MTE 423

Automatic Control II

Fall 2023

Final Project: 2D Self-Balancing Platform

Submitted by

Yomna Mohamed – 120200175

To

Prof. Ayman Nada

Table of Contents

Table of Figures

1. Introduction

This automatic control project presents a comprehensive system that integrates Arduino Nano microcontroller with an IMU sensor, and two servo motors mounted on an aluminum platform that moves along two axes to achieve precise control over the platform holder's orientation for roll and pitch. The entire system is programmed using LabVIEW.

2. Project Setup

2.1. Components

The utilized components to build this project were:

1. Arduino Nano Microcontroller
2. MPU 6050 Sensor
3. 2 Hitec HS-422 Standard Deluxe Servo Motors
4. Mini Breadboard
5. 2 DOF Aluminum Platform

2.2. Circuit Wiring

The Electronic components were connected as shown in the figure below:

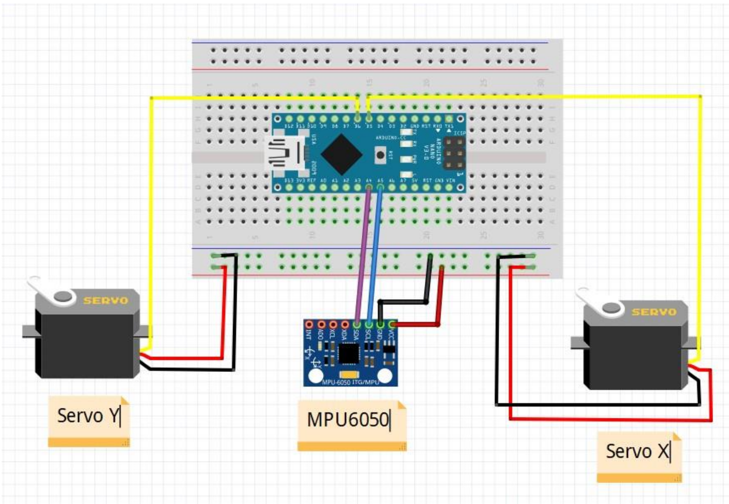

Figure 1: Wiring

2.3. Hardware Setup

The final setup of all the components assembled is shown in 4 views in the figures below:

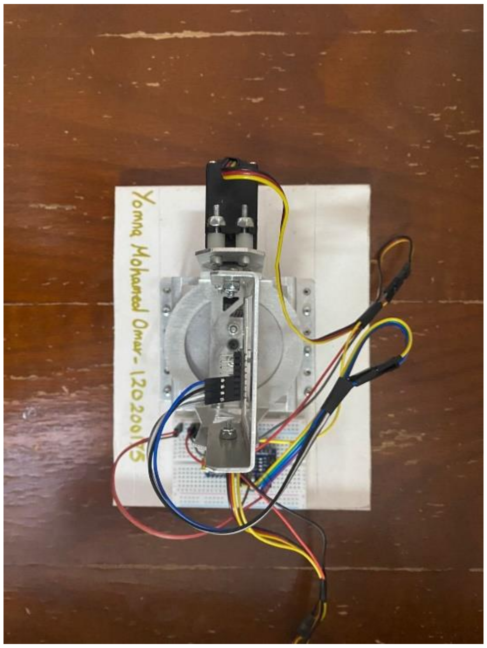

Figure 3: Top View

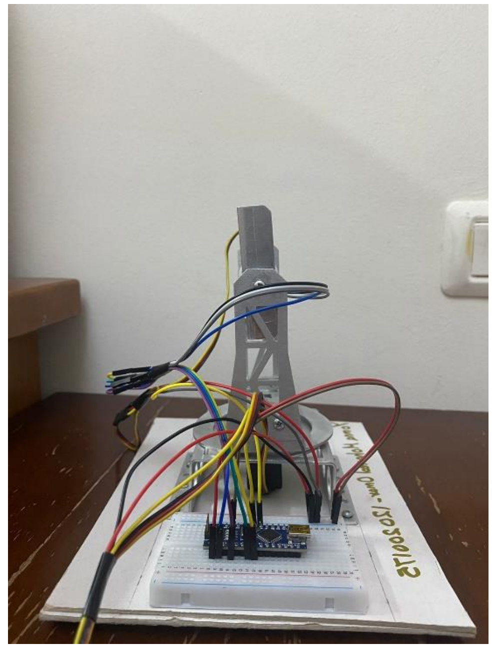

Figure 2: Side View

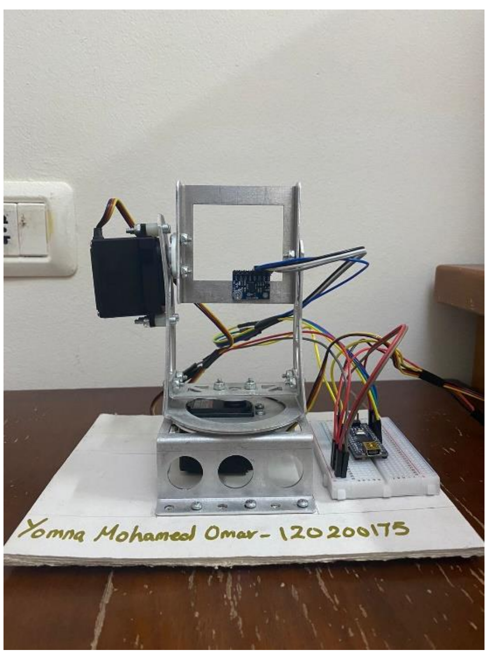

Figure 5: Front View

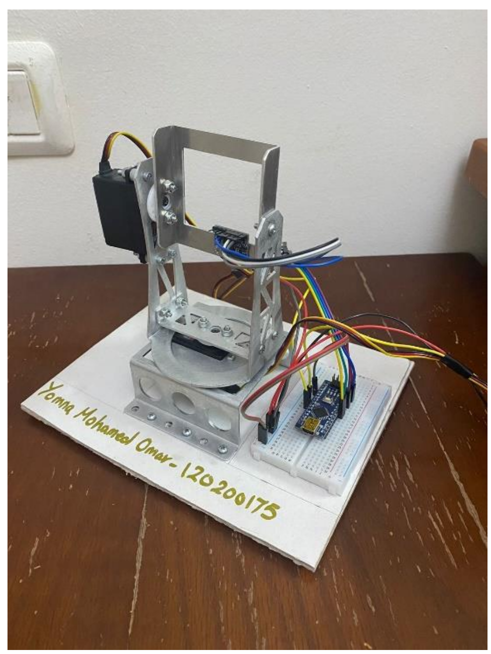

Figure 4: Isometric View

3. LabVIEW Programming

LabVIEW serves as the control interface tool for this project, providing a user-friendly environment to monitor and control the platform. Through the LabVIEW graphical programming interface and LINX toolbox, users can input desired parameters, view platform's real-time position feedback, and adjust control parameters. LabVIEW communicates with the Arduino Nano to receive and process data from IMU sensor, then relay commands to the servo motors.

Development of the project was implemented as a series of tasks divided into sub VI routines and then integrated into one VI file with the panel interface.

3.1. Servo Motors Mapping

First, the datasheet of the HS-422 servo motors was checked to obtain data of its Pulse Width Modulation (PWM) and angle ranges. It operated with PWM starting from 500 to 2500 with 1500 as the neutral position and rotated 190 degrees from 0.

The neutral position of the servos wasn't upright, so it was modified along with the PWM range to be symmetrical around it. Then, an equation was identified for each servo separately to link the input angle with the output PWM sent to the Arduino Nano assuming linearity between both parameters. The results were computed to be [60, -60] degrees and [2115, 885] PWM for y-axis, and [48.75, -48.75] degrees and [1750, 750] PWM for x-axis.

