Journal of Intelligent Systems and Telecommunications

Journal homepage: https://journal.unesa.ac.id/index.php/jistel/index

Design of Dual Mode Remote Communication Control with Joystick and IMU-Enhanced for Quadcopter Drone

Suko Wiyanto[1], Yandhika Surya Akbar Gumilang[2], Anggit Priaji Susilo Setiawan[1] , R Hardian Eka Anggara [1] , Ahmad Saputra Sumariah[1] , Bagoes Pradana[1] , Winardi Aries Tadianta M[1]*

[1]Polteknik Angkatan Darat, Indonesia
[2]Unversitas Brawjaya, Indonesia

*Correspondence: E-mail: sukowiyanto@poltekad.ac.id

ARTICLE INFO

Article History:
Submitted/Received 8 December 2025
First Revised 25 April 2026
Accepted 28 April 2026
First Available Online 30 April 2026
Publication Date 30 April 2026

Keywords:
Dual-mode controller, Drone control system, IMU gesture control, Joystick input, Quadcopter Drone

ABSTRACT

This study presents the design and implementation of a dual-mode drone control device that integrates a conventional joystick and an inertial measurement unit (IMU)-based gesture control system into a single handheld module. The device was developed to increase operational flexibility, particularly in situations requiring rapid response or more intuitive control inputs. The joystick mode functions as a standard manual control, while the gesture mode interprets the module's orientation as command signals through IMU sensor processing. The system was designed using a multiplexer-based joystick input reader, an IMU orientation processing unit, and an integrated signal transmission protocol to communicate with the drone's flight controller. Performance tests were conducted to evaluate response characteristics, control precision, and stability in both control modes. The results show that the joystick mode provides stable performance with linear throttle response, while the gesture mode delivers fast and responsive orientation-based control. The auxiliary mechanisms for payload release and liquid spraying also functioned reliably without disrupting control performance. Overall, the proposed dual-mode controller meets the design objective s and offers greater operational flexibility compared with single-mode control systems.

1. INTRODUCTION

Unmanned aerial vehicles (UAVs), particularly quadcopters, have emerged as transformative technologies with applications spanning aerial surveillance, precision agriculture, logistics delivery, environmental monitoring, and disaster response operations [1], [2]. The global expansion of UAV deployment has created an urgent demand for control interfaces that balance operational precision with user accessibility, particularly for operators with varying levels of experience [3]. Most control devices currently on the market still rely on a single type of input, such as a conventional joystick, making them less flexible for certain situations that require quick responses or more intuitive control. This indicates a need for more adaptive control devices that can accommodate multiple operating modes[4]. Traditional radio controllers utilizing dual-stick configurations remain the dominant interface paradigm, offering fine-grained manipulation of roll, pitch, yaw, and throttle parameters [5]; however, extensive research has demonstrated that these conventional systems impose substantial cognitive burdens on operators, particularly during complex missions requiring simultaneous attention to multiple flight parameters and environmental conditions. Tezza and Andujar [6], documented that gesture-based control represents an intuitive modality offering advantages including easier learning curves and shorter training requirements compared to joystick-based alternatives. This finding has motivated considerable research efforts toward developing alternative human-drone interaction paradigms that leverage natural human movements rather than abstract stick manipulations.

The integration of Inertial Measurement Units (IMUs) into wearable controllers has emerged as a particularly promising approach for intuitive drone operation. IMUs, comprising accelerometers and gyroscopes, enable the detection of hand orientation and motion patterns that can be mapped directly to drone flight commands. Lee et al. [7] proposed a wearable drone controller utilizing an IMU placed on the back of the hand, combined with machine learning-based gesture recognition and vibrotactile feedback, achieving classification accuracy of 97.9% using ensemble methods and demonstrating that IMU-based motion capture exhibits greater robustness than vision-based tracking under variable lighting conditions. Similarly, Haratiannejadi and Trivedi [8] introduced a smart glove interface for multi-rotor aerial vehicles, wherein hand gestures sensed by IMU and complementary sensors enable rich command vocabularies without reliance on traditional stick mechanisms. Kim et al. [9] investigated the effectiveness of motion-matching controllers that manipulate drone movements based on the controller's own movements, finding that such controllers exhibited superiority over conventional joysticks in task success rate, collision avoidance, and subjective evaluation metrics, particularly when operating in first-person view mode. A number of studies have explored the use of inertial sensors such as IMUs for gesture-based control, but their implementation is generally still separate from the joystick system, or not directly integrated into a single device. In addition, there are not many solutions that combine the two control modes simultaneously so that operators can switch modes without changing devices[10] . This shortcoming presents an opportunity to develop a more flexible and efficient drone control system.

