Inform : Jurnal Ilmiah Bidang Teknologi Informasi dan Komunikasi
Vol.10 No.2 July 2025, P-ISSN : 2502-3470, E-ISSN : 2581-0367

Hybrid Multi-Servo Motor Controller Within an IoT-Enabled Smart Mechatronics Framework

Dodit Suprianto[1], Ginanjar Suwasono Adi[2], Rini Agustina[3], Nurul Hidayati[4], Azam Muzakhim Imammuddin[5]

[1,2,4]*Department of Electrical Engineering, Politeknik Negeri Malang, Indonesia*

[3]*Department of Information System, Universitas Kanjuruhan Malang, Indonesia*

[5]*Department of Mechanical Engineering, Politeknik Negeri Malang, Indonesia*

[2]ginanjar.adi@polinema.ac.id (*)

[1,4,5][dodit.suprianto, nurulhid8, azam]@polinema.ac.id

[3]riniagustina@unikama.ac.id

Received: 2025-04-22; Accepted: 2025-06-04; Published: 2025-07-04

Abstract—The increasing demand for precise motor control in industrial automation and IoT-integrated applications has driven the development of hybrid control systems for multi-servo motor management. Existing solutions often rely solely on either IoT-based automation or standalone manual control, limiting adaptability in environments with unreliable network connectivity. This study proposes a hybrid control system that integrates local potentiometer-based control with IoT-enabled remote operation to enhance flexibility and reliability. An experimental approach is employed to design and evaluate the hybrid control system, utilizing a modular controller board and MQTT as an IoT communication protocol. The system's performance is assessed based on response time and synchronization accuracy under varying network conditions. Experimental findings demonstrate that the proposed system effectively balances remote accessibility while ensuring on-site reliability. The integration of MQTT QoS level 2 enhances real-time performance by ensuring the accurate delivery of messages. Measured delays range from 21.72 ms to 55.61 ms, with jitter values between 1.17 ms and 33.89 ms, highlighting the impact of data traffic on control precision. By addressing latency, synchronization, and connectivity challenges, the proposed system bridges the gap between IoT-driven automation and manual control mechanisms, providing a scalable and reliable solution for broader automation applications.

Keywords—Hybrid Control; Multi-Servo Motor; IoT; MQTT; Potentiometer-Based Control; Real-Time Performance.

I. INTRODUCTION

The rapid advancement of smart mechatronics has driven an increasing demand for precise, efficient, and adaptable motor control systems, particularly in industrial automation, robotics, and IoT-integrated applications [1]–[3]. Multi-servo motor control plays a critical role in these fields, facilitating synchronized motion, high-precision positioning, and adaptive automation [4]–[6]. However, traditional control architectures often struggle with scalability, communication latency, and fault tolerance, limiting their effectiveness in real-world industrial environments [7]–[9].

Recent research has focused on enhancing dynamic response, minimizing torque ripple, and optimizing communication protocols to achieve real-time performance in servo motor systems [10]–[12]. The integration of IoT with motor control has further expanded capabilities, enabling real-time monitoring, predictive maintenance, and cloud-based adaptive automation [13]–[15]. Despite these advancements, ensuring seamless synchronization, low-latency operation, and high-precision performance in networked multi-servo motor systems remains a significant challenge, particularly in scenarios with unstable network connectivity [16]–[18].

To address these challenges, various approaches have been explored. FPGA-based multi-axis controllers offer high-speed processing but are hindered by high implementation complexity and limited adaptability [19][20]. PID tuning techniques offer precise control, yet they require frequent manual recalibration and struggle with dynamic load variations [21][22]. MQTT-based remote motor control has demonstrated effectiveness in IoT-driven automation but remains highly dependent on stable network conditions, often lacking a robust local fallback mechanism [23][24]. These limitations emphasize the need for a hybrid control approach that can seamlessly integrate IoT-based remote management with local manual control, ensuring adaptive, low-latency, and fault-tolerant operation in diverse industrial settings [25][26].

Traditional manual and remote-control systems also exhibit inherent constraints, primarily due to either over-reliance on network connectivity or limited adaptability in real-time applications [27]. Many existing solutions operate exclusively on either IoT automation or standalone, potentiometer-based manual control, which restricts their effectiveness in dynamic industrial environments [28][29]. Multi-axis servo motor systems continue to struggle with synchronization delays and response inconsistencies, which are detrimental to the precise management of multiple actuators [30]. There is a clear need for developing an integrated hybrid control framework that ensures real-time synchronization alongside low-latency, responsive operations, and dependable performance under changing conditions [31]. Different hybrid control approaches have been introduced, which feature FPGA-powered controllers along with PID-based precise actuation and IoT-enabled monitoring systems [32]. Existing methods do not

DOI : https://doi.org/10.25139/inform.v10i2.10100

provide a single framework that simultaneously enables remote accessibility and reliable local manual operation without compromising efficiency, adaptability, and robustness [33][34].

The study introduces a hybrid control system for multi-servo motors that integrates traditional automation methods with Internet of Things (IoT) capabilities. The system design features both a multi-potentiometer local control interface and MQTT-based IoT remote operation, which benefit from real-time synchronization capabilities. The experimental study focuses on evaluating system performance through multiple metrics. It evaluates system performance using response time metrics alongside synchronization accuracy and system stability in diverse network conditions. The analysis includes a detailed examination of latency in MQTT-based communications and input responsiveness from multi-channel potentiometers to achieve real-time precision and system adaptability.

II. Research Methodology

This study employs an experimental approach to develop and evaluate a prototype of a hybrid control system for managing multi-axis servo motors. The system integrates two primary control methods: local control using multi-channel potentiometers and Internet of Things (IoT) based remote control. These methods are specifically designed to address typical issues in servo motor control, with a focus on response time and reliability. The proposed hybrid control system architecture of this study is illustrated in Fig.1.

