GUC

Ball Balancing Plate Project

Mostafa Mohamed Abdelaziz 58-22509
Marwan ElSayed 58-20308
Ali Emad 58-13891

Group Number: T34
Date: 13/3/2024

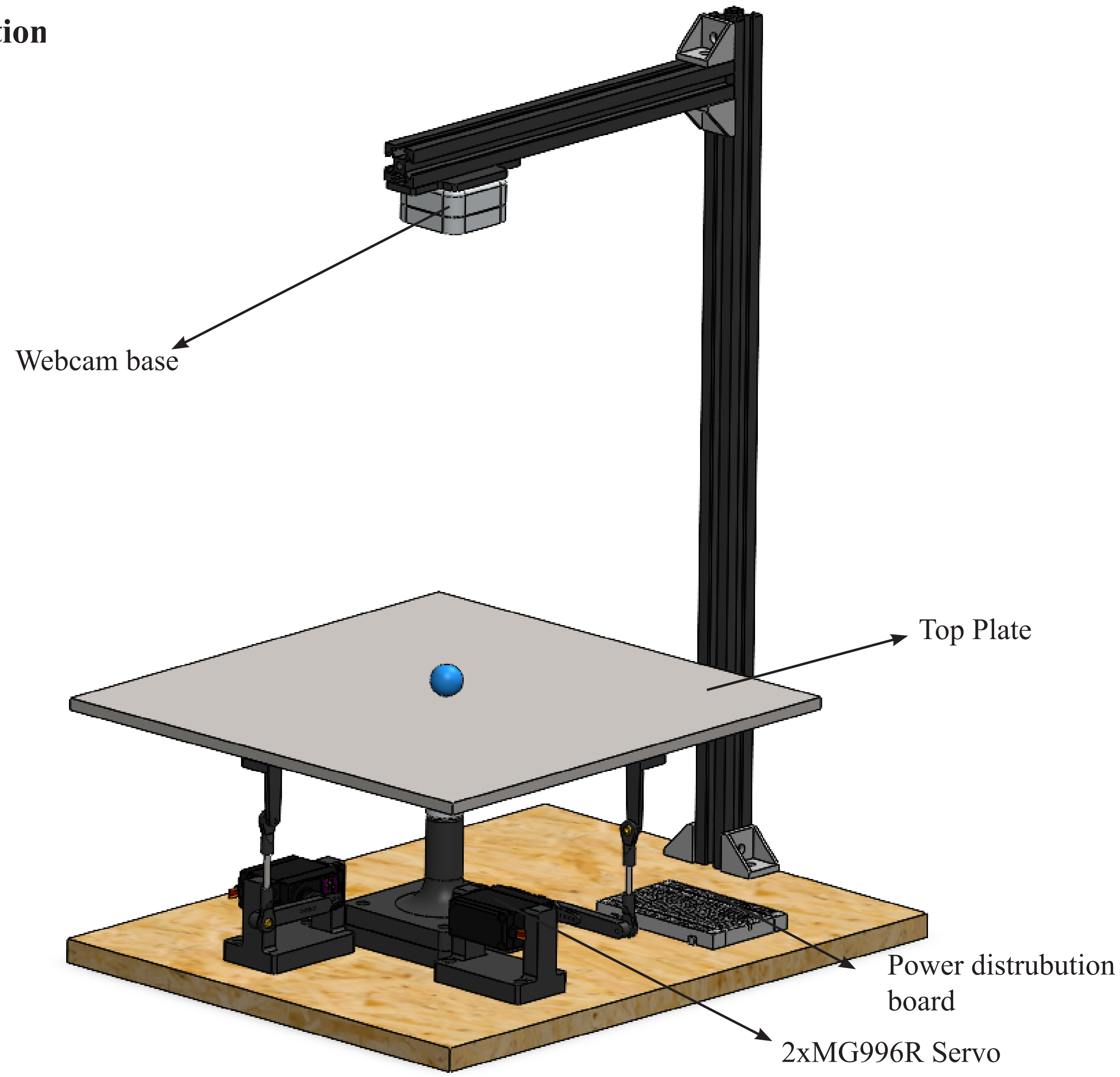
Project Description

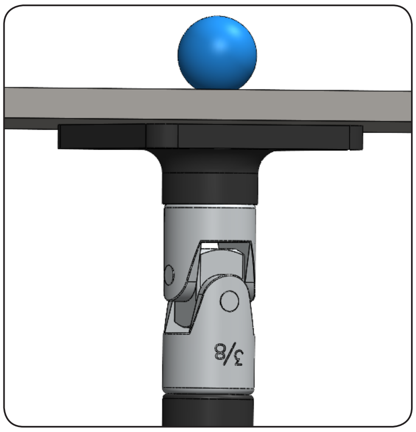
The Ball Balancing Plate project is a multidisciplinary endeavor that integrates concepts from mechanical engineering, control systems, and embedded systems. It presents several challenges, including:
Mechanical Design: Ensuring a rigid yet lightweight platform capable of precise tilt control.
Control Theory: Developing a robust control algorithm to stabilize an inherently unstable system.
Embedded Systems: Implementing real-time processing for sensor feedback and actuator control.

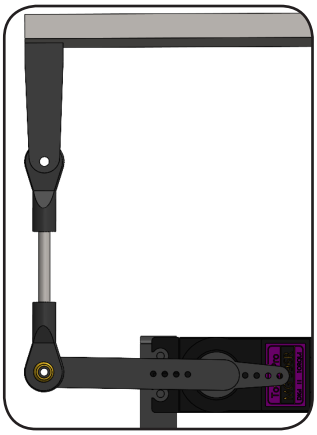
Working Principle

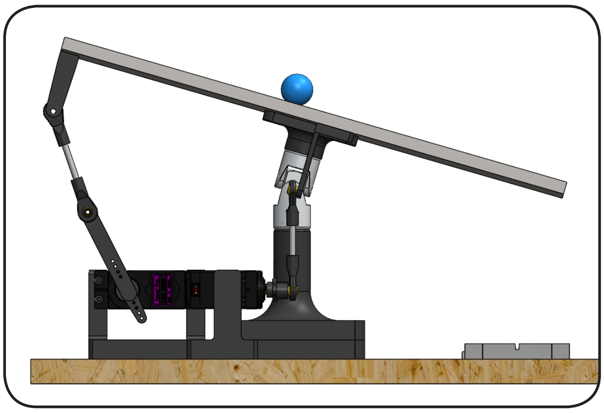
• A webcam continuously tracks the ball's position in real-time.
• The computer processes the camera data to detect the ball's position and velocity, then communicates this information to the microcontroller via the UART protocol.
• Additionally, the computer sends the desired ball position and receives system data, including the current plate tilt angle.
• An IMU sensor provides real-time tilt angle feedback to the microcontroller, closing the control loop.
• The microcontroller executes a control algorithm that processes position and velocity data, computes the required plate tilt angle, and ensures smooth servo motor operation.

