Habib University

EE 361L-T1

Prin. of Feedback Ctrl (Lab)-T1

PFC Project Report

Project: Inertial Platform Stabilizer
Team: UnStabilizers

Author(s):
Muslim
Vezhish
Abdullah

January 11, 2024

Contents

1 Introduction

This project introduces an innovative approach in the field of motion stabilization. We develop a single-axis gimbal system integrated with an Inertial Measurement Unit (IMU) sensor and a carefully calibrated Proportional-Integral-Derivative (PID) controller. The actuator and stable platform exhibit a symbiotic relationship, as they establish a mutually advantageous link that ensures that stability is maintained despite the occurrence of external motions or disturbances. The IMU sensor is of utmost importance in the real-time monitoring of orientation, effectively capturing minute angular variations with precision. The PID controller, following the principles of control theory, effectively organizes precise adjustments to counteract external disturbances, ensuring the platform's resilience against perturbations.

2 Methods & Materials

2.1 Mathematical Modelling Of Plant, Actuators, and Sensors

2.1.1 Controller Model

The Proportional-Integral-Derivative (PID) controller is a fundamental component employed inside control systems in engineering and industrial contexts. The primary objective of this system is to efficiently govern and maintain optimal setpoints by employing three essential control actions, specifically Proportional (P), Integral (I), and Derivative (D). The proportional term demonstrates a linear relationship with the current error, so enabling timely adjustments to fluctuations in the system. The integral component within a control system functions by accumulating and amplifying past error values, hence leading to the mitigation of steady-state errors. The derivative component of a control system places emphasis on the temporal rate of change of the error signal, hence reducing overshooting and oscillatory behavior, while simultaneously introducing dampening effects. The incorporation of these three components enables the attainment of an ideal balance that encompasses timely system reactions, minimal excessive deviations, and reduced errors in a stable condition.

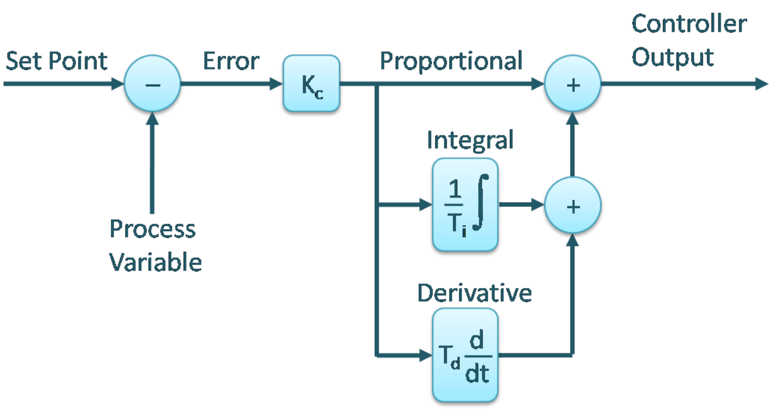


Figure 1: PID Controller

The transfer function for the controller can be written as;

$$C(s) = K_P + sK_D + \frac{K_I}{s}$$

Where;

$$K_P = \text{Proportional Gain.}$$
$$K_I = \text{Integral Gain.}$$
$$K_D = \text{Derivative Gain.}$$

2.1.2 Actuator Model

Actuator Transfer Function :

$$T(s) = \frac{K_t(N_2/N_1)}{s\{(Js + D)(sLa + Ra) + K_t^2\}}$$

Where:

$$\begin{aligned} T(s) &\text{ is the transfer function in Laplace domain.} \\ K_t &\text{ is Motor Torque Constant.} \\ J &\text{ is the moment of inertia of the system.} \\ s &\text{ is the Laplace variable.} \\ D &\text{ is the damping or friction in the system.} \\ L_a &\text{ is the inductance of Motor.} \\ R_a &\text{ is the resistance of Motor.} \\ N_2/N_1 &\text{ is the gear ratio of our Motor.} \end{aligned}$$

We used the First Principle Method and Grey Box Modelling for the modelling of the DC Motor. The DC Gearmotor has limited documentation or detailed technical specifications available, which is common for many commercial off-the-shelf components. This limited information can make it challenging to create a comprehensive first principle model from scratch. Grey box modelling allowed us to start with the fundamental physics-based equations for a DC motor but also integrate empirical data based off experiments to account for the unknown or unmeasured parameters of this specific motor model.