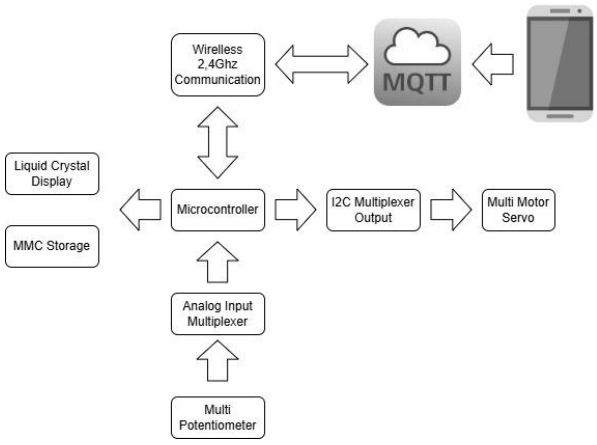

Fig.1. System Design of Proposed IoT Architecture.

The key IoT communication protocol mechanism utilized in this research is Message Queuing Telemetry Transport (MQTT). The MQTT broker is hosted on a cloud server, enabling commands to be sent from a client application to the servo controller board via an internet communication medium. This system guarantees low latency and efficient data sharing for an end-to-end real-time control application. As part of the local controller, a multi-channel potentiometer-based control device permits manual adjustment of the rotation angles of the multi-servo motors. This configuration function is independent of network connectivity, providing a reliability control mechanism in the event of potential disruption.

Additionally, the prototype features six servo motor units, each connected to the I2C multiplexer output, providing six degrees of freedom. These motors are controlled by a microcontroller-based control circuit, which serves as the sensor feedback manager, providing precise and synchronized control. The hybrid system architecture delivers a seamless balance between IoT-based flexibility and local control stability. The combination of these technological improvements can enhance the system's precision and adaptability, making it well-suited for applications that require high reliability and real-time responsiveness.

A. Design of Servo Controller System Board

The schematic design of the servo controller board proposed in this study, which emphasizes a modular and efficient system, is shown in Fig. 2. The ESP32 is selected as the central control unit in our system due to its notable processing abilities and numerous communication interfaces, including I²C, SPI, and UART. The ESP32 is also responsible for managing inter-component communication and processing input signals operating on this board. The system begins with a stabilized power supply, utilizing the 5V power output from the PSU as the VCC input to power the servo controller board. Then, this VCC socket distributes power to the ESP32 board, the PCA9685 PWM module, and servo motors, activating these modules. To ensure the stable operation of our servo controller board, we utilize a noise-filtering bridge to effectively minimize voltage fluctuations. A CD4067 16-channel multiplexer enhances the input capability of the ESP32's limited analog input. This module enables the reading of numerous signals with unparalleled efficiency, utilizing four selection lines to optimize GPIO capacity.

The system integrates six servo motors, each running on a PCA9685 PWM module that communicates using the I²C protocol. The ESP32 uses SCL and SDA to communicate with the PCA9685, receiving commands that are converted into PWM pulses for the servos. Every PWM output corresponds to a servo's signal pin, allowing it to be adjusted to the desired angle. Additionally, an SD card module connected via SPI can store control parameters when the device is turned off, which is crucial for any automated system. A system user supplies inputs using potentiometers and switches for a more engaging approach. An LCD connected via I²C, which is shared with the PCA9685, allows for immediate visualization of the potentiometer values, servo positions, and the overall system status. To avoid floating input states, logic levels are raised using pull-up resistors to enable accurate digital readings.

With the combination of stable power management, efficient I²C and SPI communication, and accurate PWM control, it fully utilizes microcontroller assets while allowing for future development. Its lightweight and cost-efficient structure enables the development of a prototype hybrid control system. At the same time, multi-axis servo motor control makes it suitable for robotics and automation systems that require distributed intelligent control and high-speed responsiveness.