Existing gesture-based controllers typically replace conventional joystick interfaces entirely or exist as separate experimental devices that are not integrated with traditional transmitters, creating operational discontinuities when pilots need to switch between different control modalities. Furthermore, very few studies combine advanced interaction techniques with application-specific payload functions such as package release or liquid dispensing mechanisms within the control interface itself, even though modern drone applications increasingly require such multi-functional capabilities. The majority of current research

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

focuses either on control interface development or payload system design as independent domains, leaving the integration challenge largely unaddressed.

Previous studies on IMU-based drone control have generally focused on replacing conventional joystick interfaces with gesture-based interaction, wearable devices, or glove-based control systems. Although these approaches demonstrate the feasibility of intuitive human-drone interaction, most of them are developed as independent control interfaces and are not directly integrated with conventional joystick-based operation in a single handheld controller. Consequently, operators may experience discontinuity when switching between manual control and gesture-based control modes. In practical UAV operations, particularly in field missions that require both precise manual input and rapid intuitive movement, this limitation reduces operational flexibility.

Therefore, the novelty of this study lies in the development of an integrated dual-mode handheld controller that combines conventional joystick input and IMU-based gesture control in a single compact module. Unlike previous IMU-based controllers that mainly emphasize gesture recognition as a standalone interface, the proposed system allows the operator to switch between joystick and gesture modes without changing the physical controller.

This research proposes the design and implementation of a dual-mode drone control device that integrates a conventional joystick interface and an IMU-based gesture interface within a single handheld module. In the proposed system, the joystick mode supports familiar manual control compatible with established operator training and mission procedures, while the gesture mode exploits IMU orientation detection to provide an alternative control scheme that can be more responsive and intuitive in scenarios such as low-speed maneuvering, close-range inspection, or tasks requiring frequent attitude adjustments. Beyond flight control, the handheld device incorporates actuation interfaces for a payload-release mechanism and a liquid-spraying system, enabling the drone to perform extended operational functions without requiring additional dedicated ground hardware. The main contributions of this work are, first, a fully integrated dual-mode controller that allows seamless switching between joystick and gesture interaction within a single module, eliminating the need to change physical devices during operation; and second, a systematic experimental evaluation of both control modes in terms of response time, positioning precision, and flight stability, grounded in control and human-drone interaction

2. METHODS

2.1. System Architecture

The proposed system was designed as a dual-mode handheld controller for quadcopter operation. The controller integrates two input modalities, namely conventional joystick input and IMU-based gesture input, into a single handheld module. The main hardware components consist of two analog joysticks, an analog multiplexer, an MPU6050 IMU sensor, an Arduino Leonardo microcontroller, a mode-selection switch, a wireless telemetry module, and auxiliary actuator control buttons.

The joystick input is used to generate conventional flight commands, including pitch, roll, yaw, and throttle. Meanwhile, the IMU input is used to estimate the orientation of the handheld controller and convert the detected pitch and roll angles into gesture-based flight commands. The mode-selection switch determines whether the flight command is generated from joystick input or IMU orientation data. The processed command is then transmitted to the quadcopter flight controller through the communication module.

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

In addition to flight control, the handheld controller also provides auxiliary command channels for payload release and liquid spraying mechanisms. These additional functions are controlled independently so that they do not interfere with the main flight command channels.

2.2 Block Diagram

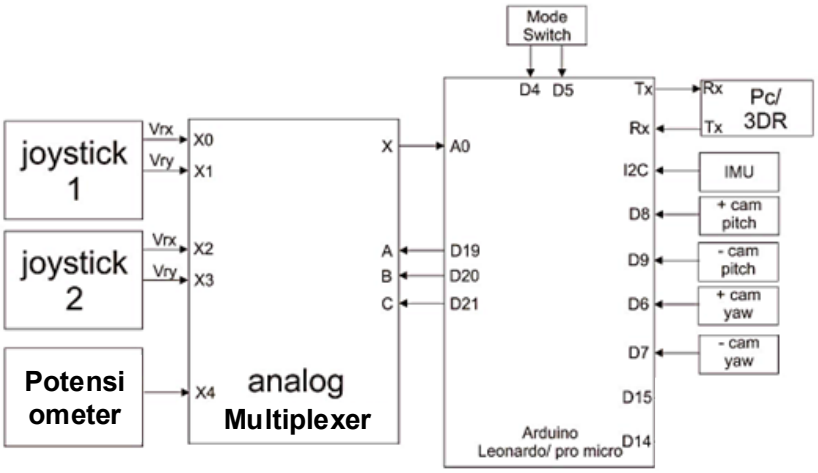

Figure 1 . Block diagram of the quadcopter drone remote control

The diagram illustrates the complete hardware architecture for a dual-mode drone control system. Here is a detailed breakdown of each component and their interconnections:

Input Devices (Left Side)

Joystick 1 & Joystick 2 provide analog voltage outputs representing X and Y axis movements. Each joystick generates two analog signals: Vrx (voltage for X-axis rotation) and Vry (voltage for Y-axis rotation). These represent conventional manual control inputs for flight parameters like roll and pitch (from Joystick 1) and yaw and throttle (from Joystick 2).

Potentiometer (Potensiometer) serves as an auxiliary analog input, producing a single analog voltage output (X4) that can be mapped to functions such as camera tilt control or auxiliary flight parameters. This provides additional operator control flexibility beyond the primary joystick axes.

Signal Processing (Central Component)

The Analog Multiplexer acts as an input selector that reads multiple analog sources (X0, X1, X2, X3, X4) and converts them into digital signals via an analog-to-digital converter (ADC). This component consolidates all analog sensor inputs into a digital format that the microcontroller can process. The multiplexer routes the appropriate joystick or potentiometer signals to the Arduino's analog input pin (A0) in a time-multiplexed fashion.

Control Processor

The Arduino Leonardo microcontroller serves as the primary computational unit. This board is selected for its multiple analog inputs (for joystick and sensor reading), digital I/O pins (for control signals), I2C communication capability (for IMU interfacing), and relatively compact form factor suitable for handheld applications. The Arduino executes the control logic that:

- Reads analog joystick and potentiometer values through the multiplexer
- Communicates with the IMU via I2C protocol to obtain orientation data

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

- Implements mode-switching logic to select between joystick and gesture control
- Generates output control signals
- Manages payload actuation timing

Mode Selection and Communication

The Mode Switch allows the operator to toggle between joystick control and IMU-based gesture control modes. When activated, the digital switch sends signals (D4, D5) to the microcontroller, enabling software-based mode switching without physical disconnection of hardware components.

The Transmitter (Tx) and Receiver (Rx) paths represent the wireless communication link between the handheld controller and the drone or ground station. The system implements a two-way communication protocol for:

- Sending flight commands from the controller to the drone
- Receiving telemetry feedback from the drone back to the controller

The PC/3DR interface suggests compatibility with standard drone communication protocols used by popular flight control stacks, enabling integration with existing drone platforms.

IMU and Sensor Array (Right Side - I2C Bus)

The I2C bus (Serial Clock and Serial Data lines) connects multiple sensors to the Arduino Leonardo Pro Micro in a daisy-chain configuration, allowing multiple devices to communicate using just two wires:

- IMU (Inertial Measurement Unit) measures three-axis acceleration and angular velocity, providing the orientation data necessary for gesture-based control. The IMU sends raw sensor readings to the microcontroller, which then processes this data to detect hand gestures or tilt angles corresponding to desired drone movements.
- Camera Pitch Control (labeled "+ cam pitch" and "- cam pitch") manages gimbal pitch rotation for adjusting camera elevation angle. These are typically servo control signals that tilt the camera mounting platform.
- Camera Yaw Control (labeled "+ cam yaw" and "- cam yaw") manages gimbal yaw rotation, enabling horizontal camera panning.

Data Flow Summary

1. Input Stage: Joysticks and potentiometer generate analog voltages representing operator intent
2. Multiplexing Stage: Analog multiplexer sequences these inputs to the Arduino's single ADC input
3. Processing Stage: Arduino Leonardo Pro Micro reads multiplexed analog values and IMU orientation data via I2C
4. Mode Logic Stage: Software determines whether to use joystick values or IMU-derived gesture commands based on mode switch position
5. Control Generation Stage: Arduino generates appropriate output signals for flight control commands

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

This architecture enables seamless operator switching between two distinct control paradigms (joystick and gesture) while maintaining integrated control over both flight parameters and application-specific payload functions within a single handheld module.

2.3 Joystick Input Acquisition and Normalization

The joystick produces analog voltage signals corresponding to the displacement of the stick along the X and Y axes. These analog signals are read by the microcontroller through an analog-to-digital converter. Since multiple analog inputs are used, an analog multiplexer is employed to sequentially select the joystick and throttle channels.