To find

$$(K_t, R_a, D, L_a, J)$$

the ‘First Principle Method’ is utilized and to find motor parameters.

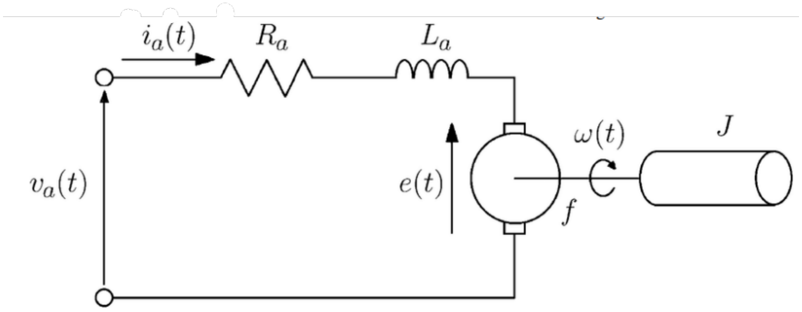


Figure 2: Armature Circuit

1. $R_a = 21\ \Omega$ (Measured through DMM)
2. $L_a = 10\ mH$ (Measured through LCR)
3. $J = \frac{1}{2}mr^2 = \frac{1}{2}(0.935)(0.002)^2 = 1.87\ \mu\ Kg.m^2$
4. $N_2/N_1 = 1/14$
5. $K_t = \frac{\tau_{stall} R_a}{E_a} = \frac{1.57*21}{12} = 2.7475\ \frac{Kg.m}{A}$
6. $D = \frac{E_a I_0}{\omega_{no\ load}} = \frac{12*0.053}{37.175} = 0.017\ N-m-s/rad$

Thus;

Actuator Transfer Function :

$$T(s) = \frac{2.7475(1/14)}{s\{(s1.87\mu + 0.017)(s10m + 21) + 7.55\}}$$
$$= \frac{0.196}{s\{s^2(0.0187\mu) + s(20.927m) + 7.907\}}$$

2.1.3 Model of Plant

For the equivalent model of the plant we will use a $J_{equivalent}$. The J_{plant} will be the moment of inertia of the 2D square acrylic sheet we actuate towards stability.

To find the moment of inertia (J) of an acrylic sheet, we need to know its thickness (t) and the orientation of the axis with respect to the sheet. The moment of inertia is a measure of an object's resistance to changes in its rotational motion.

The moment of inertia of a thin, flat sheet about an axis perpendicular to its plane is given by the formula:

$$J = \frac{1}{12} \cdot m \cdot (a^2 + b^2)$$

Where:

$$J = \text{Moment Of Inertia}$$
$$m = \text{Mass of the sheet}$$
$$a = \text{Length of one side of the sheet}$$
$$b = \text{Length of the other side of the sheet}$$

We use the size of the acrylic sheet as a by b, which means both a and b are known. However, we also need the thickness of the sheet (t) and the density of acrylic to calculate the mass (m).

