DOI: 10.2478/sbeef-2023-0002

LABVIEW AND REMOTEXY INTEGRATION FOR QUADROTOR STABILIZATION AND CONTROL

[1,2]HAMZA DJIZI, [3]ZOUBIR ZAHZOUH, [4]AZZEDINE BOUZAOUIT
[1]Department of Mechanical Engineering, Mohamed Cherif Messaadia University, P.O. Box 1553, Souk-Ahras, 41000, Algeria.
[2]INFRA-RES Laboratory, Mohamed Cherif Messaadia University, Algeria
[3]Laboratoire de Recherche en Électromécanique et Sûreté de Fonctionnement, LRESF Laboratory, Mohamed Chérif Messaadia University, P.O. Box 1553, Souk-Ahras, 41000, Algeria.
[4]University of 20 August 1955 Skikda, Algeria
E-mail: hamzadjizi@gmail.com

Abstract. ***Nowadays, Small quadcopters have made significant advancements in recent years, thanks to the development of control systems, the availability of sensors, and affordable and reliable materials for their production. Additionally, programs have been developed to model and analyze these aircraft before production. The professional applications of quadcopters are seemingly endless due to their many advantages. The aim of this research is to build a quadcopter and test its stability utilizing Arduino Mega, IMU sensor (Inertial Measurement Unit) and MPU-6050 in LabVIEW environment. The objective is to select the suitable PID parameters and create a remote-control program that can be operated using a smartphone and RemoteXY app on Android OS.***

Keywords: quadcopter, control, LabVIEW, Arduino, sensor.

1. INTRODUCTION

Quadcopters with sensors have developed in terms of their ability to perceive their surroundings, allowing them to fly in appropriate conditions and maintain balance. They consist of four motors with propellers attached to a wooden or other material cross, with each motor connected to an electronic speed controller (ESC) controlled by a control card. The control card receives its commands from a radio control receiver or a smartphone. Designing and building a quadcopter is less complicated than a standard helicopter, but still requires careful consideration of the parts and assembly during the design process. When editing, it is crucial to consider some general points based on experience.

One of the most challenging mounting locations when designing a custom frame is where the motors and frame meet since the four mounting holes there need to be set precisely. Any extra parts should preferably be arranged symmetrically around an axis to make it easy to determine the quadcopter's centre of gravity. The middle of the circle that connects all motors should ideally house the controller (Arduino Mega). As it is heavy enough for the quadcopter, the battery should also be placed in the middle of the device.

Quadrotor control has been an area of interest for many researchers in recent years due to its potential for applications in a wide range of fields such as aerial photography, environmental monitoring, and search-and-rescue operations [1]. PID (Proportional-Integral-Derivative) controllers have been widely used in quadrotor control due to their simplicity and effectiveness in providing stable flight control [2-3]. Previous studies have shown that PID controllers can achieve accurate and stable flight control for quadrotors under various conditions, including disturbances and changing environments [4]. However, researchers have been proposed several regulators for quadrotor control, both linear and nonlinear. Linear regulators such as PD (Proportional-Derivative) and LQR (Linear Quadratic Regulator) have been widely used due to their simplicity and effectiveness in achieving stable flight control [5-6]. Nonlinear regulators such as MPC (Model Predictive Control) [7], SMC (Sliding Mode Control)[8], and fuzzy logic controllers have also shown promising results in achieving stable and robust control of quadrotors under various conditions [9]. Moreover, the use of neural networks has been investigated in quadrotor control as they have the ability to learn complex nonlinear relationships between the inputs and outputs of the system [10].

In this work, we will try to study the stability of a quadrotor using LabVIEW. To achieave that, we need a range of components, including a wooden frame for constructing the quadcopter, an Arduino Mega 2560 microcontroller, four A2212-6T 2200KV motors, four 30A electronic speed controllers (ESC), four 8045 propellers, an MPU-6050 gyroscope, ESP8266 Wi-Fi module, wires, a 3S 4000mAh Li-Po battery, an IMAX B6 Li-Po balance charger (Figure 1). In addition to LabVIEW and an Android OS smartphone equipped with the RemoteXY app. However, Section 2 provides a detailed guide on how to construct a quadrotor and link its components together. Section 3 delves into the control equations and how they can be used to control the quadrotor's movement. Section 4 explains how to implement the control on a LabVIEW environment, while Section 5 discusses the use of RemoteXY to control the quadrotor. Section 6 presents the results and discusses their implications, while the final section, the conclusion, summarizes the article's key points and emphasizes the significance of the findings.