DOI : https://doi.org/10.25139/inform.v10i2.10100

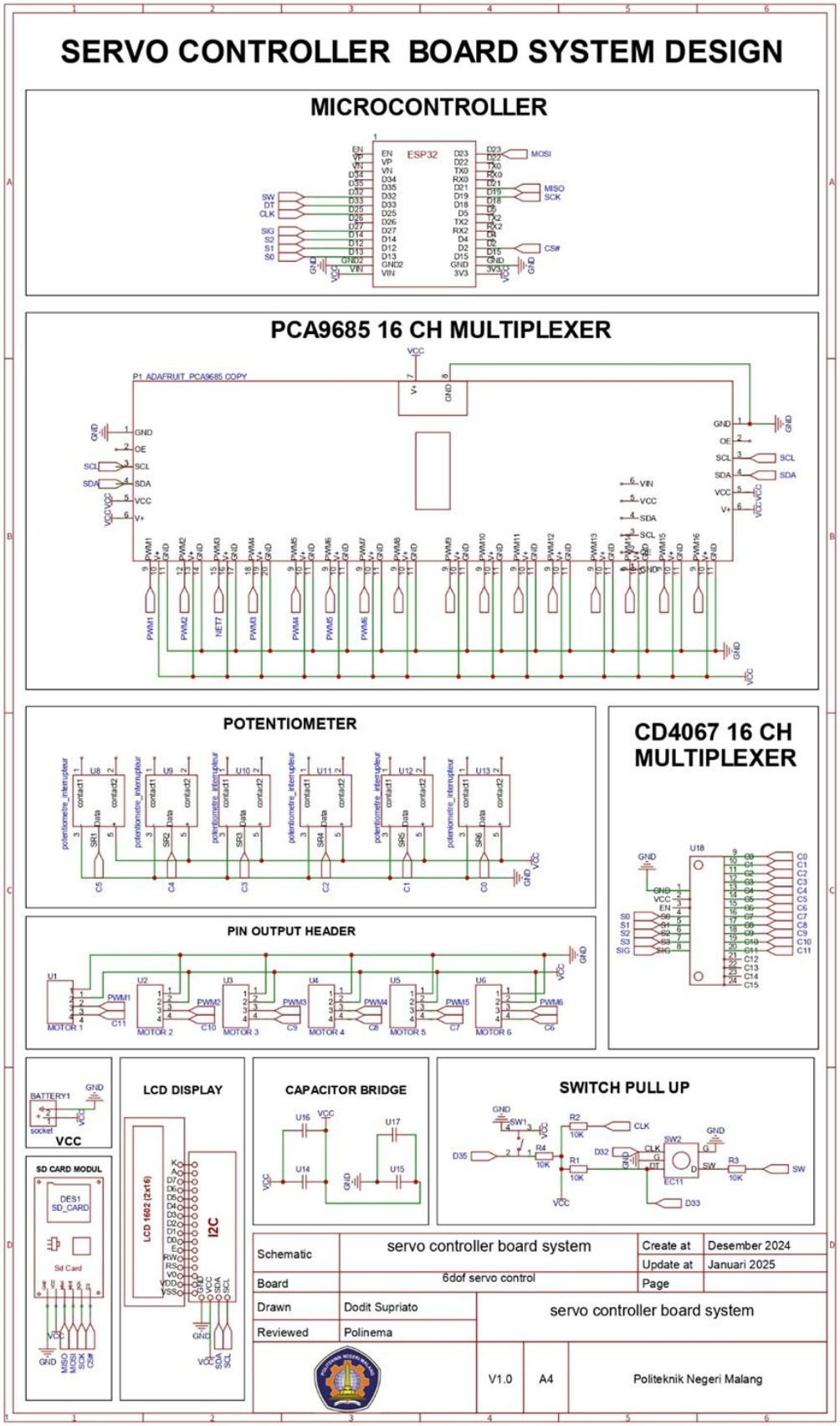

Fig.2. Design of Servo Controller System Board

DOI : https://doi.org/10.25139/inform.v10i2.10100

B. *Message Queuing Telemetry Transport (MQTT)*

The network architecture implemented in the system, which consists of a Servo controller board, an MQTT Broker, and an IoT Servo Controller application running on an Android device illustrated in Fig.3. The Servo controller board is responsible for managing six servo motors, interfacing with the MQTT broker via an Access Point Router (AP), and receiving or publishing control commands. Communication is facilitated using the MQTT protocol, ensuring real-time servo control through the use of subscriber and publisher topics. The LCD module displays servo positions and control parameters, while potentiometers enable local manual adjustments.

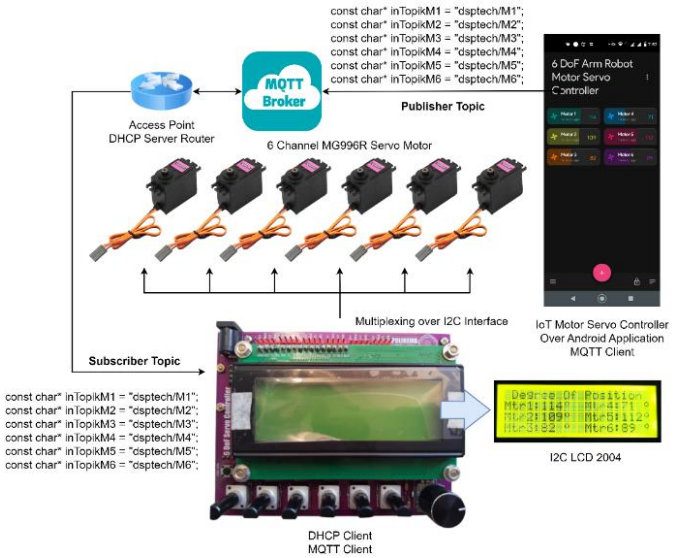

Fig.3. Proposed Design of Hybrid Control System Architecture

The ESP32 microcontroller serves as the system's core, handling communication over I²C, SPI, and Wi-Fi. During initialization, the Servo controller board connects to the MQTT Broker using predefined credentials and establishes the Servo control topics. If the connection is lost, the system automatically attempts to reconnect every five seconds, ensuring a robust and reliable remote-control mechanism.

As part of the local servo control mode, users can manually adjust the servo positions using six potentiometers, which are multiplexed using CD4067 ICs. The ESP32 reads these analog signals, processes them, and sends PWM commands via a PCA9685 module for servo actuation. Then, current servo positions are displayed on the I²C LCD, providing real-time position feedback to the user.

In IoT servo control mode, servo commands are received from the MQTT Broker and processed by the ESP32. Each servo motor is assigned a unique MQTT topic (e.g., dsptech/M1 to dsptech/M6), enabling independent control. The MQTT Client Android application allows users to send position commands via sliders, setting angles between 0° and 180°. Messages are sent through the broker, ensuring precise servo actuation.

Additional system features include menu navigation and Wi-Fi reset capabilities. The rotary encoder enables users to navigate through menu options displayed on the LCD screen, allowing them to access functions such as "Control: IoT," "Motor Degree Ctrl," "Record Movement," and "Play Movement". If network settings need to be changed, the Wi-Fi Reset feature can be triggered by pressing a button, prompting the ESP32 to restart in Access Point mode (SSID: AutoConnectAP).

C. *Quality of Service (QoS)*

In MQTT, QoS parameters play a vital role in guaranteeing reliable message delivery for IoT applications. This study focuses on the key aspects of MQTT Quality of Service (QoS) Level 2 operation and QoS traffic parameters, including end-to-end delays and jitter.