The mass (m) can be calculated using the formula:

$$m = \text{volume} \cdot \text{density}$$

Assuming the acrylic sheet is of uniform thickness, the volume is the product of its area and thickness:

$$\text{volume} = a \cdot b \cdot t$$

A more convenient way is to use a spring balance to find M, combined mass of the stabilizing platform and the mass that we intend to keep over it. This can be two phones, which when kept on top of an acrylic sheet bring the mass M to an approximate value of 0.5kg.

$$J = \frac{1}{12} \cdot 0.5\,\text{kg} \cdot (0.15\,\text{m})^2 + (0.15\,\text{m})^2$$
$$J = 1.875 \times 10^{-3}\,\text{kg} \cdot \text{m}^2$$

This moment of inertia will be transferred to the load side of the motor and the transfer function will exactly be the same, just replacing J with J_m:

Plant Transfer Function :

$$T(s) = \frac{K_t(N_2/N_1)}{s\{(J_m s + D)(sLa + Ra) + K_t^2\}}$$

Where;

$$J_m = J_{motor} + (N_1/N_2)^2 J = 0.3675 \text{ kg} \cdot \text{m}^2$$

Thus;

$$T(s) = \frac{0.196}{s\{s^2(3.675m) + s(7.7177) + 7.907\}}$$

2.1.4 Model of Gyroscopic Sensor

Gyroscopic sensors are included into the structure of a singular-axis gimbal to accurately measure the angular velocity or displacement along the specific axis that the gimbal is designed to steady. The aforementioned sensors provide a constant capability to detect and monitor any unintended motion or oscillation that may occur within the platform. Following this, the data is sent to a specialist PID controller for additional processing.
Typically, a gyroscope model is represented as a second order transfer function given as

$$G_g(s) = \frac{\omega_n^2}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

Where;

$$\zeta \text{ is the damping ratio of gyroscope} = 0.7\ N - s/m$$
$$\omega_n \text{ is the natural frequency of gyroscope.} = 50\ Hz$$

Thus the transfer function of the Gyroscope is;

$$G_g(s) = \frac{2500}{s^2 + 70s + 2500}$$

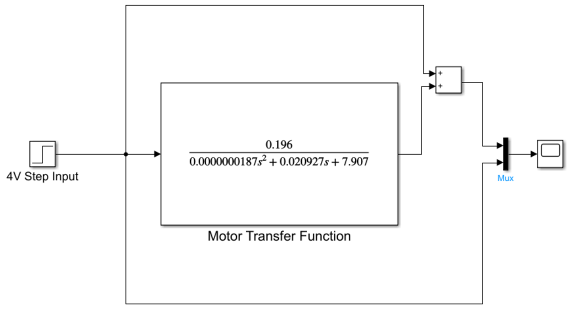
Converting our frequency into radians per second we have;