Illustration









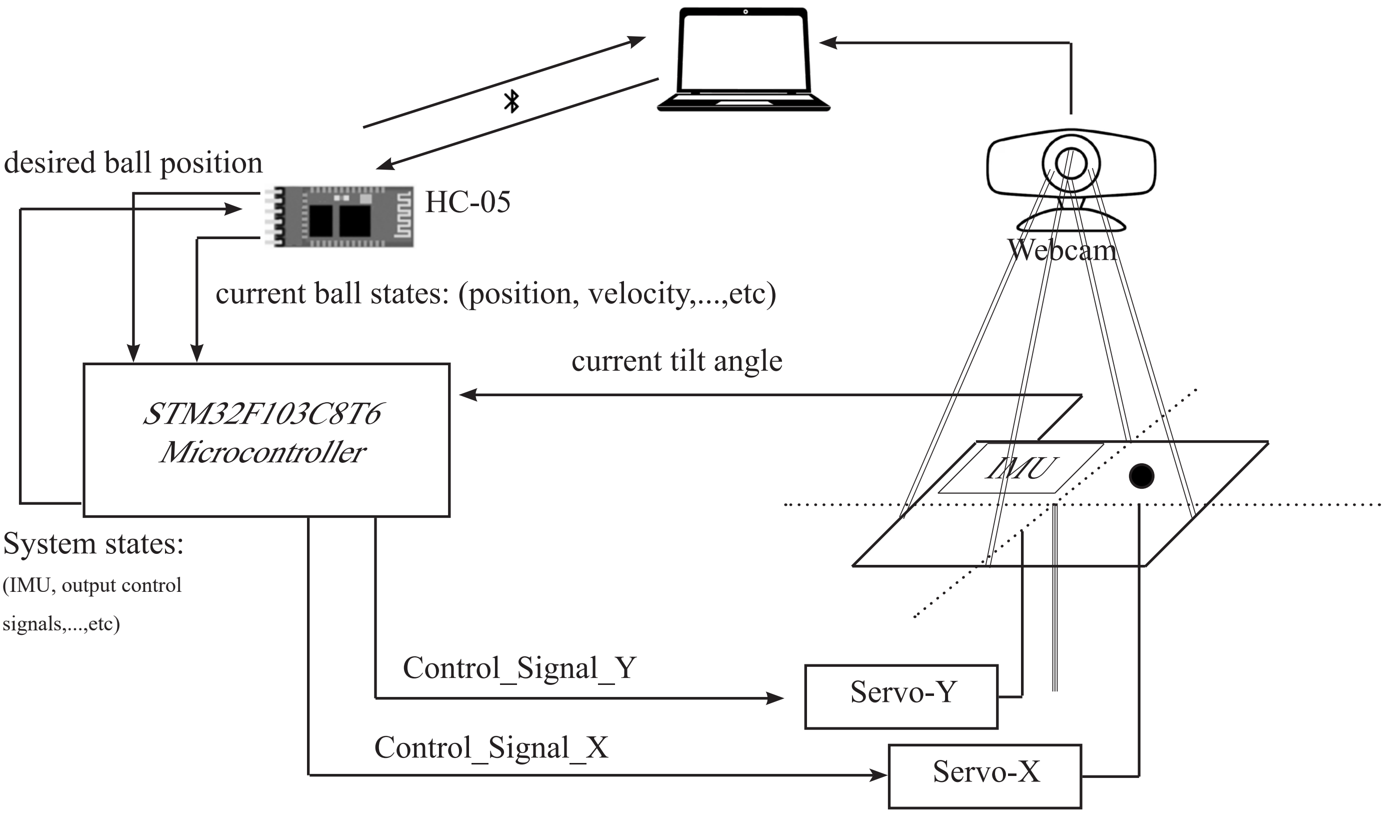
Universal joint Connector Rod, Link, Servo arm

List of proposed components:

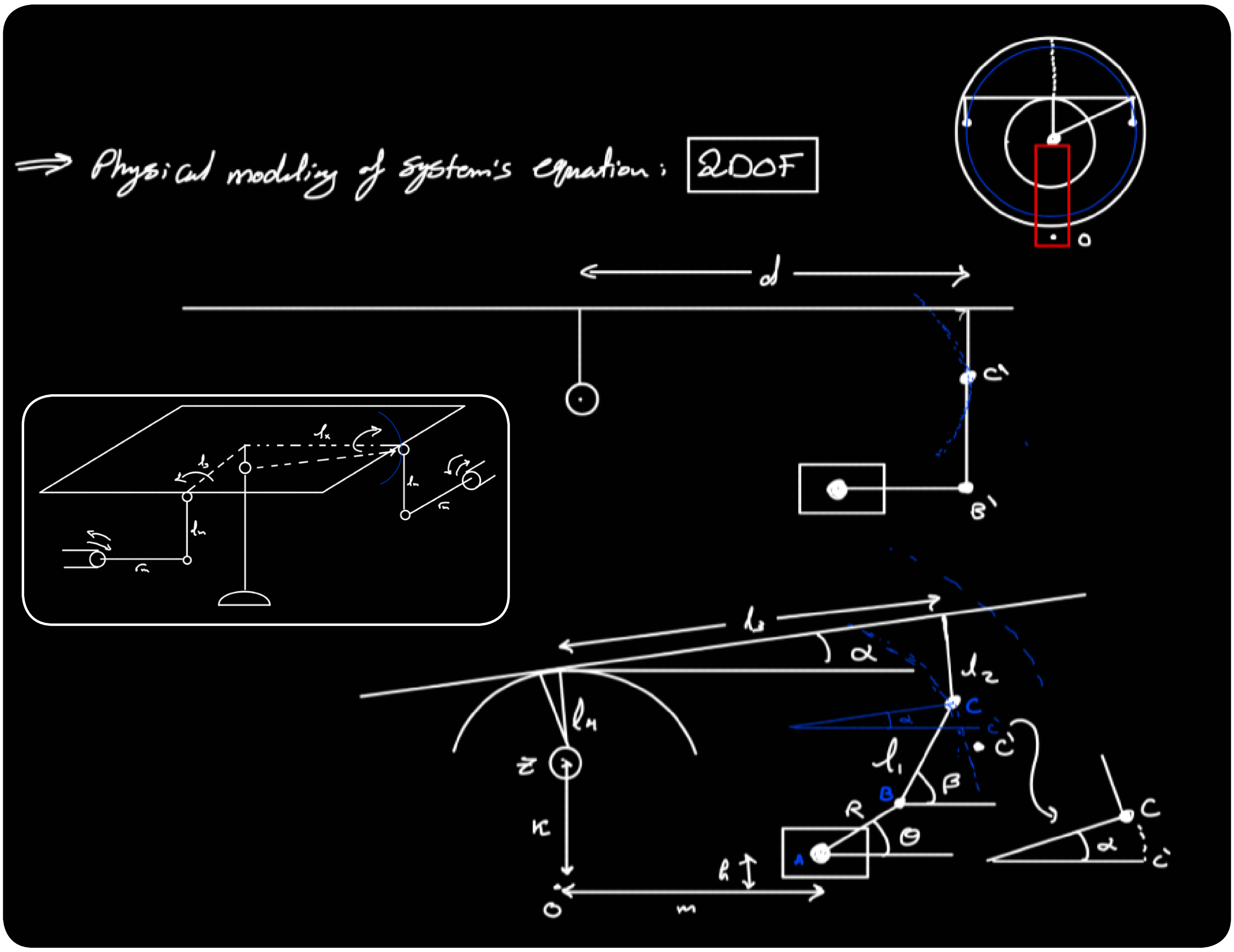
- Microcontroller: STM32F103C8T6.
- 2xMG996R Servo Motor Tower Pro 180 Degree 13kg.cm.
- Webcam.
- MPU-6050 IMU Sensor Module with 6 DOF (Gyroscope, Accelerometer).
- Bluetooth Module HC-05, Serial TTL
- Adjustable Max 5A Step Down Module XL4015 Converter DC-DC 0.8-30V To 5-32V
- 220V to 12V Electric Transformer.
- Full Bridge Rectifier.
- Universal Joint 3/8.
- 2xConnecting Rod with ball joints.

Note: Most of the model parts are 3D printed with specific dimensions according to our calculations.