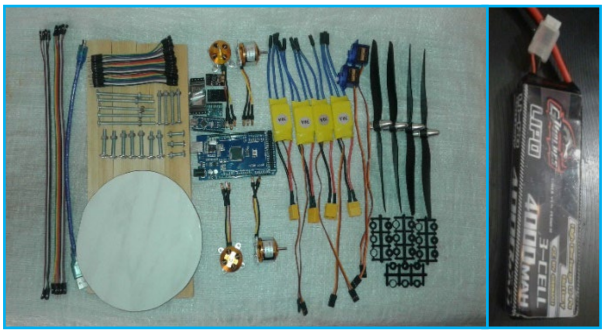


Figure 1. The quadcopter and its basic components.

2. CONSTRUCT THE QUADCOPTER AND LINK ITS COMPONENTS

The quadcopter will be developed in steps, starting with the mechanical assembly and continuing with the functional testing of the parts. The quadcopter's whole structure will be created by connecting the electrical components using connection cables (Jumper-Wire), then assembling them onto the frame. Also, to ensure that the quadcopter can handle the intended weight or achieve the desired thrust, The motors will be selected based on their capabilities to provide the necessary thrust for the quadcopter's weight and flight performance.

Developing an effective command interface in LabVIEW or an interface in Android using the RemoteXY website, it is imperative to gain a comprehensive understanding of the operational and connection mechanisms of the various components of the quadcopter, including but not limited to Arduino, motors, gyroscope, Wi-Fi, and ESC. Only by thoroughly comprehending the functioning of these components and their interplay can we design and implement a robust interface that can cater to the specific needs of the quadcopter and enable efficient control and maneuverability.

Connecting the fundamental electrical components will be established according to the diagram presented in Figure 2, which illustrates the optimal way to link the Arduino Mega, the four A2212-6T 2200KV motors, the MPU-6050 gyroscope, the ESP8266 Wi-Fi module, the four 30A electronic speed controllers (ESC), the Li-Pro battery (3S, 4000 MAH), and the necessary Jumper-Wire cables to ensure the proper functioning of the quadcopter.

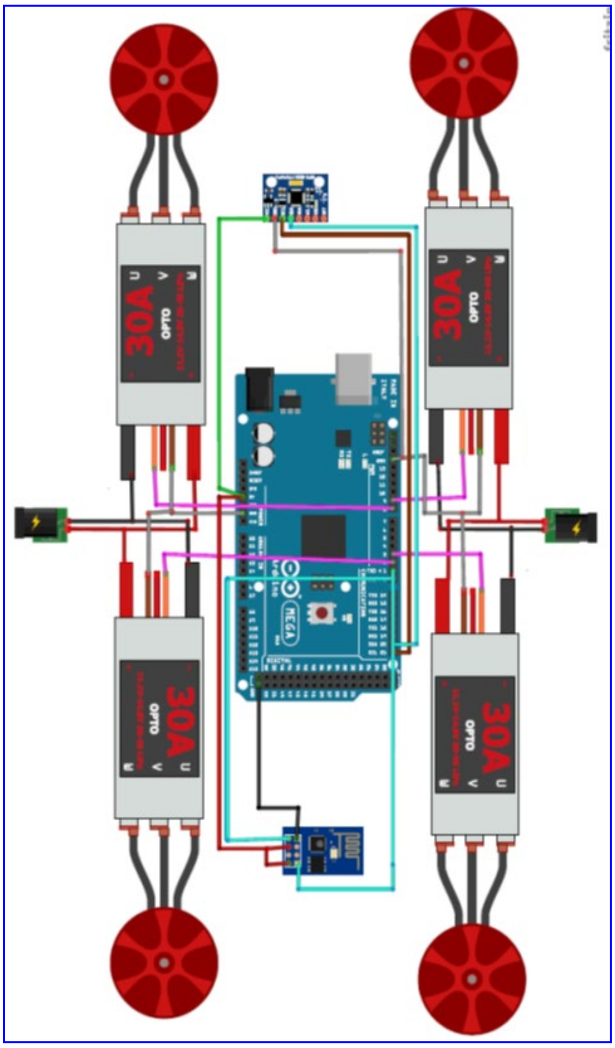


Figure 2. Linking the electronic components.