$$G_g(s) = \frac{98694}{s^2 + 439.82s + 98694}$$

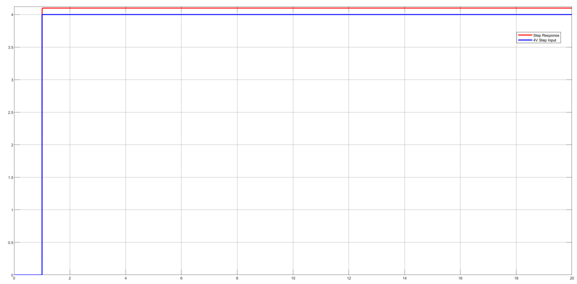
2.2 Model Validation

2.2.1 Motor

2.2.2 Simulink Block Diagram

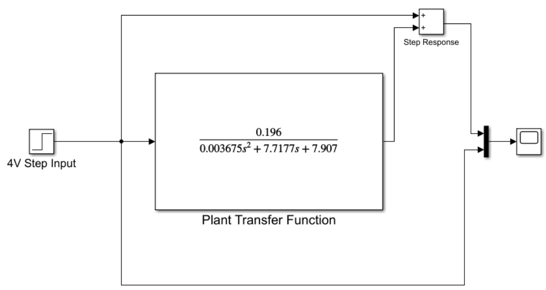


2.2.3 Output Graph

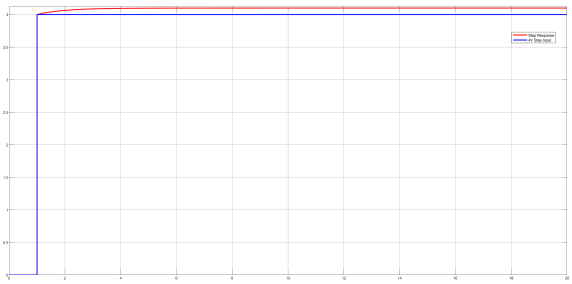


2.2.4 Plant

2.2.5 Simulink Block Diagram

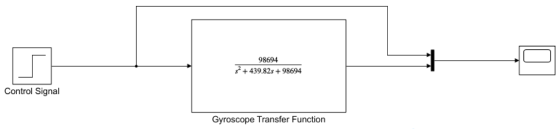


2.2.6 Output Graph

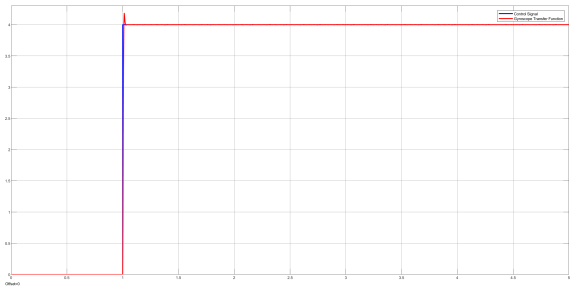


2.2.7 Gyroscope

2.2.8 Simulink Block Diagram



2.2.9 Output Graph



2.3 Gimbal Dynamics: Precision Analysis & Model Validation

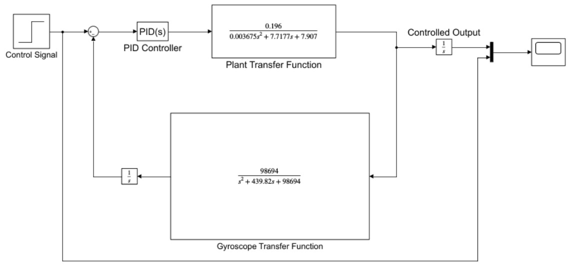


Figure 3: Final Simulink Model

The attached simulink block diagram shows the final block diagram that represents the entire project. It consists of an input signal, a PID controller, the plant, MPU (gyroscope) and an integrator at the output before the scope.

2.3.1 Highly Fine Tuned Model

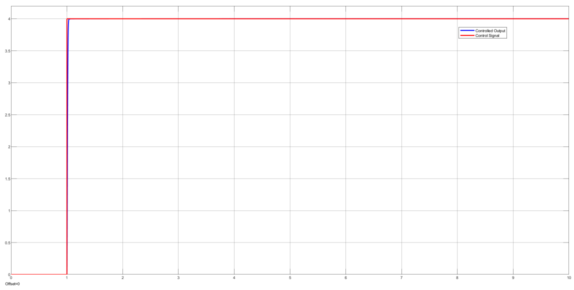


Figure 4: Output Response

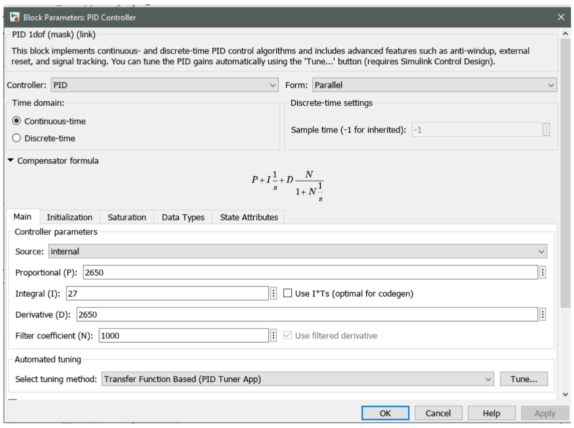


Figure 5: PID Values After Tuning

A PID Controller is used to tune the system to achieve an output response of the system with no error between the output and the input signals. The PID Controller receives the input and feedback and generates a control signal to actuate the gimbal axis via the DC Motor.

