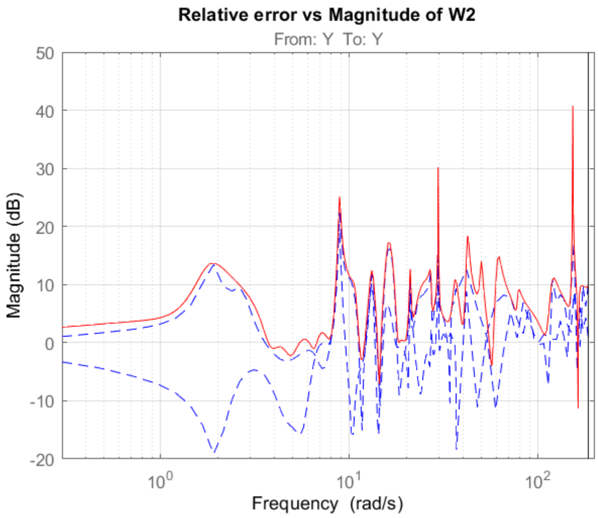
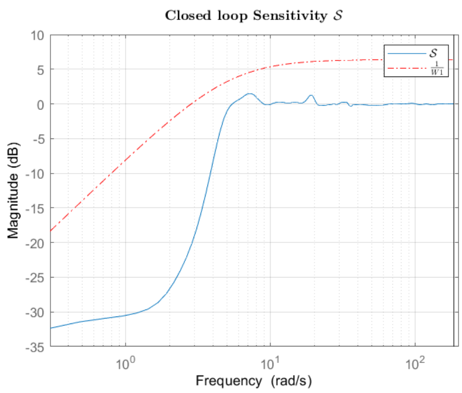
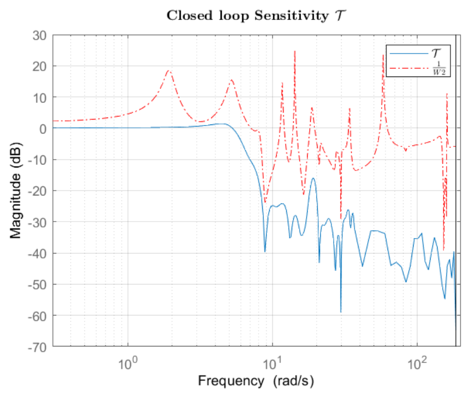


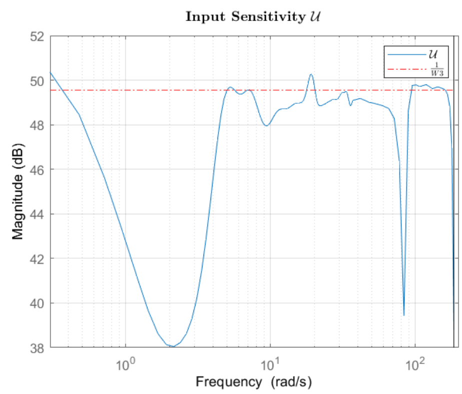
Figure 10: Weighting filter W_2 for Y coordinate considering all models



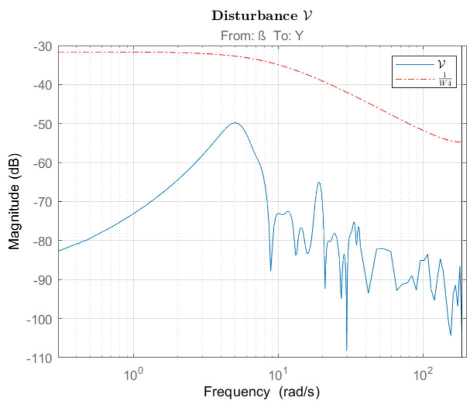
(a) Closed loop Sensitivity S for improved controller on Y coordinate



(b) Closed loop Sensitivity T for improved controller on Y coordinate



(c) Input Sensitivity U for improved controller on Y coordinate



(d) Disturbance V for improved controller on Y coordinate

Figure 11: Closed loop functions and weighting filter for the improved controller on Y coordinate