Functional Diagram

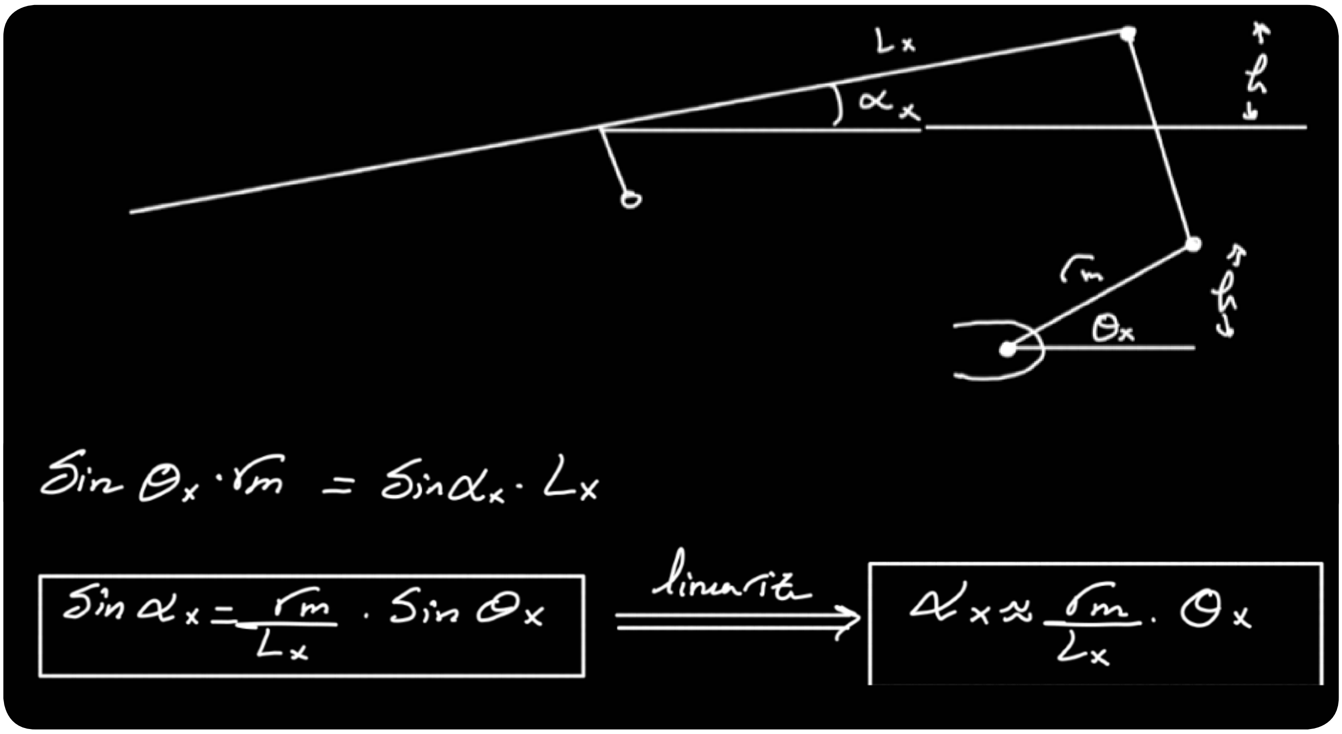


Mathematical system modelling



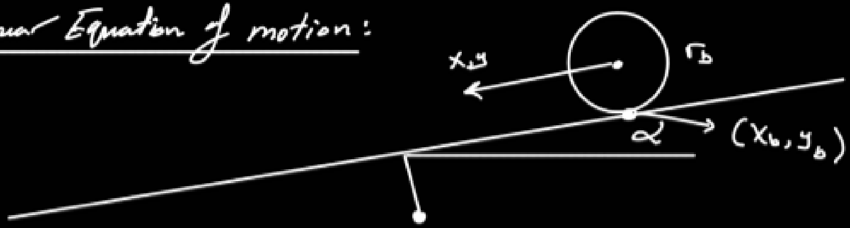
Design Criteria:

Obtain close to a linear relationship between servo angle and plate tilt angle.



Equations of motion

Non-linear Equation of motion:



$$\Rightarrow \text{Lagrange method } (L = K.E - P.E) \quad \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial \dot{x}}\right) - \frac{\partial L}{\partial x} = 0$$

$$K.E = K.E_{trans.} + K.E_{rotation} = \frac{1}{2} m_b v_b^2 + \frac{1}{2} J_b \cdot \omega_b^2 \qquad \omega_b = \frac{V_b}{r}$$

$$V_b = \sqrt{V_x^2 + V_y^2}$$

$$K.E_{Tot} = \frac{1}{2} m_b \cdot (\dot{x}_b^2 + \dot{y}_b^2) + \frac{1}{2} J_b \cdot \frac{(\dot{x}_b^2 + \dot{y}_b^2)}{r_b^2}$$

$$K.E_{Tot} = \frac{1}{2}(\dot{x}_b^2 + \dot{y}_b^2)\left(m_b + \frac{J_b}{r_b^2}\right)$$

$$Pot. = m_b g h_b = - m_b g x_b \cdot \sin\alpha_x - m_b g y_b \cdot \sin\alpha_y$$

$$L = K.E - Pot = \frac{1}{2}(\dot{x}_b^2 + \dot{y}_b^2)\left(m_b + \frac{J_b}{r_b^2}\right) + m_b g x_b \cdot \sin\alpha_x + m_b g y_b \cdot \sin\alpha_y$$

$$\frac{\partial}{\partial t} \cdot \left(\frac{\partial L}{\partial \dot{x}}\right) = \frac{\partial}{\partial t}\left(\frac{\partial L}{\partial x}\left(\frac{1}{2} \cdot \dot{x}_b^2 \left(m_b + \frac{J_b}{r_b^2}\right)\right)\right)$$

$$= \frac{\partial}{\partial t}\left(\dot{x}_b\left(m_b + \frac{J_b}{r_b^2}\right)\right) = \ddot{x}_b \left(m_b + \frac{J_b}{r_b^2}\right)$$

$$\frac{\partial L}{\partial x} = \frac{\partial}{\partial x}(m_b g x_b \sin\alpha_x) = m_b g \sin\alpha_x$$

$$\ddot{x}_b\left(m_b + \frac{J_b}{r_b^2}\right) - m_b g \sin\alpha_x = 0$$

$$\ddot{x}_b = \frac{m_b \cdot g \cdot r_b^2}{m_b \cdot r_b^2 + J_b} \cdot \sin(\alpha_x)$$

$$\ddot{y}_b = \frac{m_b \cdot g \cdot r_b^2}{m_b \cdot r_b^2 + J_b} \cdot \sin(\alpha_y)$$

⇒ Newton's laws of motion:

$$\Sigma F = F_{x,t} - F_{x,r} \rightarrow \text{no slip}$$

$$F_{x,t} = mg\sin\alpha_x \quad , \quad \text{Torque} = F_{x,r}\cdot r_b = J_b\cdot\alpha = \frac{J_b\cdot\ddot{x}_b}{r_b} \qquad \alpha \rightarrow \frac{a}{r}$$

$$F_{x,r} = \frac{J_b\cdot\ddot{x}_b}{r_b^2}$$

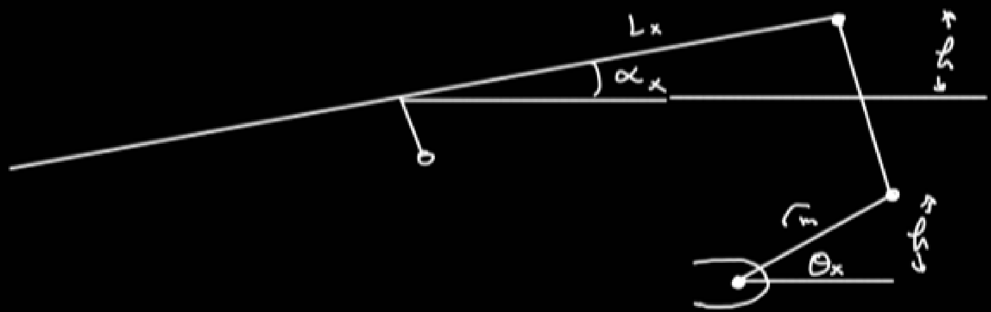
$$F = mg\sin\alpha_x - \frac{J_b\cdot\ddot{x}_b}{r_b^2} = m_b\cdot\ddot{x}_b$$

$$m_b\cdot\ddot{x}_b + \frac{J_b\cdot\ddot{x}_b}{r_b^2} = m_b\cdot g\sin\alpha_x$$

$$\ddot{x}_b\left(\frac{m_b}{1}\times + \frac{J_b}{r_b^2}\right) = m_b\cdot g\sin\alpha_x \quad \Rightarrow \quad \boxed{\ddot{x}_b = \frac{m_b\cdot g\cdot r_b^2\cdot\sin\alpha_x}{m_b\cdot r_b^2 + J_b}} \quad (1)$$

$$\hookrightarrow \frac{m_b\cdot r_b^2 + J_b}{r_b^2}$$

$$\boxed{\ddot{y}_b = \frac{m_b\cdot g\cdot r_b^2\cdot\sin\alpha_y}{m_b\cdot r_b^2 + J_b}}$$



$$\sin\theta_x\cdot r_m = \sin\alpha_x\cdot L_x$$

$$\boxed{\sin\alpha_x = \frac{r_m}{L_x}\cdot\sin\theta_x} \xrightarrow{\text{linearize}} \boxed{\alpha_x \approx \frac{r_m}{L_x}\cdot\theta_x} \Rightarrow (2)$$