The raw ADC value from each joystick channel is normalized into a command range suitable for quadcopter control. The normalization process is expressed as:

$$u_j = \frac{ADC_j - ADC_{center}}{ADC_{max} - ADC_{center}} \tag{1}$$

where $u_j$is the normalized joystick command, $ADC_j$is the measured ADC value, $ADC_{center}$is the joystick value at the neutral position, and $ADC_{max}$is the maximum ADC value. The resulting normalized command ranges from -1 to +1 for pitch, roll, and yaw channels.

For throttle control, the ADC value is mapped into a unidirectional command range from 0 to 1 using:

$$u_t = \frac{ADC_t - ADC_{min}}{ADC_{max} - ADC_{min}} \tag{2}$$

where $u_t$is the normalized throttle command, $ADC_t$is the throttle slider reading, $ADC_{min}$is the minimum ADC value, and $ADC_{max}$is the maximum ADC value.

The normalized joystick commands are then mapped into flight command channels as follows:

$$Pitch_{cmd} = K_p u_{pitch} \tag{3}$$

$$Roll_{cmd} = K_r u_{roll} \tag{4}$$

$$Yaw_{cmd} = K_y u_{yaw} \tag{5}$$

$$Throttle_{cmd} = K_t u_t \tag{6}$$

where K_p, K_r, K_y, and $K_t$are scaling factors used to adjust the sensitivity of each control channel.

2.4 IMU-Based Orientation Estimation

In gesture mode, the quadcopter movement command is generated from the orientation of the handheld controller. The MPU6050 IMU sensor provides acceleration and angular velocity data along three axes. The accelerometer data are used to estimate the tilt angle, while the gyroscope data are used to estimate angular changes over time.

The pitch and roll angles from the accelerometer are calculated using:

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

$$\theta_{pitch} = \tan^{-1}\left(\frac{a_y}{\sqrt{a_x^2 + a_z^2}}\right) \tag{7}$$

$$\theta_{roll} = \tan^{-1}\left(\frac{-a_x}{a_z}\right) \tag{8}$$

where a_x, a_y, and $a_z$are acceleration components measured by the IMU sensor.

To reduce noise and improve angle estimation stability, a Kalman filter is applied. The prediction stage is formulated as:

$$\hat{\theta}_k^- = \hat{\theta}_{k-1} + \omega_k \Delta t \tag{9}$$

$$P_k^- = P_{k-1} + Q \tag{10}$$

where $\hat{\theta}_k^-$is the predicted angle, $\hat{\theta}_{k-1}$is the previous estimated angle, $\omega_k$is the gyroscope angular velocity, $\Delta t$is the sampling time, $P_k^-$is the predicted covariance, and $Q$is the process noise covariance.

The correction stage is expressed as:

$$K_k = \frac{P_k^-}{P_k^- + R} \tag{11}$$

$$\hat{\theta}_k = \hat{\theta}_k^- + K_k(\theta_{acc} - \hat{\theta}_k^-) \tag{12}$$

$$P_k = (1 - K_k)P_k^- \tag{13}$$

where $K_k$is the Kalman gain, $R$is the measurement noise covariance, $\theta_{acc}$is the angle estimated from the accelerometer, and $\hat{\theta}_k$is the corrected angle estimate.

2.5 Command Mapping for Quadcopter Control

The estimated pitch and roll angles from the IMU are converted into quadcopter movement commands. A dead-zone threshold is applied to prevent small unintentional hand movements from generating flight commands. The gesture command is defined as:

$$u_g = \begin{cases} 0, & |\theta| < \theta_d \\ \dfrac{\theta}{\theta_{max}}, & |\theta| \geq \theta_d \end{cases} \tag{14}$$

where $u_g$is the normalized gesture command, $\theta$is the estimated tilt angle, $\theta_d$is the dead-zone threshold, and $\theta_{max}$is the maximum tilt angle used for command saturation.

The IMU-based pitch and roll commands are then generated as:

$$Pitch_{cmd} = K_{g,p}\frac{\theta_{pitch}}{\theta_{max}} \tag{15}$$

$$Roll_{cmd} = K_{g,r}\frac{\theta_{roll}}{\theta_{max}} \tag{16}$$

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

where $K_{g,p}$and $K_{g,r}$are gesture sensitivity gains for pitch and roll commands, respectively.

In this study, the gesture mode was limited to pitch and roll control because these axes are directly related to the natural tilting motion of the handheld controller. Yaw and throttle commands remain controlled using dedicated joystick or slider inputs to maintain safety and prevent unintended altitude or heading changes.