Below, is the block diagram built to implement the equations with both the scale and offset values as well as the minimum and maximum ranges that can be modified according to the desired application.

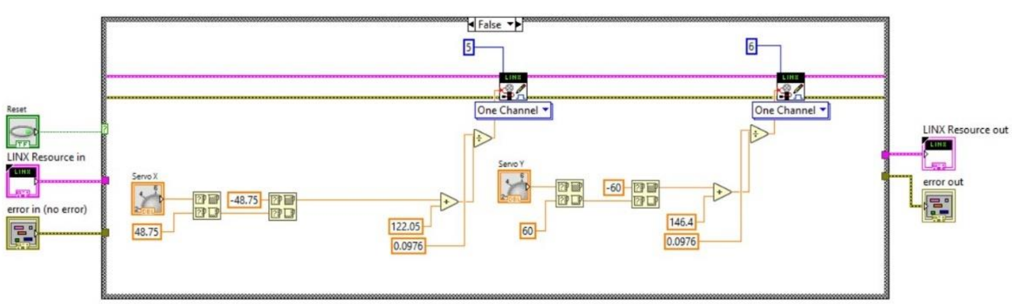

Figure 6: Servo Write Sub-VI

3.2. IMU Sensor Calibration

It is essential to get a smooth signal as an accurate representation of the platform's orientation around the x and y axes to be sent as feedback to the servo motors. The used MPU-6050 utilizes a built-in accelerometer and gyroscope that operate separately and can be used as two different sources of data to compute the angular position for both the pitch and roll.

3.2.1. Accelerometer Readings

Roll and pitch can be computed directly from the acceleration data using both equations below:

$$Roll\ (\theta_x) = arctan2(a_y, a_z) * 57.3$$

$$Pitch\ (\theta_y) = arctan2\left(-a_x, \sqrt{{a_y}^2 + {a_z}^2}\right) * 57.3$$

3.2.2. Gyroscope Readings

Integration can be done to the gyro data of the angular velocities to directly get the roll and pitch values. To do so, a flat sequence was used to perform a constant time step that can be adjusted according to the desired value and multiplied by the gyro data to do integration and collect output in a shift register for the next cycles during the robot's operation.

3.2.3. Filtering Sensor Signals

Data of roll and pitch computed from the gyroscope and accelerometer is then processed using complementary filter with tuned weights to combine them into one filtered output for each axis. After testing the movement of the whole system, the scale and offset for each sensor was altered as well for calibration. Final block diagram of the sensor data is illustrated in the figure below:

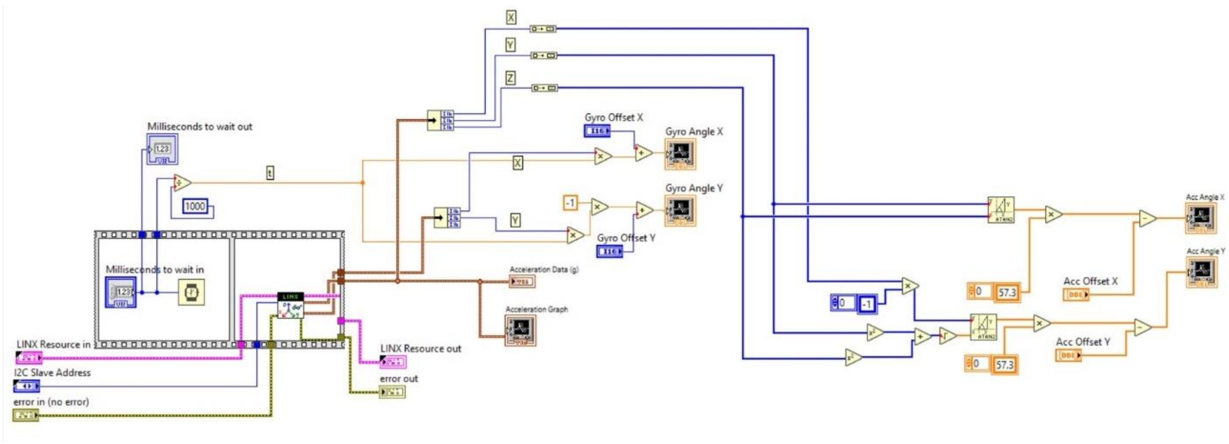

Figure 7: Sensor Readings

3.3. PID Controller Design

A PID Controller was created to take a desired setpoint and feedback signal to obtain error signal that is then processed through the controller which can be P, PD, or PID as desired then sends the actuation signal as an output.

The gains for P and PD were obtained from fine tuning of the parameters through trial and error of the system to get the most optimum results that satisfy requirements of a quick responsive performance with high accuracy and precision along with smooth movement of the servo motors without overshoot.

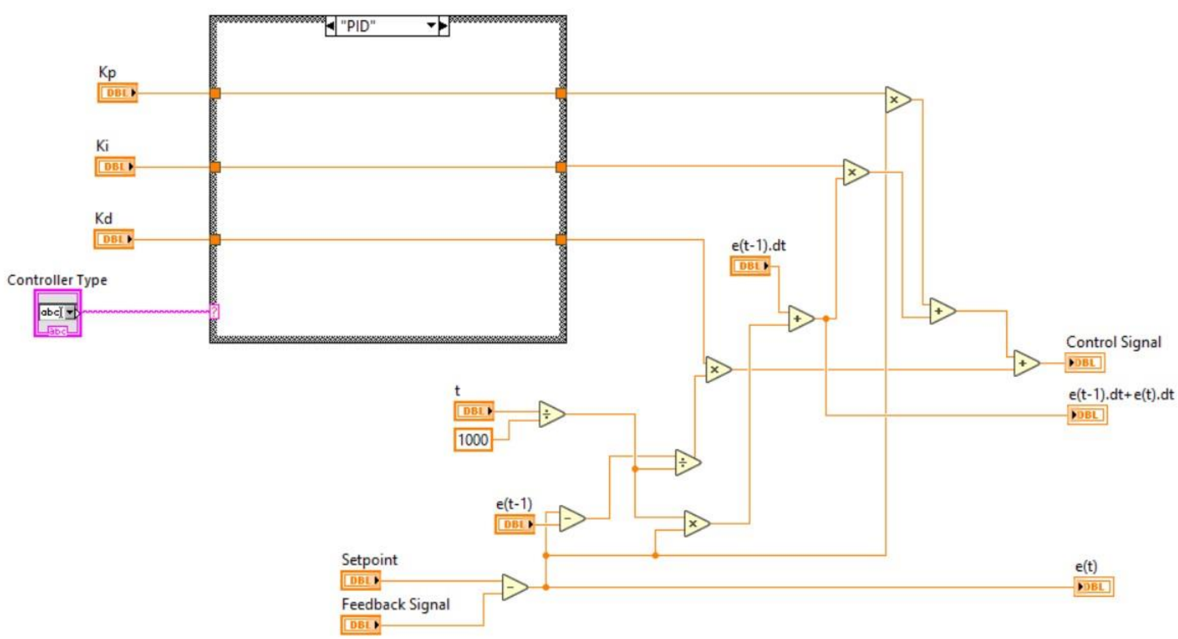

Figure 8: PID Controller

3.4. System Integration

With all the sub-VIs ready, the final file was built to initialize the Arduino Nano, read data from the IMU to be processed, then sent to the PID Controller block, with tuned parameters, that will send the final actuation signal to each of the servo motors that will keep the platform stable at a fixed orientation as its base moves, and plots the data of its orientation during the whole process.