2 Substitute in 1

$\theta \rightarrow$ rad

$$\boxed{J_b = \frac{2}{5}m_b\cdot r_b^2}$$

$$\boxed{\ddot{x}_b = \underbrace{\frac{m_b\cdot g\cdot r_b^2\cdot r_m}{(m_b\cdot r_b^2 + J_b)\cdot L_x}}_{\text{constant } K_B}\cdot\theta_x}$$

$$\boxed{\ddot{y}_b = \frac{m_b\cdot g\cdot r_b^2\cdot r_m\cdot\theta_y}{(m_b\cdot r_b^2 + J_b)\cdot L_y}}$$

$$\theta = \begin{cases} \pm 5.7^\circ \Rightarrow \text{error} = 0.017\% \\ \pm 11.5^\circ \Rightarrow \text{error} = 0.13\% \\ \pm 17.5^\circ \Rightarrow \text{error} = 0.6\% \end{cases}$$

Simplifications and assumptions

To derive the previously mentioned equations of motion for a ball-on-platform system, the following assumptions are made:
- The ball rolls without slipping on the platform.
- Friction is neglected.
- The ball is perfectly spherical and homogeneous.
- There is no vertical displacement of the ball relative to the platform.

Observations

Based on the derived equations of motion and the system's dynamic model, the system can be linearized due to the small plate tilt angle range (±15°), resulting in a minimal relative error of approximately 0.6%. The linearized equations remain valid for small deviations. Notably, the theoretical model is independent of both the ball's mass and radius.

Control Algorithm

These requirements are based on previous attempts to construct a similar stable system.
Requirements:
- Allow for the platform to incline (±15°) in every direction
- Settling time (5%) < 1s (Ts < 1)
- Overshoot < 5% (M<0.05)
- Static error for a step input < 5mm ($e_0 < 0.005$)

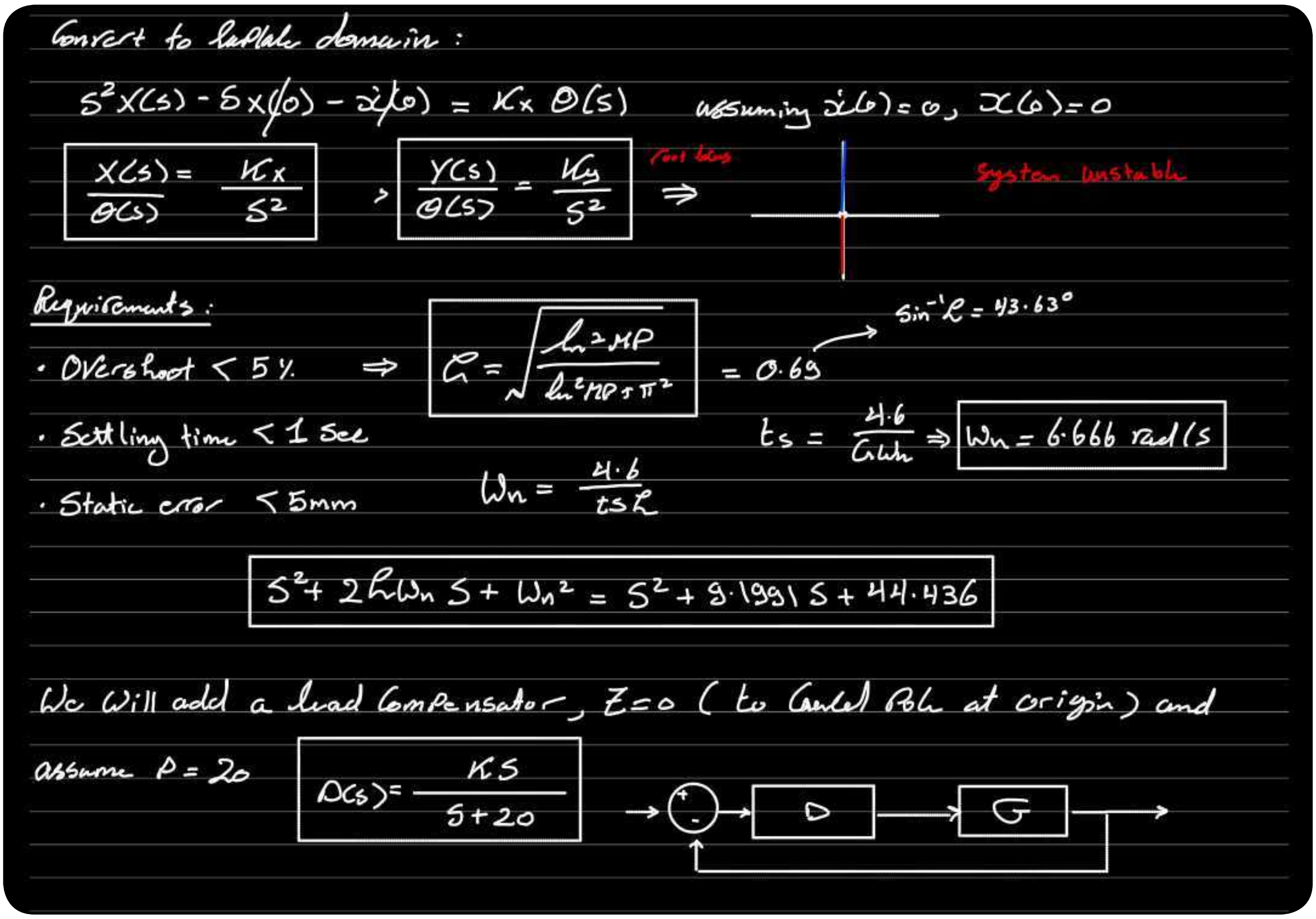


Plate system dynamic equation without considering transfer function of the servo.

Closed loop $P = \dfrac{DG}{1+DG}$

$$1 + DG = 1 + K \frac{s}{s+p} \cdot \frac{K_x}{s^2} = 0 \qquad \times s^2(s+p)$$

$$\boxed{s^3 + p s^2 + K \cdot K_x \cdot s = 0}$$

$$(s+a)(s^2 + 9.1991 s + 44.436) = s^3 + p s^2 + K_x \cdot K \cdot s$$

$$s^3 + (9.1991 + a) s^2 + (44.436 + 9.1991 a) s + 44.436 a = s^3 + p s^2 + K_x K s$$

$a = 0$

$$s^3 + 9.1991 s^2 + 44.436 s = s^3 p s^2 + K_x K \cdot s$$

$$\boxed{P = 9.1991} \qquad \boxed{K = \frac{44.436}{K_x}}$$

accurately $\boxed{P = \dfrac{9.2}{t_s}}$ $\quad K = \left(\dfrac{4.6}{t_s \cdot \zeta}\right)^2 \quad \zeta = \sqrt{\dfrac{\ln^2 M_P}{\ln^2 M_P + \pi^2}}$