2.6 Dual Mode Algorithm

The controller uses a mode-selection switch to determine the active input source. When joystick mode is selected, the command values are obtained from the joystick and throttle channels. When gesture mode is selected, the pitch and roll commands are obtained from the IMU orientation estimation, while throttle and yaw remain controlled through conventional input channels.
The command selection logic is expressed as:

$$C_{out} = \begin{cases} C_{joystick}, & Mode = 0 \\ C_{IMU}, & Mode = 1 \end{cases} \quad (17)$$

where $C_{out}$is the output command sent to the quadcopter, $C_{joystick}$is the command generated from joystick input, and $C_{IMU}$is the command generated from IMU-based gesture input.
The simplified algorithm of the dual-mode control system is described as follows:

Algorithm 1 Dual-Mode Control Algorithm

Input : Joystick input J, throttle input T, IMU data I, mode switch M
Output : Flight command C = {Pitch, Roll, Yaw, Throttle}

Initialize J, T, I, multiplexer, and communication module
Set dead-zone threshold θd and saturation limit θmax

while system is running do
    Read mode switch M
    if M = 0 then
        Read joystick and throttle ADC values
        Normalize ADC values into pitch, roll, yaw, and throttle commands
        C ← Cjoystick
    else if M = 1 then
        Read accelerometer and gyroscope data from IMU
        Estimate pitch and roll angles using Kalman filter
        Apply dead-zone and saturation limits
        Convert IMU angles into pitch and roll commands
        Combine IMU-based pitch and roll with joystick-based yaw and throttle
        C ← CIMU

    end if
    Send C to the flight controller
end while

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

2.7 System Accuracy

The accuracy calculation of the system in this study uses equations found in the Kalman Filter. The Kalman Filter works in two main stages, namely prediction and correction. In the context of IMU, the angle is predicted based on gyroscope data using equation (1), while the covariance uses equation (2).

$$\widehat{x_k} = x_{k-1} + \omega \, . \, dt \quad (18)$$

$$P_k = P_{k-1} + Q \quad (19)$$

Explanation:

$\widehat{x_k}$ = pitch or roll angle at time k
x_{k-1} = Previous angle
P_k = Current covariance
P_{k-1} = Previous covariance
Q = Process disturbance constant (uncertainty in the system model)
ω = Angular velocity of the gyroscope
dt = Sample time interval

3. RESULTS AND DISCUSSION

3.1. Hardware Implementation Results

The developed control device was successfully realized as a compact handheld module, making it easy to operate in the field. The physical component layout is shown in Figure 9 front view, rear view and side view. Each figure shows the structure of the device and the placement of each electronic module.

Figure 2. IMU-Enhances Remote Control

This system combines several key components consisting of two analog joysticks, a throttle slider, four function buttons, an MPU6050 IMU, an Arduino Leonardo microcontroller, a 3DR module, and a Li-Ion 18650 battery-based power supply with a 5V converter.

This configuration demonstrates that all subsystems, from motion sensors, manual inputs, signal processing, to wireless communication, can work together seamlessly without interference between modules. In addition, the device design allows the operator to switch between joystick mode and gesture mode using the IMU without replacing devices or adjusting connections, thereby increasing flexibility in drone operation.

3.2 Calibration and Functional Validation

3.2.1 Joystick Calibration

The joystick calibration process is carried out using joy.cpl to ensure that each axis responds linearly and stably to the mechanical movement of the lever. The test results shown in Table 1 indicate that the analog values read by the ADC are consistent with the HID output, so that the joystick movements can be accurately interpreted by the microcontroller.

Table 2. Joystick Calibration Results

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

Test Conditions	Pitch	Roll	Yaw	Throttle	Output joy.cpl
Neutral Position	0	0	0	0	5000
Negative Maximum	-127	-127	-127	50	1000
Positive Maximum	127	127	127	255	10000

The consistency between the ADC values and the joy.cpl display indicates that the CD4051 multiplexer circuit and the analog reading process function without significant noise. Thus, the joystick is capable of providing smooth, precise, and reliable input to control the drone's direction of movement.

3.2.2 IMU Calibration and Accuracy

IMU accuracy testing was conducted by comparing the pitch and roll angles generated by the MPU6050 with the reference angles obtained from manual measurements. After the data was processed using a Kalman Filter, the resulting angle values became more stable and consistent, as shown in Table 2.