After assembling the quadrotor, it's essential to carry out a series of tests on the various components to ensure everything is working correctly (Figure 3). This includes establishing a stable WiFi connection using the esp8266 module, initializing the ESC to optimize the performance of the brushless motors, and calibrating the gyroscope to remove any offsets in the six axes. These steps are critical for ensuring a smooth and stable flight, and they must be performed carefully and thoroughly to minimize the risk of any malfunctions or accidents during operation. Once all of these tasks are completed successfully, the quadrotor should be ready for programming and testing and further optimization.

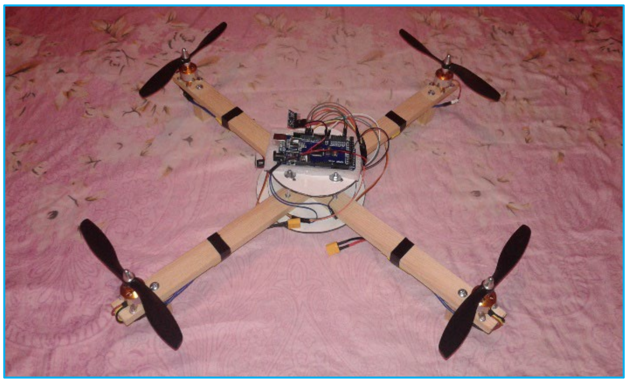


Figure 3. The quadcopter model.

3. CONTROL AND EQUATIONS

Accelerometers and gyroscopes are two commonly used sensors in quadrotors to achieve stabilization and control. The accelerometer measures the linear acceleration of the quadrotor in all three axes. Based on this information, the control system can calculate the orientation of the quadrotor and adjust the motor thrust accordingly to maintain stability. Gyroscopes, on the other hand, measure the angular velocity of the quadrotor around all three axes. By integrating the gyroscopic data, the system can calculate the orientation of the quadrotor over time. Combining the accelerometer and gyroscope data, the control system can achieve accurate attitude estimation of the quadrotor and make necessary adjustments to the motor thrust to stabilize and control the device. The integration of these two sensors provides a robust and reliable control system for quadrotors, enabling them to perform complex maneuvers and maintain stability in various flight conditions.

To comprehensively assess the dynamics and stability of a quadcopter, determining the angles of both accelerometer and gyroscope is essential. To accurately calculate the angles of the gyroscope, employing the equations below is crucial.

$$AngleX = \mathrm{atan}\left(\frac{Y}{\sqrt{X^2+Z^2}}\right) \qquad (1)$$

$$AngleY = \mathrm{atan}\left(\frac{X}{\sqrt{Y^2+Z^2}}\right) \qquad (2)$$

Accelerometer angles:

$$acc\_angle\_x = atan((aY/16384.0)/sqrt(pow((aX/16384.0),2) + pow((aZ/16384.0),2))) * rad\_to\_deg; \qquad (3)$$

$$acc\_angle\_y = atan(-1 * (aX/16384.0)/sqrt(pow((aY/16384.0),2)\ pow((aZ/16384.0),2))) * rad\_to\_deg; \qquad (4)$$

$$acc\_angle\_z = atan(sqrt(pow((aY/16384.0),2) + pow((aX/16384.0),2))/(aZ16384.0)) * rad\_to\_deg; \qquad (5)$$

Gyroscope Angles:

$$Gyro\_angle\_x = gX/131.0; \qquad (6)$$

$$Gyro\_angle\_y = gY/131.0; \qquad (7)$$

$$Gyro\_angle\_z = gY/131.0; \qquad (8)$$

Total Angles:

$$totale\_angle\_x = 0.98 * (total\_angle\_x + Gyro\_angle\_x * elapsedtime) + 0.02 * acc\_angle\_x; \qquad (9)$$

$$totale\_angle\_y = 0.98 * (total\_angle\_y + Gyro\_angle\_y * elapsedtime) + 0.02 * acc\_angle\_y; \qquad (10)$$

$$totale\_angle\_z = totale\_angle\_z + Gyro\_angle\_z * ellapsedtime \qquad (11)$$