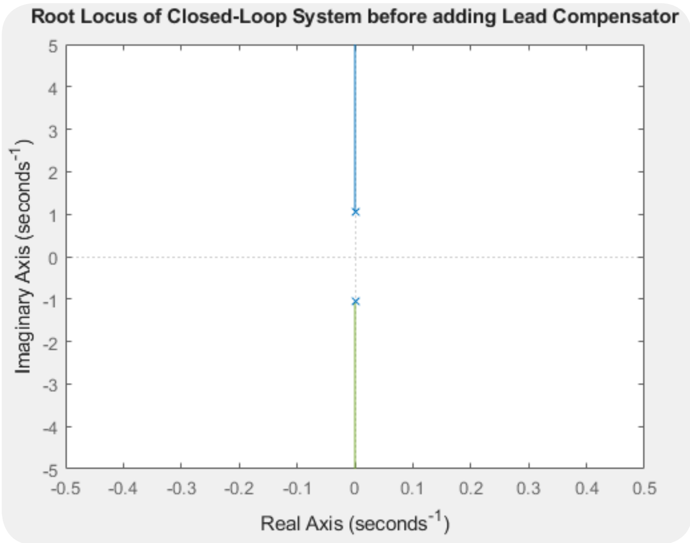
Analyzing the open-loop transfer function of the plant, excluding the servos' transfer function, reveals that the system is inherently unstable due to its poles being located at the origin. As shown in the previous equation and the MATLAB root locus simulation, the system is classified as type 2 ($1/s^2$). To achieve stability, a pole at the origin must be canceled, and the root locus must be shifted to the left-half plane.

A PID controller can stabilize a double integrator system ($1/s^2$) by appropriately tuning the proportional (Kp), integral (Ki), and derivative (Kd) gains. The derivative term (Kd) plays a crucial role in introducing damping, which helps shift the poles into the left-half plane (LHP), ensuring asymptotic stability. However, achieving stability depends on careful tuning of the PID gains, as improper tuning may result in instability or degraded performance.

In theory, due to the system having a embedded integrator, the static error is self-regulated by the system itself. Thus, a integrating part could be redundant in the PID controller.

PID Effects:

Proportional (P) gain affects the system gain but does not change pole locations significantly.
Integral (I) gain introduces an additional pole at the origin, making the system unstable with three integrators.
Derivative (D) gain can increase damping and improve transient response, but alone it cannot place the poles in the left-half plane.



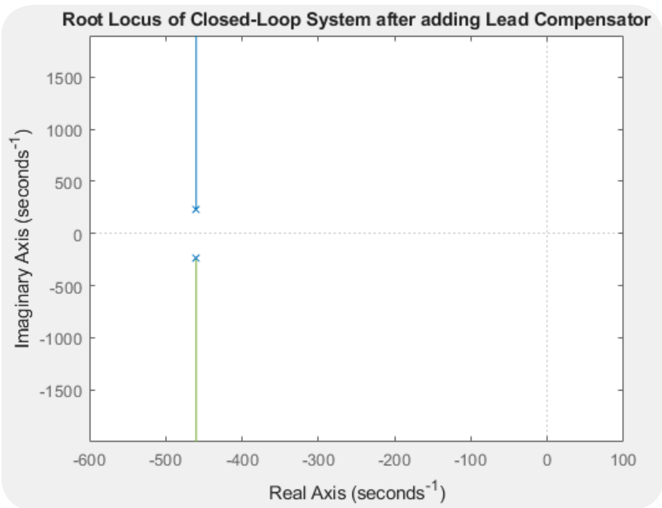
How to Stabilize the System?

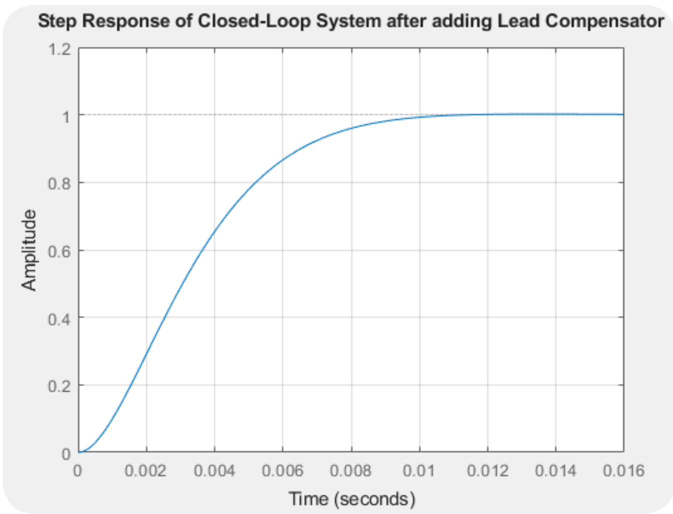
To make the system asymptotically stable, we need to introduce a lead compensator or a PD controller to shift the poles leftward.

• Lead Compensator: K * (s + alpha) / (s + beta), where alpha > beta

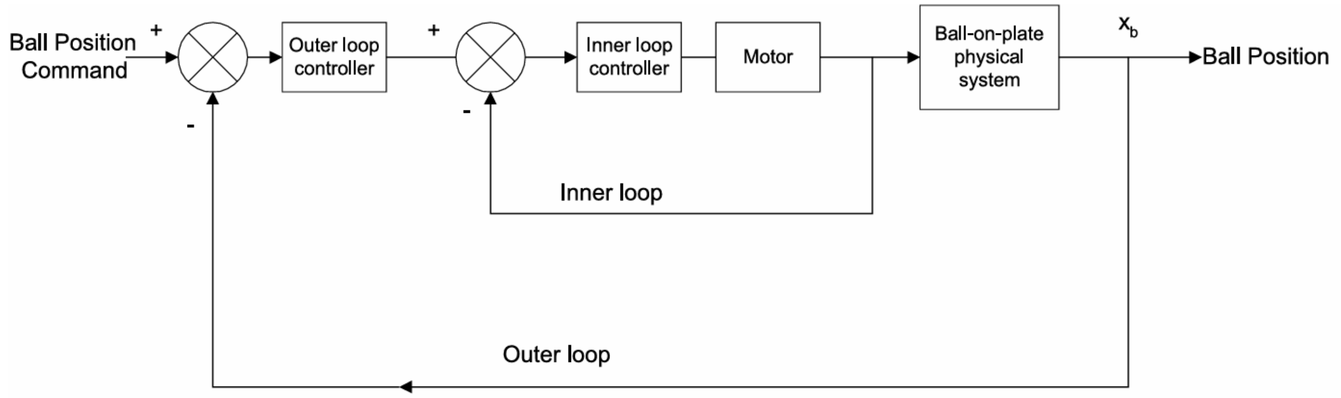
• Introduces a zero in the left-half plane, improving phase margin.
• Effectively shifts the dominant poles leftward for stability.

• The derivative term introduces a zero, improving transient response.
• This can cancel one integrator, turning the system into a stable first-order system.





Controller Design



The control system consists of an inner loop and an outer loop. The inner loop is a unity feedback closed-loop system responsible for controlling the position of the servomotor. The servomotor, which is part of the ball balancing table, automatically adjusts its position using its built-in controller. Due to the fast response of the inner loop, it reaches steady-state almost instantaneously. As a result, the inner loop dynamics can be approximated as an ideal unit first-order system with a time delay. This simplification allows the inner loop to be reduced, resulting in a streamlined block diagram for the overall system.

The control loop operates as follows:

1-The user defines a setpoint, which represents the desired ball position.

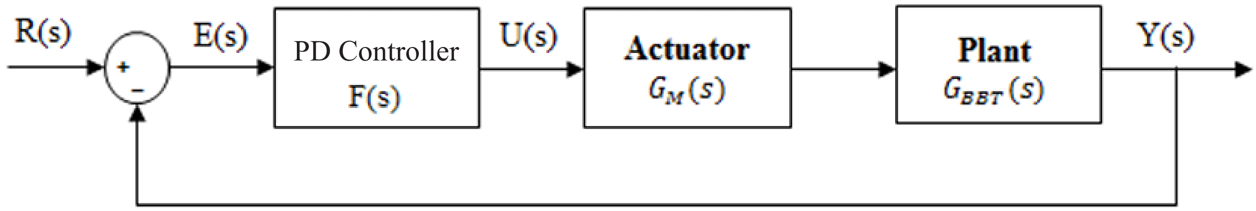
2-The error is calculated as the difference between the setpoint and the actual ball position.