1) MQTT QoS Level 2: To ensure reliable and real-time communication in a multi-axis servo motor control system, MQTT QoS 2 is employed in this research as the primary communication protocol. QoS 2 ensures that each control message is delivered "exactly once", preventing duplication and message loss, which is critical for precise motor coordination. While this level of reliability introduces additional overhead due to its four-step handshake mechanism, it is necessary for applications where accurate and consistent message delivery is essential.

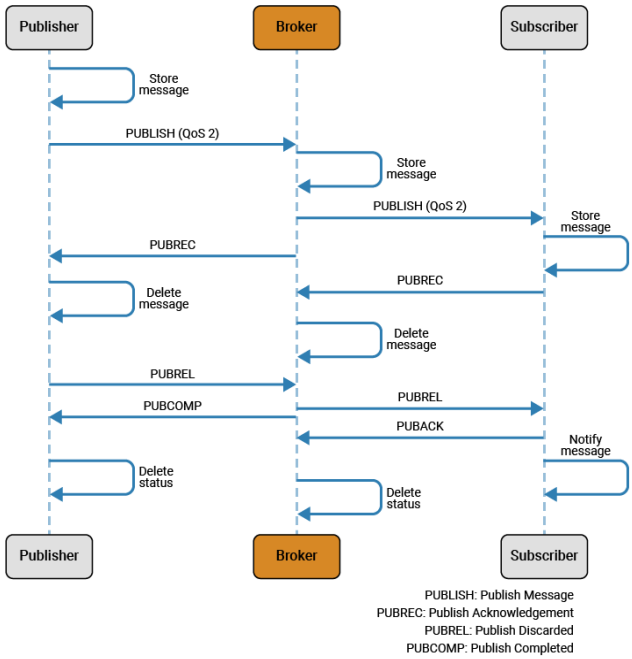

Fig.4. MQTT QoS Level 2 Architecture

MQTT QoS 2 establishes a structured communication protocol to ensure message integrity, as illustrated in Fig. 4 [35]. When a control message is provided, the receiver processes it and returns a Publish Received (PUBREC) acknowledgment. If the sender does not get the PUBREC packet, it will resend the PUBLISH packet until the acknowledgment is confirmed. After PUBREC is received, the sender discards the original message and sends a Publish

DOI : https://doi.org/10.25139/inform.v10i2.10100

Released (PUBREL) packet. The receiver then examines the message and responds with a Publish Complete (PUBCOMP) packet, allowing both the sender and receiver to erase stored states while reusing the packet identity. This approach enables precise message synchronization in multi-axis motor control. Using MQTT QoS 2, the hybrid control system aims to improve multi-axis servo motor coordination by ensuring precise and consistent message delivery.

2) Delay: The delay parameter is particularly critical for real-time applications, such as those involving multi-axis servo motors, where rapid communication is essential for precise and swift command execution. Reducing delay can enhance the responsiveness of the control system, which is crucial for maintaining stability and operational efficiency in dynamic conditions. Equation (1) explains the computation for delay in this experimental study.

$$Delay = Rx\ Timestamp\ - Tx\ Timestamp \quad (1)$$

3) Jitter: Jitter, on the other hand, signifies the variation in message delivery timings. In servo motor control, unpredictable communication delays can lead to inconsistent motor responses, negatively impacting precision and stability. Minimizing jitter is crucial for guaranteeing the system's capability to handle various tasks while maintaining smooth and precise performance. This measure is considered the deviation of each message's delay, which can be represented as Equation (2). Where, the Di variable is the delay of the i-the packet and Di-n for the previous delay of i-the packet. By reducing jitter, the proposed framework ensures consistent performance, enhancing its adaptability to varying operational demands.

$$Jitter = |D_i - D_{i-n}| \quad (2)$$

III. Result and Discussion

A. Servo Controller System Board

We have designed and developed a customized servo controller board system to meet the specific requirements of this study. The Servo controller board system consists of several components that work together to manage servo motor operations. Fig.5 provides detailed information on the servo motor output header pins. (A) The ESP32 Microcontroller serves as the actuator controller, handling data processing and internet communication. (B) A 4-row, 20-column LCD functions as a monitor screen to display the application menu and changes in the rotation angles of each servo motor. This LCD communicates with the ESP32 via the I2C interface. (C) Six potentiometers enable independent control of the angle for each servo motor, spanning a range of 0° to 180°. The potentiometers operate in analog mode with a PWM precision level of 212, corresponding to a value range of 0–4096, and are active only in 'Local' mode, being disabled in 'IoT' mode.

The male header (D) for servo motor output provides connections for the servo motors, including ground and VCC pins, a PWM pin for receiving input signals from either the potentiometer or MQTT data, and an FB pin for future development to monitor the motor's rotation angle. (E) A rotary encoder enables navigation through the LCD menu to select the system mode ('Local' or 'IoT') and display rotation angle values. (F) To address the limited number of input pins on the ESP32, an analog signal multiplexer module (IC CD4067) is used to interface with potentiometers. Similarly, (G) is an I2C interface multiplexer module (IC PCA9685) that serves as an output multiplexer for controlling the rotation angles of six servo motors using a single I2C address, thereby simplifying programming.

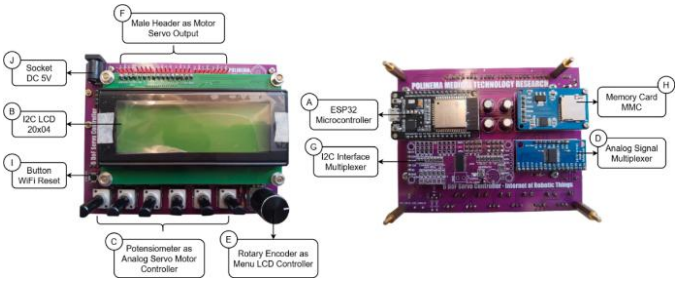

Fig.5. Servo controller board System Design

A memory card module (H) records the rotation angles of the servo motors during operation. Changes in rotation angles are automatically recorded as each motor is physically adjusted. (I) A momentary button switch for Wi-Fi reset is available for selecting a new SSID access point for the ESP32, though this feature is only used when connecting to a different access point. (J) Finally, the system is powered via a 5-volt DC socket, which connects to an external power supply.