To tune the PID controllers for this system, we need to adjust the three parameters: Kp, Ki, and Kd, which are the proportional, integral, and derivative gains, respectively. These parameters affect the response of the system in different ways:

Kp determines how much the system reacts to the error. A higher Kp means a faster response, but also a higher overshoot and oscillation. A lower Kp means a slower response, but also a lower overshoot and oscillation.

Ki determines how much the system accumulates the error over time. A higher Ki means a faster elimination of the steady-state error, but also a higher risk of instability and oscillation. A lower Ki means a slower elimination of the steady-state error, but also a lower risk of instability and oscillation.

Kd determines how much the system reacts to the rate of change of the error. A higher Kd means a faster damping of the oscillations, but also a higher sensitivity to noise and disturbance. A lower Kd means a slower damping of the oscillations, but also a lower sensitivity to noise and disturbance.

We achieve a perfect balance of Kp, Ki and Kd values with trial and error (and a great amount of help from Dr. Shafayat Abrar, without his help this model would be far from perfect).

With Kp = 2650, Ki = 27 and Kd = 2650 we achieve the desired output response that validates our system.

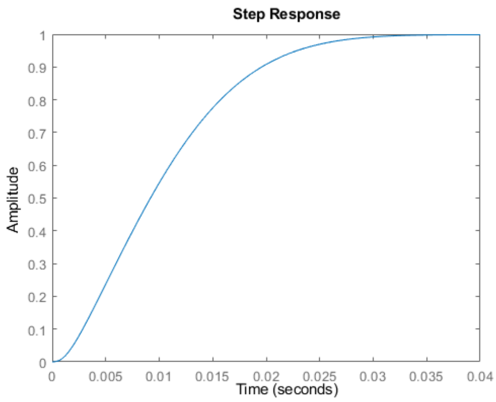


Figure 6: Step Response

```
stepinfo(tf_s)
```

```
ans = struct with fields:
         RiseTime: 0.0167
    TransientTime: 0.0267
     SettlingTime: 0.0267
      SettlingMin: 0.9058
      SettlingMax: 0.9988
        Overshoot: 0
       Undershoot: 0
             Peak: 0.9988
         PeakTime: 0.0486
```

Figure 7: System Response Statistics

The output response follows the input step signal with almost no steady state error. The settling time is almost ideal, at a minimal value of 0.0267s. This validates our mathematical model.

2.3.2 Realistically Tuned Model

In this section we tune our model away from a very ideal response to a response that is more realistic according to IEEE literature. The settling time in this tuning has been increased to 0.2376s, in line with actual gimbal settling times. This also aids the stability of the mass on the platform since inertia would act on it and nothing will strap it to the platform.