The user-friendly front interface shows clearly all the parameters that can be altered according to the operating conditions and indicators of the system's state is shown in the figure below:

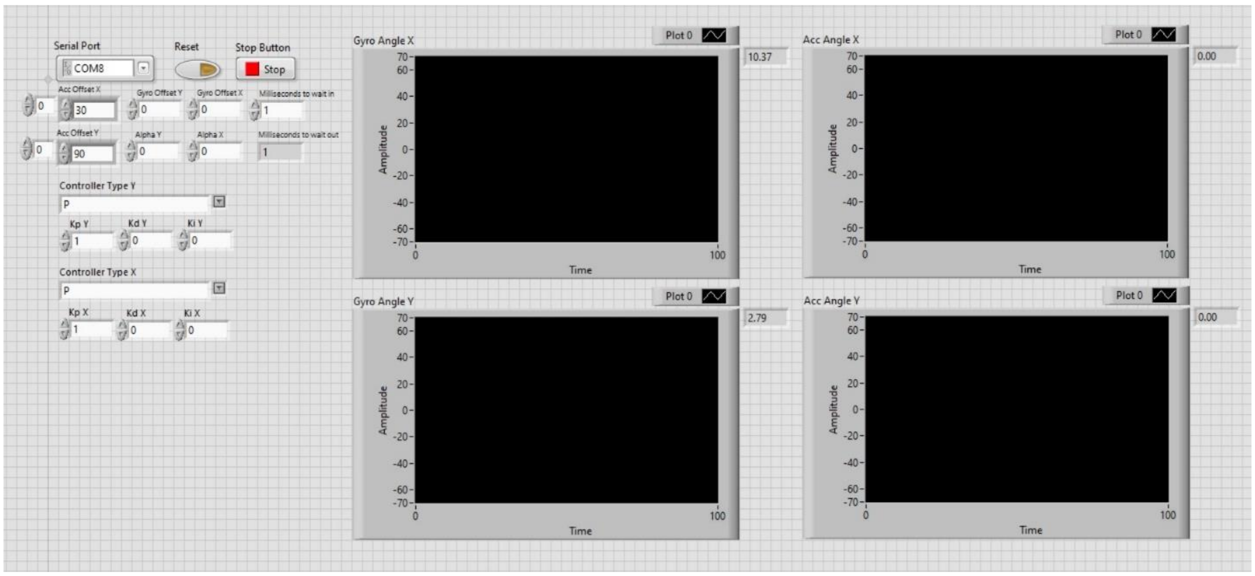

Figure 9: Front Control Panel

4. Conclusion

In conclusion, this project successfully achieved control over the orientation of a platform holder using an Arduino Nano microcontroller. The LabVIEW programming environment provided an intuitive and user-friendly interface for monitoring and controlling the platform. The integration of various components, from servo motor mapping to IMU sensor calibration and PID controller design, resulted in a responsive system.

The servo motors were mapped, taking into consideration the non-upright neutral position and adjusting the PWM range for symmetry. IMU sensor calibration was crucial for obtaining smooth and accurate signals for roll and pitch, with a complementary filter effectively combining accelerometer and gyroscope data. The PID controller design underwent fine-tuning to ensure optimal performance.

The system integration successfully brought together all the components, creating a cohesive and functional unit. The front control panel in LabVIEW provided users with the ability to input desired parameters, view real-time position feedback, and adjust control parameters, enhancing the overall usability of the system.

To suggest further enhancements to improve the system performance it’s suggested to:

1. Develop a mathematical model to further describe and understand the system behavior adequately.
2. Implement state feedback for the position and velocity so that the control system can achieve improved stability, faster response times, and enhanced disturbance rejection, making it more adaptable to different operating conditions and disturbances.
3. Implement a Kalman filter to estimate the true state of the system based on noisy sensor readings from the accelerometer and gyroscope. This will mitigate the limitations of each sensor and provide a more reliable estimate of the roll and pitch angles.