B. System Testing Scenario

The testing scenario for the proposed system is conducted based on the network architecture shown in Fig. 6. The public MQTT broker is hosted in the cloud, using port 1883. This broker acts as the central communication hub, managing data exchange between MQTT clients. Meanwhile, the laptop running the Wireshark application and the Servo controller board are connected to the same network with the subnet 192.168.0.1/24. The laptop, with the IP address 192.168.0.2, functions as an MQTT client and data packet traffic monitor. The servo controller board functions as an MQTT client with an IP address of 192.168.0.3. This module is responsible for communicating with a motor servo to receive and execute rotation commands. Both devices are connected via an Access Point Router to access the internet.

In this test scenario, the Wireshark application monitors data packet traffic [36], specifically focusing on the communication port between the cloud-based MQTT broker and the servo controller board. Moreover, the MQTT Dashboard Android application, which controls the rotation angle of the servo motor, operates on a separate network with the subnet 192.168.100.1/24 and connects to the internet via a different network. This isolation allows the system to simulate cross-network communication and identify any delays or congested packets that may occur in the internet network.

The evaluation procedure for the servo controller board system involves monitoring the responsiveness of the QoS 2

DOI : https://doi.org/10.25139/inform.v10i2.10100

mechanism process, as well as latency and jitter. The test initiates at Point A, where the MQTT Dashboard Android application sends a command to the MQTT broker. Then, the broker forwards designated topics and payloads to the servo controller board at Point B via the Access Point Router. The board then processes the command and delivers data back to the laptop for examination, which is observed at Point C using the Wireshark application.

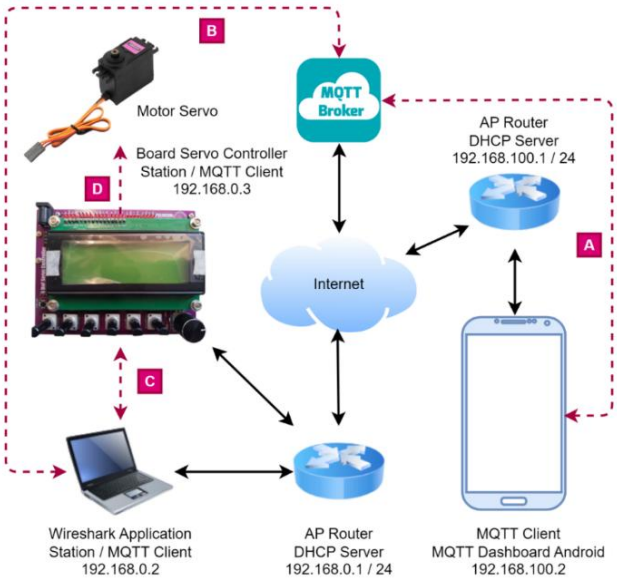

Fig.6. Testing Scenario for QoS Parameters

Overall, the test aims to evaluate the time taken from sending data through the MQTT Dashboard Android application to when the command reaches the servo motor, the motor responds, and the movement starts. By monitoring data traffic at critical points, the test provides insights into the system's performance, ensuring the QoS 2 mechanism functions correctly while specifically analyzing delay and jitter.

C. *QoS Test Results*

1) Result of MQTT QoS Level 2: The log provided in Fig.7 captures the interaction between an MQTT client, identified as "Motor Servo 1," and an MQTT broker hosted at broker.hivemq.com on port 1883. The client uses the mosquitto_pub.exe tool to publish a message with a payload of "20" to the topic of "dsptech/M1" with a Quality of Service (QoS) level of 2. The process begins with the client sending a CONNECT packet to the broker, which responds with a CONNACK packet, indicating a successful connection. The client then sends a PUBLISH packet using QoS 2, confirming reliable message delivery through a four-step handshake. This handshake mechanism comprises the client receiving a PUBREC (Publish Received) acknowledgment from the broker, followed by the transmission of a PUBREL (Publish Release) packet, and ultimately receiving a PUBCOMP (Publish Complete) packet, verifying that the message has been properly handled. At that point, the client ends the session by sending a DISCONNECT packet to the broker.

```
C:\>"C:\Program Files\mosquitto\mosquitto_pub.exe" -d -h broker.
hivemq.com -p 1883 -t "dsptech/M1" -q 2 -m 20 --id "Motor Servo
1"
Client Motor Servo 1 sending CONNECT
Client Motor Servo 1 received CONNACK (0)
Client Motor Servo 1 sending PUBLISH (d0, q2, r0, m1, 'dsptech/M
1', ... (2 bytes))
Client Motor Servo 1 received PUBREC (Mid: 1)
Client Motor Servo 1 sending PUBREL (m1)
Client Motor Servo 1 received PUBCOMP (Mid: 1, RC:0)
Client Motor Servo 1 sending DISCONNECT
```

Fig.7. Log of MQTT QoS 2 Mechanism Process

On the subscriber side, particularly in microcontroller-based devices, monitoring the whole QoS 2 handshake can be challenging due to the lack of specialized debugging tools or libraries that provide extensive insights into the MQTT protocol's internal workings. Microcontrollers typically rely on lightweight MQTT libraries that prioritize resource efficiency over systematic logging. As a result, while the subscriber may receive the message, it may not be feasible to monitor intermediary processes, such as PUBREC, PUBREL, or PUBCOMP, which are vital for confirming the message's delivery under QoS 2. This constraint can make debugging and validating end-to-end message delivery more challenging in resource-constrained applications.