Error calculation:

$$error\_x = totale\_angle\_x - desired\_angle\_x \qquad (12)$$

$$error\_y = totale\_angle\_y - desired\_angle\_y \qquad (13)$$

$$error\_z = totale\_angle\_z - desired\_angle\_z \qquad (14)$$

PID equations:

$$pid\_p = kp * erreur; \qquad (15)$$

$$pid\_d = kd * ((erreur - erreur\_précédent)/temps\_écoulé); \qquad (16)$$

$$PID = pid\_p + pid\_i + pid\_d; \qquad (17)$$

With:
aX, aY, aZ: accelerometer variables (axes)
gZ, gY, gZ: gyroscope variables (axes)
kp, ki, kd: PID parameters

4. CONTROL ON LABVIEW ENVIRONMENT

The implementation of a command interface utilizing LabVIEW has facilitated the study of the quadcopter's distinct movements in real-life scenarios. The simulation results derived from this analysis have enabled us to accurately maintain the speed and stability of the quadcopter, thus ensuring optimal flight conditions. The successful integration of these findings led to the development of an ideal interface under the Android platform, allowing for remote control of the quadcopter with unparalleled precision and reliability.

The implementation of the LabVIEW interface was executed through a systematic construction of the block diagram, which was employed to configure all essential functions required for the comprehensive control of the quadcopter's various components, including the Arduino, motors, and MPU6050 (Figure 4). Beginning with the development of the PIDs diagram, algorithm mixer diagram, and followed by the motors and MPU6050 diagrams, the block diagram that based on the equations above enables the creation of a seamless control interface, which represented in Figure 5.

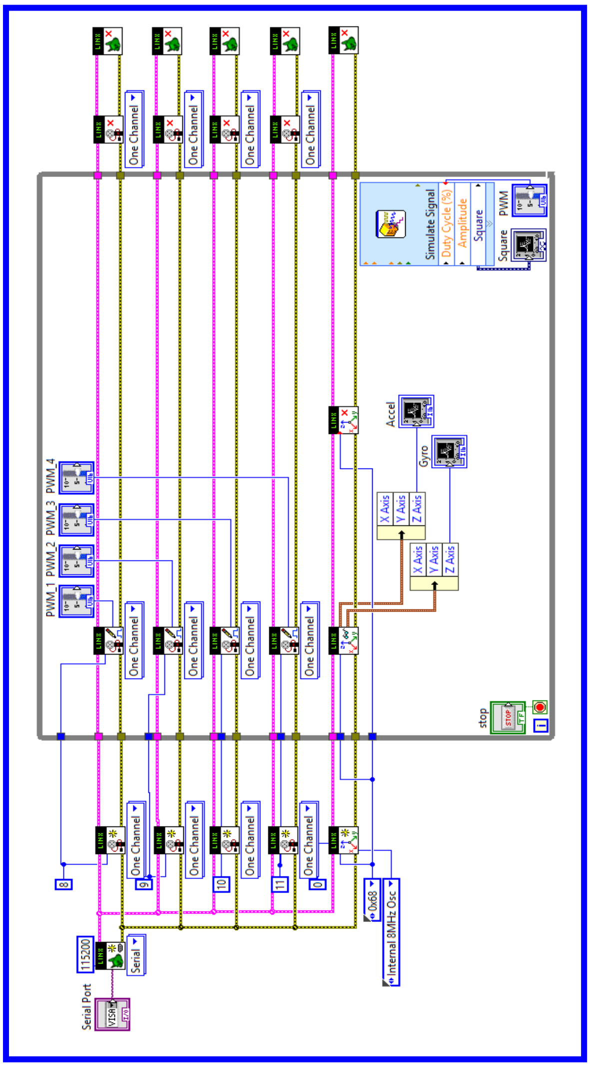
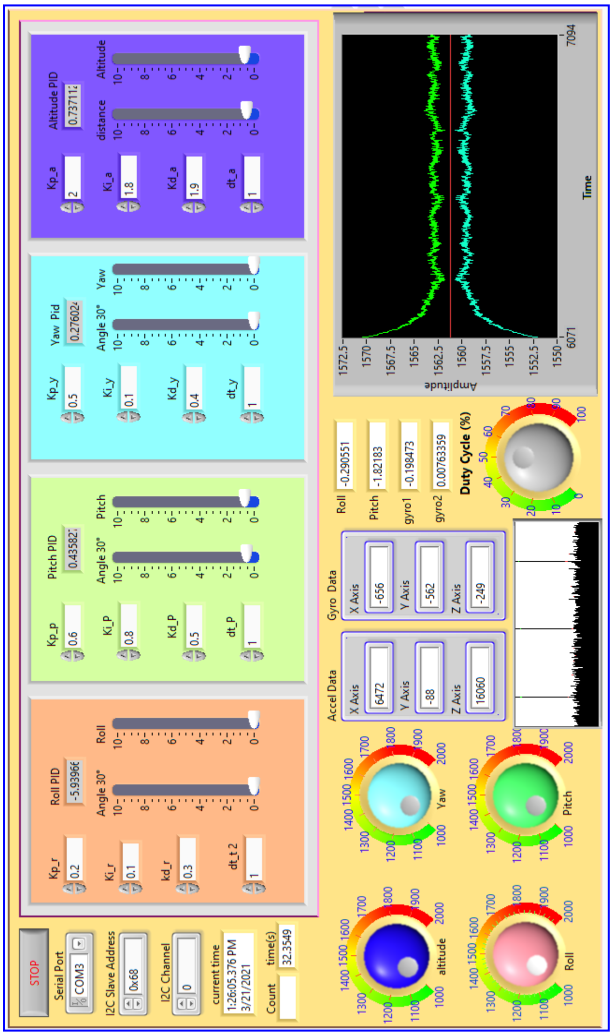