Table 3. IMU Accuracy Results

Tilt Condition	IMU Reading (°)	Reference Angle (°)	Output joy.cpl
Neutral	0	0	5000
Negative slope	-37 to -38	-40	100
Positive Slope	39	40	1000

The difference between the measured angle and the reference angle is within a small range, indicating that the IMU is capable of detecting orientation with sufficient accuracy to be used as a gesture-based controller. With a maximum error rate of around 7.5%, the IMU sensor is still suitable for use in drone motion control applications.

3.3 Throttle Characteristics and Drone Lift Dynamics

Throttle testing was conducted to see how changes in the ADC value on the slider affect the motor rotation speed and drone lift force. The test results in Table 3 show an almost linear relationship between throttle increase and RPM increase.

Table 4. Throttle Response During Takeoff

ADC Value	Throttle (%)	Motor Speed (RPM)	Vertical Speed (m/s)
0	0	0	0
64	25	1200	0
128	50	4000	3.1
153	60	5000	3.7
194	76	8000	7.5
217	85	10,000	9.5
255	100	11,910	10

The drone begins to generate lift at a throttle position of approximately 50%, with motor speed reaching 4000 RPM and vertical speed of approximately 3.1 m/s. This value is consistent with the general characteristics of brushless motors, where thrust only increases significantly after passing the midpoint of the PWM range. Higher throttle increases result in a gradual increase in RPM, making altitude control easier and more responsive.

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

Table 5. Throttle Response During Landing

ADC Value	Throttle (%)	Motor Speed (RPM)	Drone Condition
128	50	4000	Hover
64	25	1200	Down
0	0	0	Landing

Table 4 shows the throttle response during the landing process. At a throttle range of 25%, the motor speed drops to around 1200 RPM, causing the drone to begin a steady descent. When the throttle approaches 0%, the drone can land safely without overshooting. This response pattern shows that the throttle slider is capable of providing smooth input changes, allowing the system to maintain stability both during takeoff and landing.

3.4 Drone Movement Response Using a Joystick

Attitude control testing using a joystick was conducted to evaluate the response speed and stability of the drone's movement when receiving pitch and roll commands. The test results in Table 5 show that the angle changes produced by the drone are consistent with the operator's commands, both in forward-backward and right-left movements.

Table 6. Attitude Response Using Joystick

Command	Response Time (ms)	Angle (°)	Actual Movement
Pitch +	1400	25	Forward
Pitch -	1100	-25	Back
Roll +	1200	25	Right
Roll -	1200	-25	Left

The relatively symmetrical angular response at ±25° indicates that the joystick-based control system has a balanced input distribution between the pitch and roll axes. In addition, the response time is in the range of 1100–1400 ms, which is still suitable for manual operation on APM-based flight controllers. This value indicates that the analog reading, signal processing, and transmission to the drone can run without significant delays so that the controls remain responsive and stable.

The response time in joystick mode was measured in the range of 1100–1400 ms. It should be emphasized that this value represents the end-to-end observable movement response time, not the internal latency of the electronic control signal. The measurement was taken from the moment the joystick command was applied until the drone showed a visible pitch or roll movement. Therefore, the measured value includes sensor reading, microcontroller processing, wireless communication, flight controller interpretation, motor acceleration, and the physical movement dynamics of the quadcopter.

In low-level UAV control systems, signal latency is typically expected to be in the millisecond range. However, the present study evaluates the practical response perceived at the system level during field operation. Thus, the reported response time reflects the observable operational response of the complete drone-controller system rather than the computational delay of the controller alone. The joystick mode produced consistent pitch and roll movements of approximately ±25°, indicating that the command mapping and signal transmission were stable during the test.

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

3.5 Drone Movement Response Using Gesture Mode (IMU)

Gesture mode testing was conducted to see how changes in the orientation of the control device were directly translated into pitch and roll commands on the drone. The test results in Table 6 show that the angle response produced by the drone was in line with changes in the IMU tilt, both in the pitch and roll directions.

Table 7. Attitude Response Using IMU

Command	Response Time (ms)	Angle (°)	Actual Movement
Pitch + (IMU)	1000	25	Forward
Pitch - (IMU)	1200	-25	Back
Roll + (IMU)	1100	25	Right
Roll – (IMU)	1000	-25	Left

The response time in gesture mode is in the range of 1000–1200 ms, which is faster than joystick mode. This difference occurs because the angle data from the IMU is processed directly without going through an analog reading stage, resulting in a shorter signal processing path. Additionally, the angles generated by the drone are consistent at ±25°, indicating that the IMU reading algorithm and telemetry data transmission are stable without significant fluctuations.