2) Delay and Jitter Measurements: This subsection provides an overview of the system's performance in terms of QoS connection time, from the application side to the hardware and vice versa. The graph in Fig. 8 and the data in Table 1 show the correlation between total delay and jitter across various payload angles, providing a comprehensive evaluation of system performance. The experiment was conducted by sending nine distinct variations of rotation angles, ranging from 20° to 180° in increments of 20°. The delay and jitter measurements for QoS level 2 on the MQTT protocol reveal considerable changes based on the payload size. The total delay, which is the sum of the delay from the MQTT Dashboard Android application to the MQTT Broker and from the Broker to the Servo controller board, varies across different payload degrees. The maximum overall delay (55.61 ms) occurs at 160°, while the lowest (21.72 ms) is observed at 180°. Jitter, which represents delay variation, is similarly inconsistent, with the highest value (33.89 ms) measured at 180° and the lowest (1.17 ms) at 120°. These results suggest that QoS level 2 in MQTT communication exhibits varying levels of stability and responsiveness, depending on data traffic conditions and payload size.

A relationship between data traffic and QoS performance is evident in the delay and jitter information. The delay from the MQTT Dashboard to the Broker appears to be the dominant factor influencing total delay, with some values exceeding 40 ms, such as at 60° (45.90 ms) and 160° (49.12 ms). In contrast, the delay from the Broker to the Servo Controller remains relatively stable, ranging from 5.82 ms to 6.50 ms. The inconsistencies in jitter, particularly at 20°, 80°, 160°, and 180°, indicate fluctuations in network performance and

DOI : https://doi.org/10.25139/inform.v10i2.10100

potential congestion, which may affect the precision of the hybrid control system for multi-axis servo motor management.

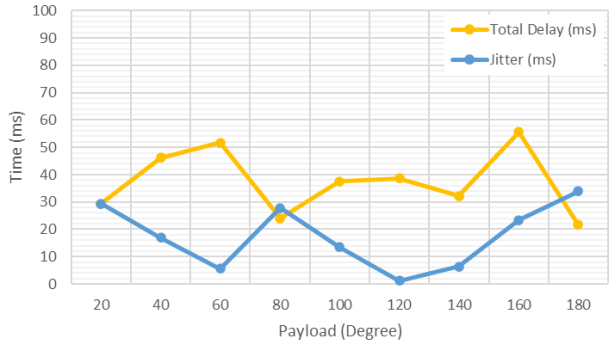

Fig.8. QoS Parameters for Delay and Jitter vs Payload (Degree)

TABLE I
CORRELATION BETWEEN TOTAL DELAY AND JITTER ACROSS VARIED PAYLOAD ANGLES

Payload (Degree in String)	Data Package (byte)	Delay from Broker MQTT to Board Servo Controller (ms)	Delay from MQTT Dashboard Android to Broker MQTT (ms)	Total Delay (ms)	Jitter (ms)
20	2	5,88	23,47	29,35	29,35
40	2	5,83	40,37	46,20	16,85
60	2	5,82	45,90	51,72	5,52
80	2	5,83	18,18	24,01	27,72
100	3	5,83	31,69	37,52	13,51
120	3	6,50	32,19	38,69	1,17
140	3	6,50	25,74	32,23	6,45
160	3	6,50	49,12	55,61	23,38
180	3	6,49	15,23	21,72	33,89
			Average:	37,45	39,66

Overall, the data suggests that while MQTT's QoS level 2 ensures message delivery reliability, it does not guarantee low-latency performance under all conditions. The variations in delay and jitter highlight the need for optimized network configurations or additional buffering mechanisms to ensure smoother real-time control. With an average total delay of 39.66 ms and an average jitter of 37.45 ms, future improvements may focus on mitigating fluctuations, particularly during high data traffic scenarios, to enhance the efficiency of the prototype hybrid control system.

In addition, the observed delays (21.72–55.61 ms) under hybrid operation represent a significant improvement over pure IoT-based systems in unstable networks, where prior studies report >100 ms delays during packet loss [23], [24]. Unlike standalone IoT solutions that fail during network outages, our hybrid system maintains functionality via local control, eliminating downtime. While pure potentiometer-based control offers negligible latency (<1 ms) [28], it lacks remote programmability. Conversely, IoT-only systems enable remote access but suffer from unreliability under poor connectivity. Our hybrid approach bridges these extremes, adding <6.5 ms overhead versus manual mode while ensuring fallback operability, which is a critical advantage for industrial settings.

IV. CONCLUSION

This study presents a hybrid control system for multi-axis servo motor management, combining IoT-based remote control via MQTT with local potentiometer-based control to ensure operational continuity. The system effectively balances flexibility and reliability, allowing seamless transitions between remote and local modes. Experimental validation demonstrates the efficient management of six servo motors using an ESP32 microcontroller, an I2C multiplexer, and a PWM controller, enabling precise and synchronized motor actuation. The findings indicate that while QoS level 2 introduces additional overhead, it significantly improves message reliability, which is crucial for multi-axis coordination. Performance evaluation reveals delay variations that depend on payload size and traffic conditions, with the highest observed delay at 55.61 ms and the lowest at 21.72 ms. These results emphasize the importance of QoS level 2 in maintaining system stability.

Although a direct comparison with standalone systems was not the primary focus of this study, the observed delays were significantly lower than those typically reported for IoT-only systems under identical network conditions. Moreover, the hybrid model's ability to fall back on local control prevents total system failure during outages—a critical limitation of purely IoT-based architectures**.** However, testing for simultaneous motor control was not conducted, as this study did not focus on developing user-friendly Human-Computer Interaction (HCI) technologies such as joysticks or camera-based virtual skeletons. On the other hand, scalability beyond six servos is architecturally feasible through PCA9685 cascading, which supports up to 16 channels integrated with distributed ESP32 nodes. However, empirical validation of these configurations remains a work for the future. Subsequent research will aim to quantify the trade-offs in larger deployments. Overall, the proposed hybrid control system enhances real-time motor control by combining IoT communication with a fail-safe local mechanism, providing a reliable and scalable solution for intelligent automation applications that require high precision and adaptability.