6 Conclusion

In conclusion, this project was successful is designing a controller to track trajectories with low frequency, despite the challenges in the non linearity, coupling between inputs and outputs and friction that amplifies both of them. The setup to test the controller was presented, and involved the design and fabrication of a PCB to connect the device to the microcontroller and a computer vision software to acquire ball coordinates. A first attempt at controlling the 2DBB used cascade control, with a proportional controller in the inner loop, and a PID controller at the outer loop that was tuned online. By using this structure, the complexity of the overall system was reduced. This then made it easier to design a data driven controller, by using the outer controller as an initial controller. This procedure required a frequency response model, which was obtained through spectral analysis method. Multiple models were computed to account for uncertainty in the ball starting position. The identified model is a MIMO systems where cross talk components that were treated as disturbance to decouple both coordinates. The improved controller can track reference trajectories with a maximum frequency of 1.3 Hz.

This project has several areas where further exploration is needed. The design of a data driven MIMO controller for the 2DBB should be explored, since it can take into account cross components models, will will surely lead to a better controller. A second area of exploration lies in the system identification method, which in this paper used a black box model. Another approach is to exploit the structure presented in the mathematical model in the identification process, thus employing a grey box identification method. This approach can lead to less uncertainty, which can give a less constrained weighting filter when computing the controller. Another area of improvement is related to the PCB design, where it can be useful to use a component that can generate positive and negative voltages to reduce quantization interval.

References

[1] Mohammad Nokh, Daniel Khashabi. *Modelling and Control of Ball-Plate System*. Amirkabir University of Technology, 2011. URL: `https://danielkhashabi.com/files/2011_LinearControl/16.pdf` (visited on 01/15/2023).

[2] Electrical Engineering. *Transfer Function of Armature Controlled DC Motor - Latest 2023*. Section: Control systems. May 14, 2017. URL: `https://www.electricalengineeringinfo.com/2017/05/transfer-function-armature-controlled-dc-motor.html` (visited on 01/15/2023).

[3] National Instruments. *myRIO-1900 Getting Started Guide and Specifications - NI*. Mar. 11, 2022. URL: `https://www.ni.com/docs/en-US/bundle/myrio-1900-getting-started/resource/376047d.pdf` (visited on 01/15/2023).

[4] Alireza Karimi. *Advanced Control Systems*. Lausanne, Apr. 2021.

[5] Alireza Karimi. *System Identification*. Lausanne, Sept. 2019.

Appendix

```matlab
% Generate plots for stabilizing controller performance
clear; close all; clc;
%% XY plots
fileID = fopen('logsTEST2.bin');
data = fread(fileID, 'double');
fclose(fileID);

N = length(data)/6;
data = reshape(data,6,N);

start = 160;
stop = 1800;
y_11 = data(1,start:stop);
y_12 = data(2,start:stop);

r = 1.5;
xc = 0;
yc = 0;
theta = linspace(0,2*pi);
x_c = r*cos(theta) + xc;
y_c = r*sin(theta) + yc;

figure;
plot(y_11,y_12,'LineWidth',2)
hold on
plot(x_c,y_c,'LineWidth',2)
xlim([-8 8])
ylim([-8 8])
title('Ball trajectory with stabilizing controller')
legend('Ball','Stability interval')
hold off

%% Separate X and Y
fileID = fopen('logsTEST1.bin');
data = fread(fileID, 'double');
fclose(fileID);

N = length(data)/6;
data = reshape(data,6,N);
Ts = 0.017; % Sampling time

start = 1;
stop = 3588;
N = stop-start+1;
y_11 = data(1,start:stop);
y_12 = data(2,start:stop);

time = 0:Ts:Ts*(N-1);

```

```
figure;
subplot(2,1,1);
plot(time,y_11,'Linewidth',2)
title('X coordinate of ball')
xline(29,'Linewidth',1.5)
xline(41,'Linewidth',1.5)
text(29.5,3,'Disturbance')

subplot(2,1,2);
plot(time,y_12,'Linewidth',2)
title('Y coordinate of ball')
xline(29,'Linewidth',1.5)
xline(41,'Linewidth',1.5)
text(29.5,-3,'Disturbance')
```