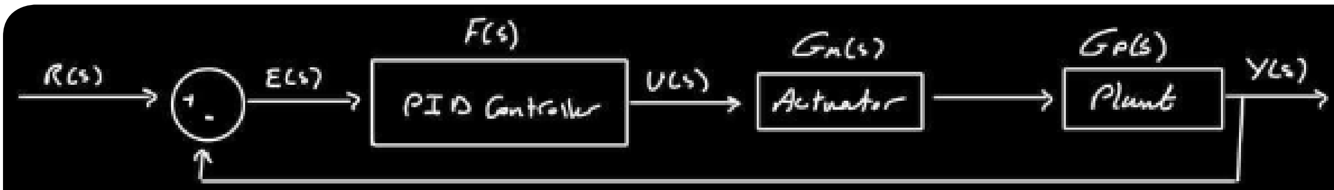
3-Based on this error, the microcontroller computes the desired servo angle required to achieve the setpoint.

4-The servomotor rotates to the desired angle, which geometrically determines the inclination of the platform. This inclination serves as the input to the system dynamics, influencing the ball's motion.

5-The actual ball position is measured by the webcam and fed back into the system, closing the loop.

Since the servomotors are equipped with built-in control circuits, there is no need to design separate servo controllers. Instead, the servomotor dynamics can be modeled as an ideal unit with a time delay, representing their ability to accurately and promptly execute the desired angular rotation.





$G_P(s) = \frac{K}{s^2}$ $\quad$ $K \rightsquigarrow$ Plant Constant

Controller $\Rightarrow$ PD Controller

$G_M(s) = \frac{\theta(s)}{V(s)} = \frac{K_m}{\tau s + 1}$ $\quad$ $K_m \rightsquigarrow$ gain, $\tau \rightsquigarrow$ Time Constant

- If a 1ms PWM corresponds to a 45° rotation $\Rightarrow K_m = 45°/ms$
- Time constant determines how quickly the system responds.
  - ↳ Practically (time to reach 63.2% of final position)

$$F(s) \times G_M \times G_P = \frac{K_m \cdot K}{(\tau s+1)(s^2)} \cdot K_c(1 + T_d s)$$

let $K_m \cdot K_c \cdot K = C$

$$T(s) = \frac{Y(s)}{R(s)} = \frac{F(s)\cdot G_M(s)\cdot G_P(s)}{1 + F(s)\cdot G_M(s)\cdot G_P(s)} = \frac{K_m \cdot K \cdot K_c(1 + T_d s)}{K_m \cdot K \cdot K_c(1+T_d s) + s^2(\tau s + 1)}$$

Characteristic equation: $C + T_d \cdot C \cdot s + \tau s^3 + s^2$

ch. eq $\Rightarrow$ $\tau s^3 + s^2 + T_d \cdot C s + C = 0$

$$(as+b)(s^2 + 2\zeta\omega_n s + \omega_n^2) = \tau s^3 + s^2 + T_d \cdot C \cdot s + C$$

$$as^3 + 2a\zeta\omega_n s^2 + a\omega_n^2 \cdot s + bs^2 + 2\zeta\omega_n \cdot b \cdot s + b\omega_n^2$$

$$as^3 + (2a\zeta\omega_n + b)s^2 + (a\omega_n^2 + 2\zeta\omega_n \cdot b)s + b \cdot \omega_n^2$$

$s^3: a = \tau$

$s^2: (2a\zeta\omega_n + b) = 1$

$s: a\omega_n^2 + 2\zeta\omega_n \cdot b = T_d \cdot C$

$s^0: b\omega_n^2 = C$

$b = \frac{C}{\omega_n^2}$

$2\tau\zeta \cdot \omega_n + b = 1$

$b = 1 - 2\tau\zeta \cdot \omega_n = \frac{C}{\omega_n^2}$

$C = \omega_n^2 - 2\tau \cdot \zeta \cdot \omega_n^3$

$$\boxed{T_d = \frac{\tau \cdot \omega_n^2 + 2\zeta\omega_n \cdot (1 - 2\tau\zeta \cdot \omega_n)}{\omega_n^2 - 2\tau \cdot \zeta \cdot \omega_n^3}}$$

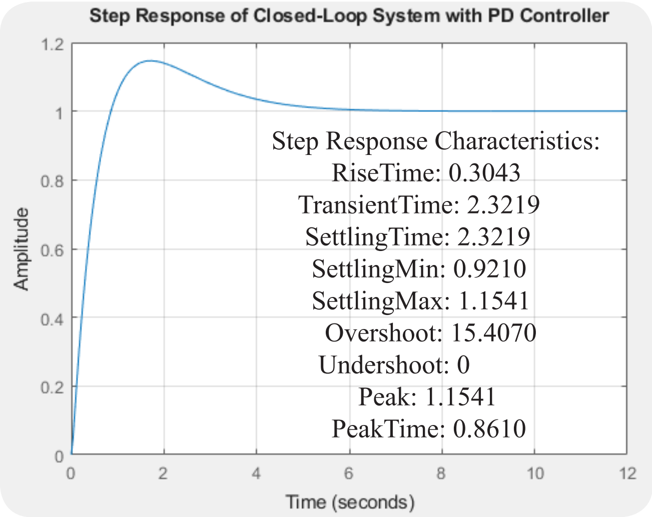
$$\boxed{K_c = \frac{\omega_n^2 - 2\tau \cdot \zeta \cdot \omega_n^3}{K_m \cdot K}}$$

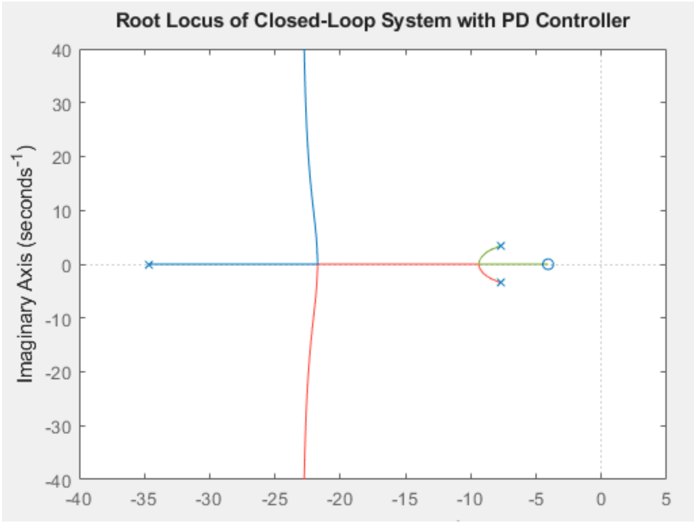
MP $\quad$ ts

$\downarrow$ inputs $\downarrow$

$$\boxed{\zeta = \sqrt{\frac{\ln^2 MP}{\ln^2 MP + \pi^2}} \qquad \omega_n = \frac{4.6}{t_s \cdot \zeta}}$$

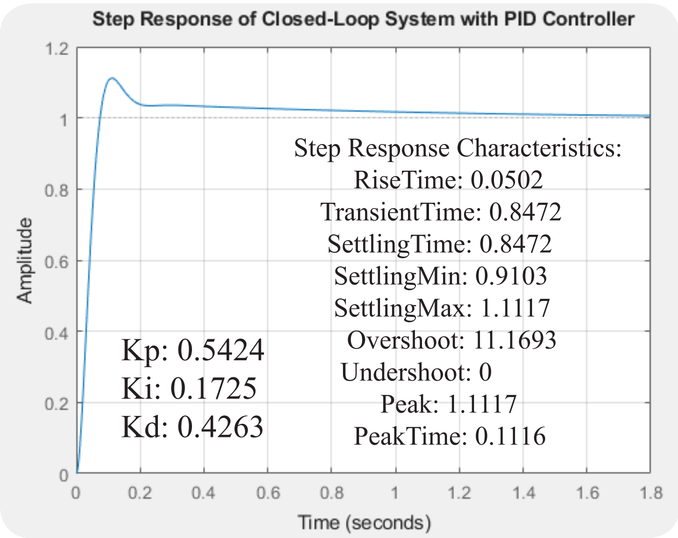
Solving the system in MATLAB and choosing an appropriate and realistic settling time and overshoot we end up with:

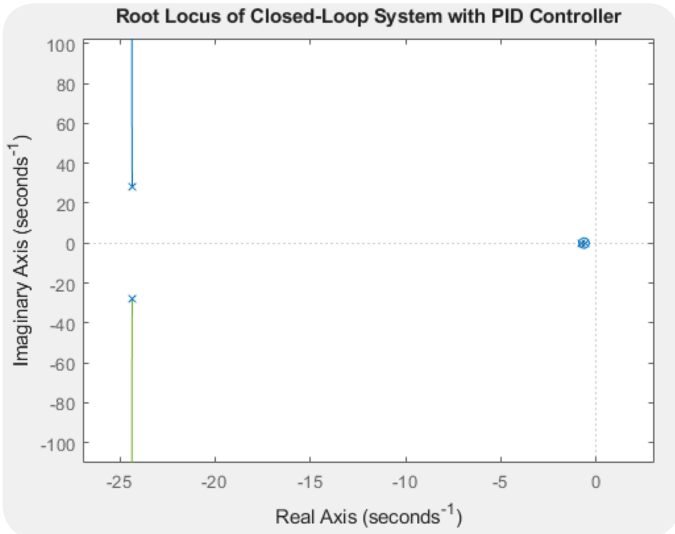




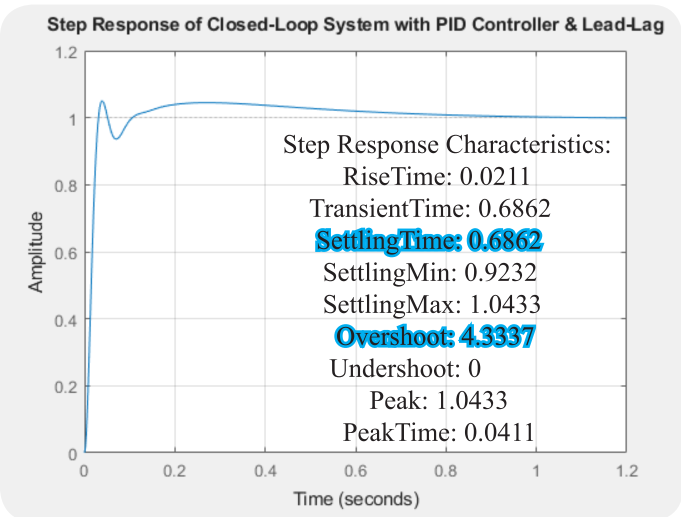
Observation:

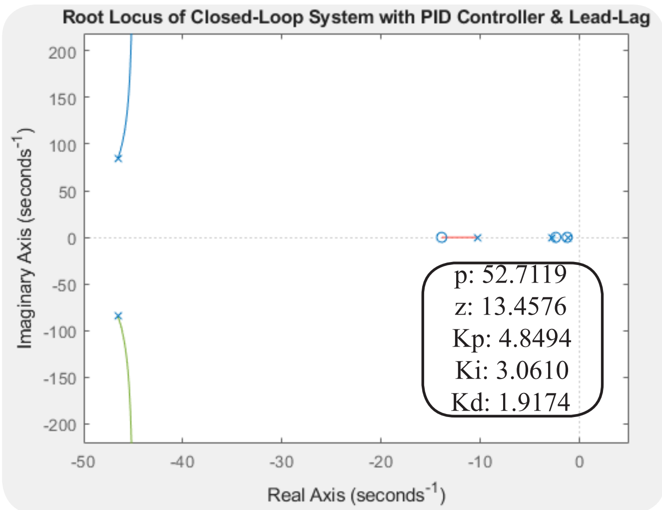
The Overshoot is in a very high range (15%), Closed loop system does not satisfy the expected overshoot. PD controller design is not preferred for ball balancing table because of the system's integrator terms. Replacing it with PID gives slightly better results.





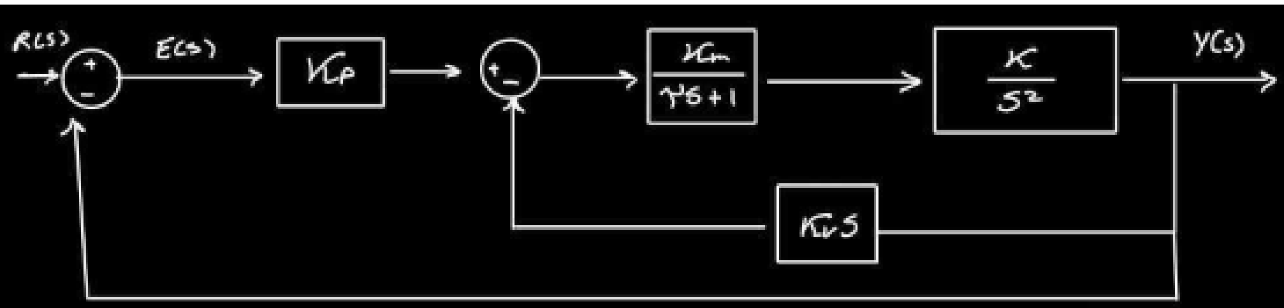
We suggest adding another Lead-Lag compensator, simulation gives far better results





Final Controller Design

To improve system performance and achieve zero overshoot with minimal settling time, we will redesign the system by introducing an inner-loop feedback around the motor and plant transfer function. This inner loop will include a velocity feedback term KvS to introduce a zero in the root locus and affect the pole locations, improving system stability. Once the inner loop is established, we will introduce a proportional gain in the forward path, which will complete the outer loop and allow fine-tuning of the system response.



T·F:

$$\frac{\frac{x}{y}}{1+K\frac{x}{y}} = \frac{x}{y+Ksx}$$

$$T\cdot F = \frac{K_m \cdot K}{s^2(\tau s+1) + K_v \cdot K \cdot K_m \cdot s}$$

$$\text{Inner } T\cdot F = \frac{K_m \cdot K \cdot K_p}{\tau s^3 + s^2 + K_v \cdot K \cdot K_m \cdot s}$$

$$\boxed{\text{Outer } T\cdot F = \frac{K_p \cdot K_m \cdot K}{\tau s^3 + s^2 + K_v \cdot K \cdot K_m s + K_m \cdot K \cdot K_p}}$$

$$(as+b)(s^2+2\zeta\omega_n s+\omega_n^2) = \tau s^3 + s^2 + K_v \cdot K \cdot K_m \cdot s + K_m \cdot K \cdot K_p$$

$$as^3 + (2a\zeta\omega_n + b)s^2 + (a\omega_n^2 + 2\zeta\omega_n \cdot b)s + b\cdot\omega_n^2 = \text{''}$$