Figure 4. Brushless Motors and gyroscope control diagram (MPU-6050).



Altitude PID 0.73711
Kp_a 2
Ki_a 1.8
Kd_a 1.9
dt_a 1

Yaw Pid 0.27602
Kp_y 0.5
Ki_y 0.1
Kd_y 0.4
dt_y 1

Pitch PID 0.43582
Kp_p 0.6
Ki_P 0.8
Kd_P 0.5
dt_P 1

Roll PID -5.9396
Kp_r 0.2
Ki_r 0.1
kd_r 0.3
dt_t 2 1

Serial Port COM3
I2C Slave Address 0x68
I2C Channel 0
current time 1:26:05.376 PM 3/21/2021
Count
time(s) 32.3549

Roll -0.290551
Pitch -1.82183
gyro1 -0.198473
gyro2 0.00763359

Gyro Data
X Axis -656
Y Axis -562
Z Axis -249

Accel Data
X Axis 6472
Y Axis -88
Z Axis 16060

Figure 5. The Control graphical interface on LabVIEW.

5. CONTROL USING ROMOTXY

In order to program and establish a connection between the Arduino and the website (remotexy.com) and its corresponding phone application (Figure 6). A Wi-Fi connection was selected along with an Arduino Mega device, Wifi-ESP8266 connection module, and the Arduino IDE development environment. To enable communication between the smartphone and the controller (Arduino Mega), it was necessary to connect it to the Wi-Fi-ESP8266 module via the UART inputs (Rx, Tx). The ESP8266 module was then configured as a standalone Wi-Fi access point, requiring no connection with an existing Wi-Fi network for operation. The smartphone must be connected to the access point created to establish a connection. When programming the controller, the equations and variables mentioned about will be utilized to calculate the PID values and the appropriate pulse width modulation (PWM) to enable precise control over each movement.

Figure 6. RemoteXY Command interface on smartphone.

6. RESULTS AND DISCUSSION

The analysis of the quadcopter and its real-time movements, while similar to its simulation counterpart under MATLAB, presents variations due to the potential for measurement inaccuracies and manufacturing discrepancies. As such, the current study employed the LabVIEW environment to investigate the movement of the quadcopter and derive precise values for the essential PIDs parameters, which are crucial for subsequent programming of the controller (Table 1).