ACKNOWLEDGMENT

This work was supported in part by the Center for Research and Community Service Politeknik Negeri Malang under Grant: SP DIPA-023.18.2.677606/2024.

REFERENCES

[1] M. Ryalat, E. Franco, H. Elmoaqet, N. Almtireen, and G. Al-Refai, "The integration of advanced mechatronic systems into industry 4.0 for smart manufacturing," Sustain., vol. 16, no. 19, pp. 1–39, 2024, doi: 10.3390/su16198504.

[2] H. Chen, "The practice of mechatronics technology in intelligent manufacturing," in Proc.SPIE, Mar. 2024, vol. 12981, p. 1298141. doi: 10.1117/12.3014884.

[3] A. Pchelintsev and N. Malev, "INTELLIGENT control in mechatronic systems," Ekon. I Upr. Probl. RESHENIYA, 2024, doi: 10.36871/ek.up.p.r.2024.07.05.005.

[4] A. Aliyev and A. Yalchinkaya, "Modern prospects for the application of mechatronics and robotics capabilities," J. Ecohumanism, vol. 3, no. 7, pp. 765–769, 2024, doi: 10.62754/joe.v3i7.4245.

DOI : https://doi.org/10.25139/inform.v10i2.10100

[5] A. Filipescu, I. Stamatescu, G. Simion, D. Ionescu, and A. Filipescu, "IoT-cloud based control of a flexible assembly/disassembly mechatronic system in the framework of industries 4.0 and 5.0," IEEE Int. Conf. Control Autom. ICCA, no. 47, pp. 534–539, 2024, doi: 10.1109/ICCA62789.2024.10591866.

[6] J. Zhao and B. Huang, "Research on the integration of intelligent technology and mechatronics in mechanical manufacturing," Front. Sci. Eng., vol. 2, no. 9, pp. 29–32, 2022, doi: 10.54691/fse.v2i9.2230.

[7] J. Huanca, J. Zamora, J. Cornejo, and R. Palomares, "Mechatronic design and kinematic analysis of 8 dof serial robot manipulator to perform electrostatic spray painting process on electrical panels," Proc. 2022 IEEE Eng. Int. Res. Conf. EIRCON 2022, pp. 1–4, 2022, doi: 10.1109/EIRCON56026.2022.9934104.

[8] M. Ntuen, O. Osanaiye, I. Adeosun, I. B. Muhammad, I. Lawrence, and B. M. Sambo, "Using mechatronics to facilitate manufacturing processes," 2023 2nd Int. Conf. Multidiscip. Eng. Appl. Sci. ICMEAS 2023, vol. 1, pp. 1–3, 2023, doi: 10.1109/ICMEAS58693.2023.10379366.

[9] J. Cornejo et al., "Industrial, collaborative and mobile robotics in latin america: review of mechatronic technologies for advanced automation," Emerg. Sci. J., vol. 7, no. 4, pp. 1430–1458, 2023, doi: 10.28991/ESJ-2023-07-04-025.

[10] W. Ding, J. Li, and J. Yuan, "An improved model predictive torque control for switched reluctance motors with candidate voltage vectors optimization," IEEE Trans. Ind. Electron., vol. 70, no. 5, pp. 4595–4607, 2022, doi: 10.1109/TIE.2022.3190895.

[11] S. Lee, C. Hwang, J. Shim, and J. I. Ha, "A control method of servo motor drives for fast dynamic response and low torque ripple," 2022 IEEE Energy Convers. Congr. Expo. ECCE 2022, pp. 1–5, 2022, doi: 10.1109/ECCE50734.2022.9947638.

[12] A. O. Deab, K. Karthikumar, M. Karuppiah, and P. A. G. Sankar, "A new optimal space vector modulation with dtc switching strategy for induction motor control," Int. J. Appl. Power Eng., vol. 13, no. 4, pp. 862–873, 2024, doi: 10.11591/ijape.v13.i4.pp862-873.

[13] T. T. Nguyen, T. H. Nguyen, and J. W. Jeon, "Explicit model predictive speed control for permanent magnet synchronous motor with torque ripple minimization," IEEE Access, vol. 11, no. October, pp. 134199–134210, 2023, doi: 10.1109/ACCESS.2023.3335992.

[14] X. Sun, Y. Zhu, Y. Cai, M. Yao, Y. Sun, and G. Lei, "Optimized-sector-based model predictive torque control with sliding mode controller for switched reluctance motor," IEEE Trans. Energy Convers., vol. 39, no. 1, pp. 379–388, 2024, doi: 10.1109/TEC.2023.3301000.

[15] M. Ghibeche, K. Kouzi, D. Difi, and A. Ouanouki, "Investigation of predictive direct torque control of double star permanent magnet synchronous machine (dspmsm)," Stud. Eng. Exact Sci., vol. 5, no. 1, pp. 2672–2684, 2024, doi: 10.54021/seesv5n1-132.

[16] H. Lin et al., "Three-stage duty cycle-based deadbeat predictive torque control for three-phase spmsms with cmv reduction," IEEE Trans. Power Electron., vol. 38, no. 9, pp. 11385–11398, 2023, doi: 10.1109/TPEL.2023.3288184.

[17] C. Hui, H. Yanjie, P. Yinbin, and Z. Tao, "A method for reducing torque ripple of switched reluctance motor based on partitioned tsf," Front. Energy Res., vol. 12, no. May, pp. 1–11, 2024, doi: 10.3389/fenrg.2024.1381950.

[18] J. Zhou, M. Cheng, H. Wen, X. Yan, M. Tong, and W. Wang, "Modeling and suppression of torque ripple in pmsm based on the general airgap field modulation theory," IEEE Trans. Power Electron., vol. 37, no. 10, pp. 12502–12512, 2022, doi: 10.1109/TPEL.2022.3174395.