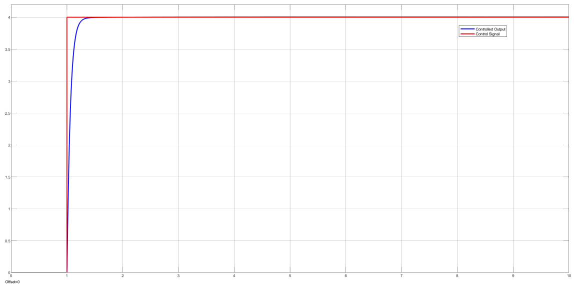


Figure 8: Output Response

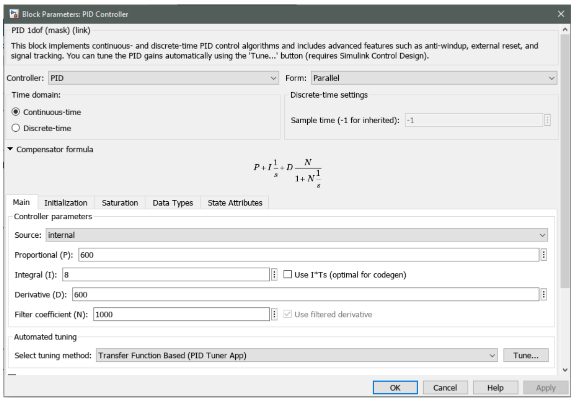


Figure 9: PID Values After Tuning

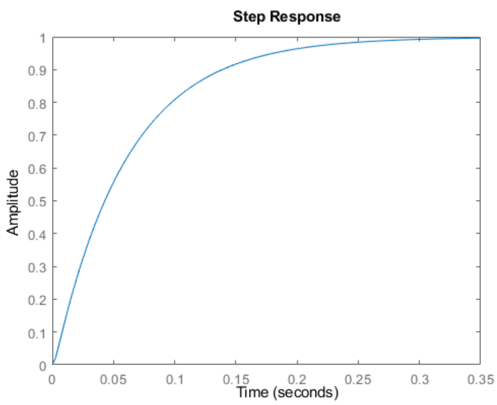


Figure 10: Step Response

```
stepinfo(tf_s)

ans = struct with fields:

         RiseTime: 0.1309
    TransientTime: 0.2376
     SettlingTime: 0.2376
      SettlingMin: 0.9001
      SettlingMax: 0.9983
        Overshoot: 0
       Undershoot: 0
             Peak: 0.9983
         PeakTime: 0.4333
```

Figure 11: System Response Statistics

```
stepinfo(tf_s)

ans = struct with fields:

         RiseTime: 0.1309
    TransientTime: 0.2376
     SettlingTime: 0.2376
      SettlingMin: 0.9001
      SettlingMax: 0.9983
        Overshoot: 0
       Undershoot: 0
             Peak: 0.9983
         PeakTime: 0.4333
```

Figure 12: System Response Statistics

2.4 Physical Design

The first step of the physical design was to create a frame for the single axis. The choice of light wood was made for this frame. A 7x7 frame was made using wood, cut diagonally at the ends for a firmly joined structure. Holes were made to accommodate the bearing that would support the actuating shaft along with the motor's rotating actuator.

Figure 13: System Response Statistics

An ESP 32 was used as a controller, an MPU 6050 was used as the IMU Sensor, a JGA 25371 GearMotor was used as the actuator. In the attached picture, these are laballed. The CU refers to the Control Unit (ESP32). ESP32 generally has more processing power compared to most Arduino boards. Since our gimbal project involves complex calculations, sensor fusion, and real-time processing, the additional processing power of the ESP32 is beneficial. We also tune Kp, Ki, Kd values using a phone, and an ESP 32 has bluetooth capabilities. For an Arduino, this would require an additional bluetooth module which would also require power.

3 Conclusion

In conclusion, this project has undertaken a thorough analysis of the development of an automated single-axis gimbal system. The establishment of transfer functions for key components such as the actuator, plant, and sensors forms the fundamental basis of an effective control system. The integration of a proportional-integral-derivative (PID) controller into the direct current (DC) motor actuator and single-axis gimbal is essential for achieving optimal motion stabilization.
However, it is imperative to acknowledge specific limitations within our endeavor. The model was simplified by making assumptions about certain quantities, such as the moment of inertia and damping components. Nevertheless, it is essential to recognize that variations in these parameters in real-world scenarios can potentially

influence the system's performance.

Despite these limitations, the primary aim of our study has been achieved - to develop a reliable and automated single-axis gimbal system that enhances precision in motion stabilization.