Table 1. PID regulator parameters

PID_Roll		PID_Pitch		PID_Yaw		PID_Altitude	
Param	Value	Param	Value	Param	Value	Param	Value
Kp	0.2	Kp	0.6	Kp	0.5	Kp	2
Ki	0.1	Ki	0.8	Ki	0.1	Ki	1.8
Kd	0.3	Kd	0.5	Kd	0.4	Kd	1.5

The simulation results for the quadrotor system using the PID regulator on LabVIEW were highly successful. The altitude results demonstrated consistent and stable flight at the desired height, with minimal oscillations and deviations from the setpoint (Figure 7). The roll and pitch results were also impressive, with the quadrotor maintaining a level and balanced flight despite varying wind conditions and disturbances (Figure 8 and 9). The yaw results were equally noteworthy, showcasing the system's ability to maintain a stable heading and respond quickly to changes in orientation (Figure 10). Overall, the simulation demonstrated the effectiveness of the PID regulator in controlling the quadrotor's movements and ensuring smooth, stable flight performance. These results provide valuable insights for further development and optimization of quadrotor systems for a variety of applications

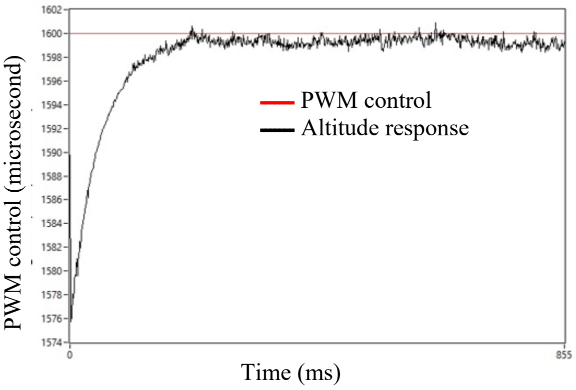


Figure 7. System response for Altitude movement

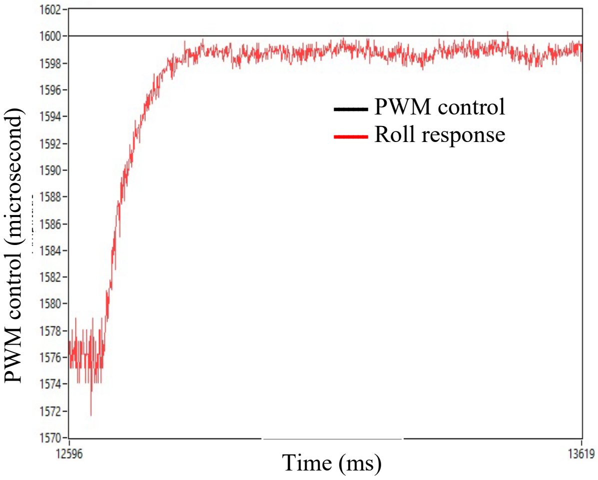


Figure 8. System response for Roll movement

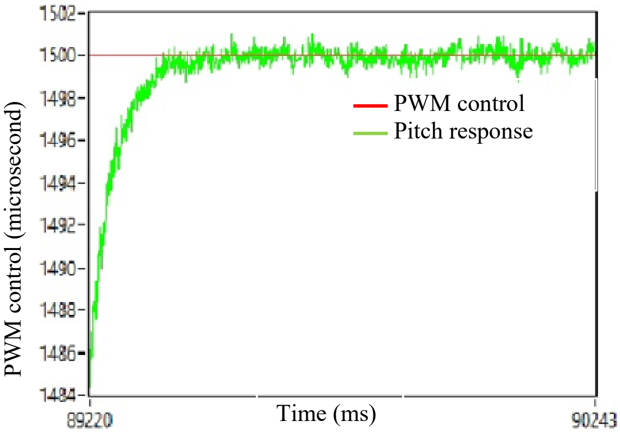


Figure 9. System response for Pitch movement

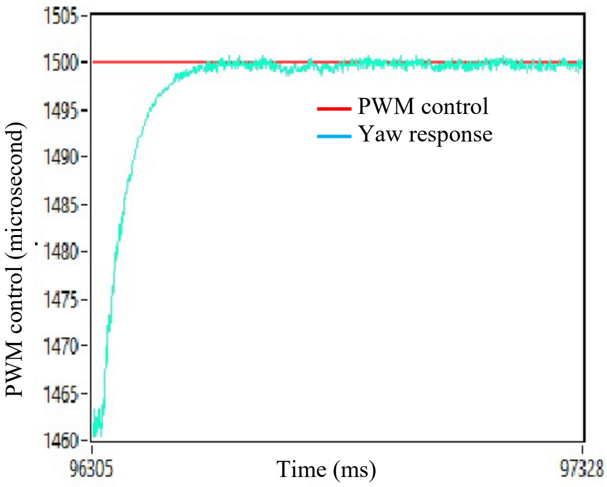


Figure 10. System response for Yaw movement

7. CONCLUSION

This study has successfully presented a system for controlling the various movements of the quadcopter and analyzed the stability of each motion using a PID regulator in the LabVIEW environment. The experimental results demonstrated that the stability of each movement is closely correlated with the speed of the motors. The practical study conducted on LabVIEW enabled us to establish appropriate PID parameters for subsequent controller programming. The IMU sensor (Inertial Measurement Unit, MPU-6050) was identified as a key component for quadcopter controller programming since it can accurately measure all the basic movements of the quadcopter. By utilizing these measurements, we can effectively control and stabilize the quadcopter with remarkable efficiency using Smartphone (RemotXY).

8. REFERENCES