[19] V. Q. B. Ngo, N. Kim Anh, and N. Khanh Quang, "FPGA-based adaptive pid controller using mlp neural network for tracking motion systems," IEEE Access, vol. 12, no. May, pp. 91568–91574, 2024, doi: 10.1109/ACCESS.2024.3422015.

[20] Y. F. Li, Z. W. Xu, Y. J. Zhang, Y. Bin Wu, and L. Chuang, "On the design of simultaneous motion controller for the four-axis scara system," Proc. Int. Conf. Power Electron. Drive Syst., vol. 2023-Augus, no. August, pp. 1–6, 2023, doi: 10.1109/PEDS57185.2023.10246556.

[21] S. Goyal, A. Shankar, K. C. Das, A. Singh, S. Oli, and M. M. Sati, "Manual and adaptive tuned pid controllers for industrial application," 2024 4th Int. Conf. Adv. Comput. Innov. Technol. Eng. ICACITE 2024, pp. 1953–1958, 2024, doi: 10.1109/ICACITE60783.2024.10616874.

[22] A. K. Maurya, M. Dutt Dwivedi, and H. Khan, "Implementation of pid controller tuning method for load frequency control in multi area power system," 4th Int. Conf. Innov. Pract. Technol. Manag. 2024, ICIPTM 2024, no. Iciptm, pp. 1–6, 2024, doi: 10.1109/ICIPTM59628.2024.10563927.

[23] C. Lertyosbordin, D. Wongsanont, N. Khurukitwanit, and W. Saowapark, "A framework for remote robot actuation using ros integrated with mqtt," 2024 Int. Tech. Conf. Circuits/Systems, Comput. Commun. ITC-CSCC 2024, pp. 1–4, 2024, doi: 10.1109/ITC-CSCC62988.2024.10628235.

[24] S. Opacin, L. Rizvanovic, B. Leander, S. Mubeen, and A. Caudevic, "Developing and evaluating mqtt connectivity for an industrial controller," 12th Mediterr. Conf. Embed. Comput. MECO 2023, pp. 6–10, 2023, doi: 10.1109/MECO58584.2023.10154921.

[25] J. Xin et al., "Design and implementation of an efficient hardware coprocessor ip core for multi-axis servo control based on universal soc," Electron., vol. 12, no. 2, 2023, doi: 10.3390/electronics12020452.

[26] M. Cristian Marin, M. Cerutti, S. Batista, and M. Brambilla, "A multi-protocol iot platform for enhanced interoperability and standardization in smart home," Proc. - IEEE Consum. Commun. Netw. Conf. CCNC, pp. 1–6, 2024, doi: 10.1109/CCNC51664.2024.10454663.

[27] H. Yu, "Method for remote-control of electrical automation instruments over wireless networks based on plc technology," IEEE 1st Int. Conf. Ambient Intell. Knowl. Informatics Ind. Electron. AIKIIE 2023, pp. 1–6, 2023, doi: 10.1109/AIKIIE60097.2023.10390481.

[28] A. H. Embong, L. Asbollah, and S. B. A. Hamid, "JOURNAL of mechanical engineering and sciences empowering industrial automation labs with iot: a case study on real-time monitoring and control of induction motors using siemens plc and node-red," vol. 18, no. 2, pp. 10004–10016, 2024.

[29] H. Zhang, I. Automation, P. Optimization, and S. Analysis, "Design and implementation of industrial automation control system based on plc," J. Electron. Inf. Sci., vol. 9, no. 2, pp. 120–126, 2024, doi: 10.23977/jeis.2024.090215.

[30] I. Ghinea and G. Bucur, "Advanced automatic system for remote control in the oil and gas industry," Rom. J. Pet. Gas Technol., vol. 4 (75), no. 2, pp. 181–192, 2023, doi: 10.51865/jpgt.2023.02.18.

[31] N. H. Damia Mohamad Razam and S. Sara Rais, "IoT-enabled automatic plant watering system," 8th Int. Conf. Recent Adv. Innov. Eng. Empower. Comput. Anal. Eng. Through Digit. Innov. ICRAIE 2023, vol. 2023, pp. 1–6, 2023, doi: 10.1109/ICRAIE59459.2023.10468389.

[32] K. Singh, A. Jain, and P. Thiyagarajan, "Design and implementation of integrated control system for iot enabled home automation," Proc. - 2022 4th Int. Conf. Adv. Comput. Commun. Control Networking, ICAC3N 2022, pp. 1433–1437, 2022, doi: 10.1109/ICAC3N56670.2022.10074496.

[33] D. M. Murali and R. L. Rao, "Cost efficient five way home automation with feedback system," Interantional J. Sci. Res. Eng. Manag., vol. 07, no. 03, pp. 1–4, 2023, doi: 10.55041/ijsrem18075.

[34] X. Li, "Design of equipment automation control and remote supervision system based on artificial intelligence algorithm," in 2023 Int. Conf. Power, Electr. Eng. Electron. Control, 2023, pp. 515–520. doi: 10.1109/PEEEC60561.2023.00106.

[35] R. Doshi, "Distributed mqtt broker : a load-balanced redis-based architecture," 2024 Int. Conf. Emerg. Smart Comput. Informatics, pp. 1–6, 2024, doi: 10.1109/ESCI59607.2024.10497427.

[36] S. R. Hashim, R. A. Enad, A. M. Al-Khafagi, and N. K. Abdalhameed, "The facilities of detection by using a tool of wireshark," Indones. J. Electr. Eng. Comput. Sci., vol. 31, no. 1, pp. 329–336, 2023, doi: 10.11591/ijeecs.v31.i1.pp329-336.

DOI : https://doi.org/10.25139/inform.v10i2.10100