In gesture mode, the response time was measured in the range of 1000–1200 ms. Similar to joystick mode, this value represents the observable end-to-end movement response of the quadcopter. The slightly faster response in gesture mode may be attributed to the direct conversion of IMU orientation data into pitch and roll commands, whereas joystick mode requires analog signal acquisition through the multiplexer before command generation.

The drone movement remained consistent at approximately ±25° for both pitch and roll commands. This result indicates that the IMU orientation estimation, Kalman filtering process, command mapping, and wireless transmission were able to generate stable movement commands. Nevertheless, the response time values should not be generalized as internal UAV control-loop latency because the measurement method used in this study was based on visible motion response. Future work should include direct signal-level latency measurement using an oscilloscope or embedded timestamp logging to distinguish between electronic latency, communication delay, motor response delay, and physical movement delay.

3.6 Discussion

Overall, the test results show that the developed dual-mode drone control system is able to work stably and responsively in all test scenarios. In terms of input precision, the joystick provides consistent readings across the entire range of motion, while the IMU produces orientation detection with an error rate of less than 7.5%, which is still acceptable for gesture-based control applications.

In terms of response time, the gesture mode using IMU showed a response time of around 1000 ms, slightly faster than the joystick, which was in the range of 1100–1400 ms. This indicates that inertial sensor-based data processing is more efficient because it does not go through an analog reading stage, resulting in a faster reaction.

The drone's movement performance is also stable, with pitch and roll angles remaining within the range of ±25°, indicating that both control modes are capable of maintaining consistent motion balance. In addition, all modules such as the IMU, joystick, multiplexer,

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

servo, and telemetry function in an integrated manner without communication conflicts, supporting overall system operation.

The experimental results demonstrate that the proposed dual-mode controller can provide stable operation in both joystick and gesture modes. The joystick mode offers familiar and precise manual control, while the gesture mode provides a more intuitive input mechanism based on the natural tilting motion of the handheld module. Compared with previous IMU-based drone controllers that mainly function as standalone gesture interfaces, the proposed system provides a more flexible control architecture by integrating both input modes into one device.

The main advantage of this integration is operational continuity. The operator does not need to replace the controller or change hardware connections when switching between conventional control and gesture-based control. This feature is particularly useful in mission scenarios that require both fine manual control and rapid intuitive movement, such as inspection, payload delivery, and short-range maneuvering.

However, the measured response time should be interpreted carefully. The values of 1000–1400 ms do not represent the internal signal latency of the controller, but rather the observable movement response of the complete system. Since the measurement includes mechanical and flight dynamics effects, future studies should perform more detailed latency analysis by separating sensor acquisition delay, microcontroller processing time, wireless transmission delay, flight controller response, ESC response, and motor acceleration delay.

4. CONCLUSION

This study has presented the design and implementation of a dual-mode handheld controller for quadcopter operation by integrating conventional joystick input and IMU-based gesture control into a single compact module. The proposed system enables operators to switch between manual and gesture-based control modes without changing the physical controller, thereby improving operational flexibility.

The joystick mode provided stable and consistent manual input for pitch, roll, yaw, and throttle control. Meanwhile, the IMU-based gesture mode successfully converted handheld orientation into pitch and roll commands using orientation estimation and command mapping. The IMU calibration results showed an error of less than 7.5%, indicating that the sensor was sufficiently accurate for tilt-based command generation.

The response time obtained in this study ranged from approximately 1000 to 1400 ms. This value represents the observable end-to-end movement response of the complete system rather than the internal electronic latency of the controller. Therefore, the result should be interpreted as a practical system-level response measurement. Future research should include signal-level latency measurement, improved command filtering, and more extensive flight testing under different maneuvering conditions.

Overall, the proposed dual-mode controller demonstrates the feasibility of integrating joystick and IMU-based gesture control in a single handheld device. This integration offers a practical alternative for improving adaptability, usability, and mission flexibility in quadcopter control applications.

5. AUTHORS' NOTE

The authors declare that there is no conflict of interest regarding the publication of this article. The authors confirmed that the paper was free of plagiarism.

6. AUTHORS CONTRIBUTION

S.W. contributed to conceptualization, system design, methodology, manuscript writing, and overall research supervision. Y.S.A.G. contributed to to documentation, figure preparation,

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X

and technical support. W.A.T.M. contributed to manuscript review, editing, and final validation. A.P.S.S. contributed to data acquisition, system calibration, and validation. R.H.E.A. contributed to drone platform preparation and flight testing. A.S.S. contributed to data analysis and interpretation of experimental results. B.P. contributed hardware implementation, controller integration, and experimental testing. All authors have read and approved the final version of the manuscript .