[1] D. Asadi, "Partial Engine Fault Detection and Control of a Quadrotor Considering Model Uncertainty," *Turkish Journal of Engineering*, Mar. 2021, doi: 10.31127/tuje.843607.

[2] L. Jin, Y. Lou, L.-A. Chen, and Q. Lu, "The Unified Tracking Controller for a Tilt-Rotor Unmanned Aerial Vehicle Based on the Dual Quaternion," in *2022 IEEE International Conference on Unmanned Systems (ICUS)*, Oct. 2022, pp. 1356–1363. doi: 10.1109/ICUS55513.2022.9986880.

[3] X. Lu and Z. Xing, "Application of IoT Quadrotor Dynamics Simulation," *Electronics*, vol. 11, no. 4, p. 590, Feb. 2022, doi: 10.3390/electronics11040590.

[4] B. Jiang, B. Li, W. Zhou, L.-Y. Lo, C.-K. Chen, and C.-Y. Wen, "Neural Network Based Model Predictive Control for a Quadrotor UAV," *Aerospace*, vol. 9, no. 8, p. 460, Aug. 2022, doi: 10.3390/aerospace9080460.

[5] S. Martini, S. Sonmez, A. Rizzo, M. Stefanovic, M. J. Rutherford, and K. P. Valavanis, "Euler-Lagrange Modeling and Control of Quadrotor UAV with Aerodynamic Compensation," in *2022 International Conference on Unmanned Aircraft Systems (ICUAS)*, Dubrovnik, Croatia: IEEE, Jun. 2022, pp. 369–377. doi: 10.1109/ICUAS54217.2022.9836215.

[6] L. Martins, C. Cardeira, and P. Oliveira, "Feedback Linearization with Zero Dynamics Stabilization for Quadrotor Control," *J Intell Robot Syst*, vol. 101, no. 1, p. 7, Jan. 2021, doi: 10.1007/s10846-020-01265-2.

[7] D. Yan, W. Zhang, and H. Chen, "Design of a Multi-Constraint Formation Controller Based on Improved MPC and Consensus for Quadrotors," *Aerospace*, vol. 9, no. 2, p. 94, Feb. 2022, doi: 10.3390/aerospace9020094.

[8] F. Ma, Z. Yang, and P. Ji, "Sliding Mode Controller Based on the Extended State Observer for Plant-Protection Quadrotor Unmanned Aerial Vehicles," *Mathematics*, vol. 10, no. 8, p. 1346, Apr. 2022, doi: 10.3390/math10081346.

[9] K. Liu, R. Wang, S. Dong, and X. Wang, "Adaptive Fuzzy Finite-time Attitude Controller Design for Quadrotor UAV with External Disturbances and Uncertain Dynamics," in *2022 8th International Conference on Control, Automation and Robotics (ICCAR)*, Xiamen, China: IEEE, Apr. 2022, pp. 363–368. doi: 10.1109/ICCAR55106.2022.9782598.

[10] J. Li, S. Mou, and D. Zhang, "A Novel Adaptive Robust Control Algorithm for Quadrotor UAV," in *2021 6th International Conference on Robotics and Automation Engineering (ICRAE)*, Guangzhou, China: IEEE, Nov. 2021, pp. 50–54. doi: 10.1109/ICRAE53653.2021.9657806.