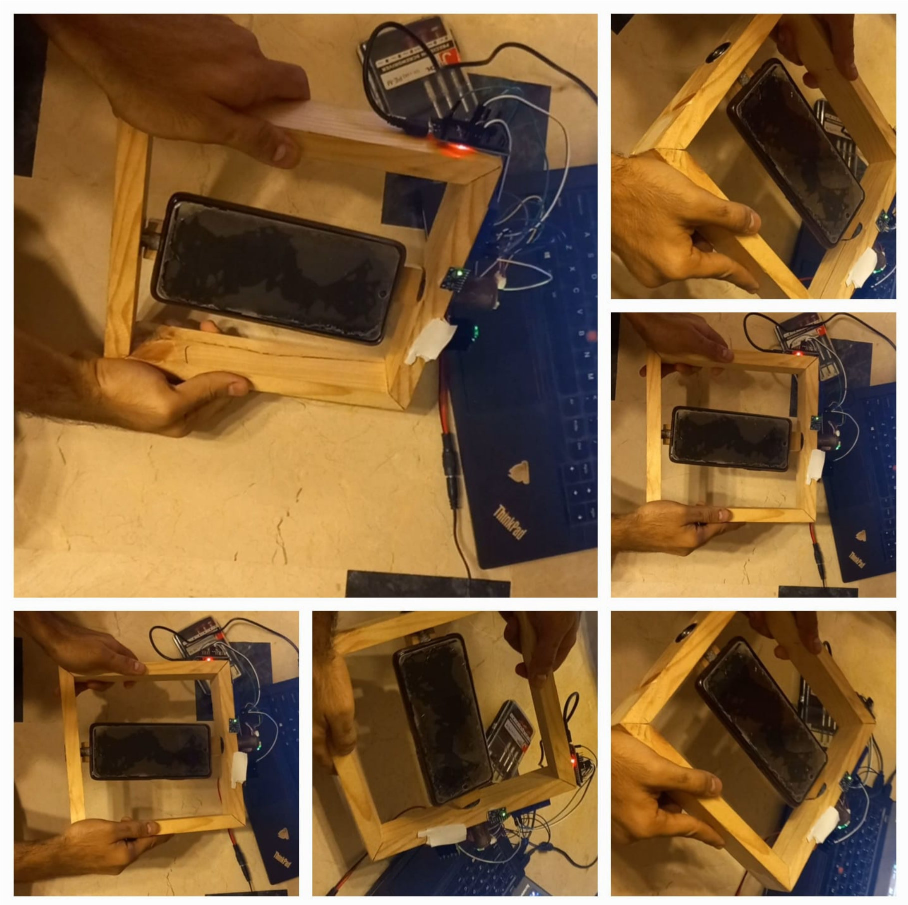


Figure 14: Final Version of Working Model

4 Task Division

Vezhish Ali: Vezhish played a pivotal role in the modeling and simulation components of the project. The approach involved the refinement and expansion of mathematical models utilized in the analysis, along with the implementation of simulations to assess the system's behavior under different conditions. Vezhish also played a key role in tuning the system, and debugging the functionality errors of different components.

S.M Muslim Hussain: Muslim took on the task of doing a comprehensive literature review, meticulously analyzing previous scholarly works to provide a solid groundwork for the project. Furthermore, Muslim made substantial contributions to the progress of the mechanical structure through his essential involvement in the development of its structural framework. Muslim implemented the PID controller and played the most crucial role in developing, designing and debugging phases of the project. From the development of the controller to the debugging of errors in the feedback, Muslim had crucial contributions in every stage of the project.

M. Abdullah: Abdullah placed emphasis on the pragmatic dimension of the project by conducting experiments with the objective of obtaining essential data for the upcoming phase of modeling. The tasks indicated above involved the measurement and analysis of specific variables that are crucial for the development and validation of the models. Moreover, Abdullah played a crucial role in the development and execution of the mechanical framework, making noteworthy contributions in both its design and implementation. Abdullah also played a key role in tuning the system, implementing the controller and debugging the components.

5 References

1. Sharma, J., Hote, Y. V., & Prasad, R. (2020). Robust PID control of single-axis gimbal actuator via stability boundary locus. IFAC-PapersOnLine, 53(1), 27–32. https://doi.org/10.1016/j.ifacol.2020.06.005

2. Lage, V. N., Segundo, A. K. R., & Pinto, T. V. B. E. (2016). Mathematical modelling of a two degree of freedom platform using accelerometers and gyro sensors. Journal of Mechanics Engineering and Automation. https://doi.org/10.17265/2159-5275/2016.08.006