7. AI USE AND DECLARATION OF GENERATIVE AI USE

The author uses the Turnitin tool at the end of the finishing paper, this process aims to determine the level of similarity of the paper that has been compiled.

8. REFERENCES

[1] J. I. Chen and H. Lin, "applied sciences Performance Evaluation of a Quadcopter by an Optimized Proportional – Integral – Derivative Controller," 2023.

[2] M. A. Bin Rodzoan and A. Shah, "Enhancing Business Process Management (BPM) Sustainability in the Agriculture Sector through Precision Agriculture, IoT, and UAV Technologies," *ICETAS 2024 - 9th IEEE International Conference on Engineering Technologies and Applied Sciences*, 2024, doi: 10.1109/ICETAS62372.2024.11120188.

[3] P. Tanpa, A. Menggunakan, P. S. Ardiantara, R. Sumiharto, and S. B. Wibowo, "Purwarupa Kontrol Kestabilan Posisi dan Sikap pada Pesawat Tanpa Awak Menggunakan IMU dan Algoritma Fusion Sensor Kalman Filter," vol. 4, no. 1, pp. 25–34, 2014.

[4] N. Setyasaputra, "6-DOF DAN METODE QUATERNION UNTUK APLIKASI AERO ROBOT (DESIGN AND IMPLEMENTATION OF FLIGHT CONTROLLER USING IMU 6-DOF AND QUATERNION METHOD FOR AERO ROBOT APPLICATION)," pp. 93–106.

[5] R. Kılıç, N. Kumbasar, E. A. Oral, and I. Y. Ozbek, "Drone classification using RF signal based spectral features," *Engineering Science and Technology, an International Journal*, vol. 28, p. 101028, Apr. 2022, doi: 10.1016/J.JESTCH.2021.06.008.

[6] D. Tezza and M. Andujar, "The State-of-the-Art of Human-Drone Interaction: A Survey," *IEEE Access*, vol. 7, pp. 167438–167454, 2019, doi: 10.1109/ACCESS.2019.2953900.

[7] J.-W. Lee, K.-H. Yu, J.-W. Lee, and K.-H. Yu, "Wearable Drone Controller: Machine Learning-Based Hand Gesture Recognition and Vibrotactile Feedback," *Sensors 2023, Vol. 23,* vol. 23, no. 5, Feb. 2023, doi: 10.3390/S23052666.

[8] K. Haratiannejadi and R. R. Selmic, "Smart Glove and Hand Gesture-Based Control Interface for Multi-Rotor Aerial Vehicles in a Multi-Subject Environment," *IEEE Access*, vol. 8, pp. 227667–227677, 2020, doi: 10.1109/ACCESS.2020.3045858.

[9] H. Kim and W. Chang, "Intuitive drone control using motion matching between a controller and a drone," *Archives of Design Research*, vol. 35, no. 1, pp. 93–112, 2022.

[10] A. Mecca and F. Scarcello, "through On-Board RGB Camera for UAV Flight Stabilization," pp. 1–20, 2023.

[11] M. Litochevski, "UART to Bus Core Specifications Written for publication on: File name: UART to Bus Core Specifications," 2010.

[12] A. Saraf, S. Moon, and A. Madotto, "A Survey of Datasets, Applications, and Models for IMU Sensor Signals," *ICASSPW 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing Workshops, Proceedings*, 2023, doi: 10.1109/ICASSPW59220.2023.10193365.

[13] A. Nayyar and V. Puri, "A review of Arduino board's, Lilypad's & Arduino shields," in *2016 3rd international conference on computing for sustainable global development (INDIACom)*, IEEE, 2016, pp. 1485–1492.

[14] H. R. Iskandar, S. Basuki, M. R. Hidayat, A. Daneraici Setiawan, D. Rukanda, and S. U. Prini, "Wireless telemetry for real-Time monitoring of photovoltaic application system using monopole antenna 3drobotics radio 915 mhz," *TSSA 2019 - 13th International Conference on Telecommunication Systems, Services, and Applications, Proceedings*, pp. 277–281, Oct. 2019, doi: 10.1109/TSSA48701.2019.8985518.

DOI: http://doi.org/10.26740/jistel.v2n1.p29-43
e- ISSN 3090-613X