$s^3: a = \tau$

$s^2: 2a\zeta\omega_n + b = 1$ $\qquad b = 1 - 2\tau\cdot\zeta\omega_n$

$s^1: a\omega_n^2 + 2\zeta\omega_n \cdot b = K_v \cdot K \cdot K_m$

$s^0: b\cdot\omega_n^2 = K_m \cdot K \cdot K_p$

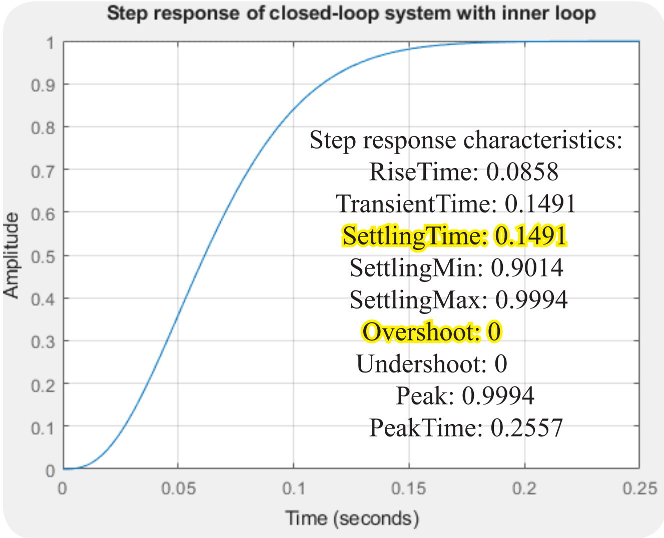
$$\boxed{K_p = \frac{\omega_n^2}{K_m \cdot K \cdot (1 - 2\tau\cdot\zeta\omega_n)}}$$

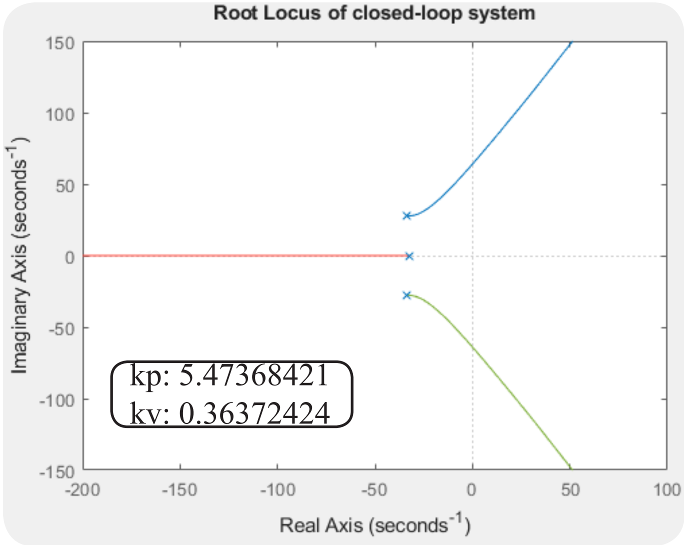
MP $\qquad$ ts

inputs

$$\boxed{\zeta = \sqrt{\frac{\ln^2 MP}{\ln^2 MP + \pi^2}} \qquad \omega_n = \frac{4.6}{t_s \cdot \zeta}}$$

$$\boxed{K_v = \frac{\tau\omega_n^2 + 2\zeta\omega_n \cdot (1 - 2\tau\cdot\zeta\omega_n)}{K \cdot K_m}}$$



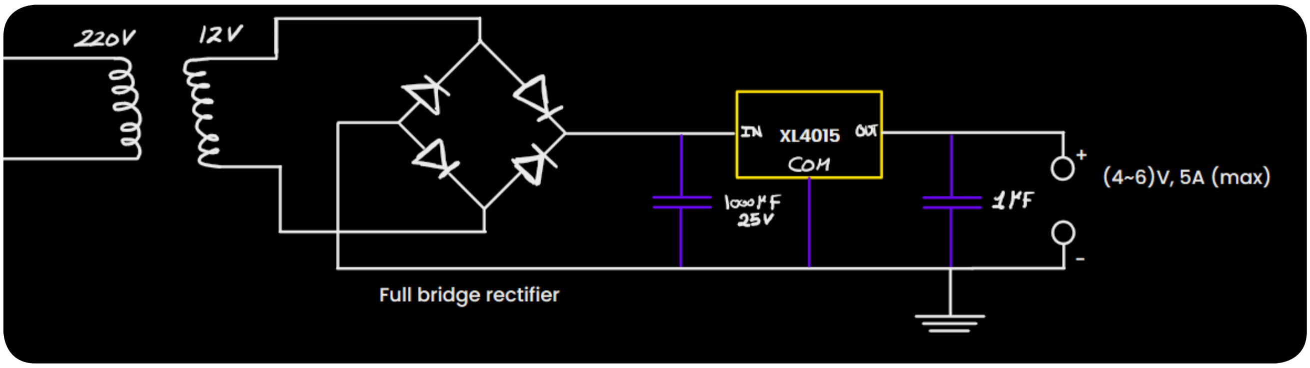


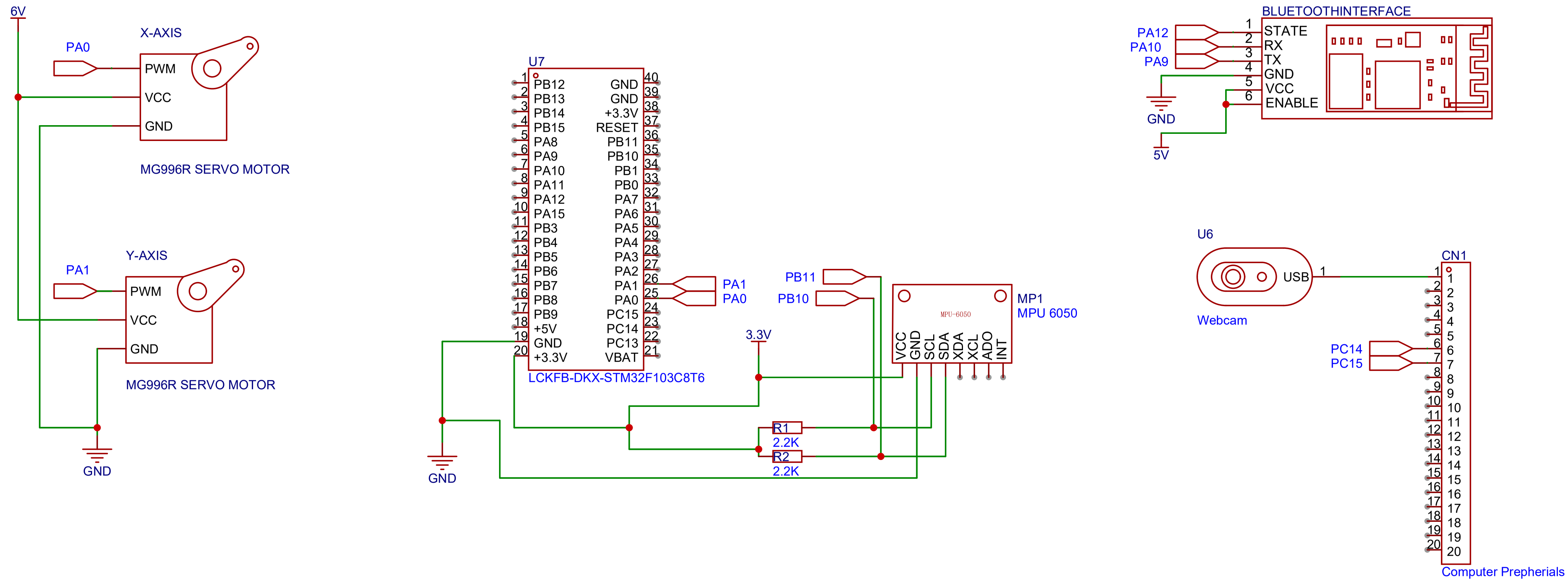
This design was achieved through multiple design iterations utilizing different control techniques. The previous design was based on the Root Locus method, which was employed to fine-tune the controller parameters and analyze system stability.

MATLAB Code link: https://drive.google.com/drive/folders/1TmTcjJNSpGJq7NGU4tqUV828N_sw-WkK?usp=sharing

Electrical circuit

Power supply circuit for the servo:





Schematic	Schematic1	Create at	2025-03-13
		Update at	2025-03-13
Board	Board1	Page	P1
Drawn		Ball Balance System	
Reviewed			
		Version	Size	Page 1 Total 1
EasyEDA		V1.0	A4	EasyEDA.com