SERBIAN JOURNAL OF ELECTRICAL ENGINEERING
Vol. 23, No. 1, February 2026, 21-55

UDC: 004.78:004.35]: 007.52 **DOI: https://doi.org/10.2298/SJEE2601021S**

Original scientific paper

Development of a Comprehensive Approach for Precise Positioning and Orientation of Multiple Mobile Robots in a Specified Formation Using Computer Vision and the Internet of Things

Sneh Soni[1], Ansh Ruparel[1], Shreedhar Shah[1],
Harsh Kapadia[1], Himanshu Patel[1]

Abstract: The integration of Internet of Things (IoT) technologies into Multi-Robot Systems (MRS) has marked a significant advancement in the field of robotics and has opened up new avenues for innovation in robotics applications. By leveraging IoT technologies, robots in an MRS can be interconnected over a network, facilitating seamless communication and data exchange which can be used in various industrial as well as commercial applications. As manufacturing processes evolve towards increased automation and connectivity, MRS provides a versatile solution for tasks such as logistics, transportation, and collaborative assembly. This work presents a complete MRS architecture that combines Message Queue Telemetry Transport (MQTT) based wireless communication, overhead camera–assisted pose estimation using ArUco markers, a graphical control interface, and onboard sensing to achieve accurate multi-robot formation control. Unlike high-cost optical motion-capture systems like Vicon or infrastructure-dependent Ultra-Wide Band (UWB) localization, the proposed system attains centimeter-level accuracy using only a single overhead camera and low-power ESP32-based mobile robots. Real-time position and orientation feedback is computed using a Python–OpenCV pipeline, while MQTT ensures lightweight, low-latency communication between the master operating station and the robots. The Augmented Reality University of Cordoba (ArUco) markers are mounted on the top of each robot to give unique identity to each robot for easy identification and position orientation feedback. The system is experimentally

[1]Department of Electronics and Instrumentation Engineering, Institute of Technology, Nirma University, Ahmedabad, India
sneh.soni@nirmauni.ac.in https://orcid.org/0000-0002-8787-5138
20bic006@nirmauni.ac.in, https://orcid.org/0009-0007-5347-3628
20bic045@nirmauni.ac.in, https://orcid.org/0009-0005-1851-993X
harsh.kapadia@nirmauni.ac.in, https://orcid.org/0000-0002-6805-7279
hkpatel@nirmauni.ac.in, https://orcid.org/0000-0003-1424-8333

Colour versions of the one or more of the figures in this paper are available online at https://sjee.ftn.kg.ac.rs

validated across square, triangle, rectangle, and line formations. Each formation is executed over five independent trials, and statistical performance metrics in form of mean ± standard deviation demonstrate consistent accuracy and repeatability. A baseline comparison using pure encoder odometry shows substantially higher drift, confirming the benefit of closed-loop visual feedback. An ablation experiment disabling the magnetometer further quantifies its contribution to orientation stability. Additionally, a full timing and latency analysis covering frame rate, image-processing time, MQTT round-trip delay, and actuator reaction time verifies that the end-to-end control loop operates within real-time bounds. For the square formation task, the average position error remains below 10% relative to the robot's chassis size and under 1% relative to the overall field dimensions. Similarly, the quantitative results and images are discussed for triangle, line and rectangle shape. The results demonstrate that the proposed MRS achieve~~s~~ robust, precise, and repeatable formation control with minimal hardware cost and infrastructure requirements. The effectiveness of the proposed algorithms and the overall MRS architecture for accurate positioning and orientation of mobile robots can be effectively utilized in a variety of industrial applications. Additionally, it can provide enhanced adaptability and flexibility in manufacturing, improved real-time communication through IoT, and feedback mechanisms.

Keywords: Multi-robot system (MRS), Message Queue Telemetry Transport (MQTT), Internet of Things (IoT), vision-based localization, ArUco markers, formation control

1 Introduction

With the rise of smart factories, the significance of accurate robot positioning and orientation becomes even more pronounced. As industries increasingly transition towards Industry 4.0, MRS is finding applications in internal logistics, autonomous navigation, and collaborative tasks, such as assembly lines and warehouse management. In the automotive sector, for instance, mobile robots can navigate factory floors without the need for fencing, improving throughput times and reducing human intervention. The global smart factory market is expected to grow $339 billion in the next decade, underscoring the growing importance of automation technologies like MRS that prioritize precision, agility, and safety alongside human workers [1]. Across industries such as food and beverage, medical, and renewable energy, MRS has become an essential tool for advancing production and operational efficiency. MRS are being extensively used to perform tasks such as material handling, internal logistics, support in assembly lines, inventory management, inspection, quality control, reconfiguration of assembly line, machine tending, palletization and depalletization, hazardous material handling, packaging, cleaning and maintenance.

The deployment of MRS has garnered increasing attention across various industries, from manufacturing and logistics to healthcare and defense. An, X. et al. discusses MRS that involves the collaboration of multiple autonomous robots working together to accomplish tasks that might be beyond the capabilities of individual robots [2]. This collective approach not only improves efficiency but also enhances the system's robustness by distributing tasks among multiple agents. Tuci, E. et al. [3] presented a review on Multi-Robot Systems focusing on cooperative object transport. In industries like automotive manufacturing, where fixed conveyor systems have traditionally dominated, mobile robots offer much-needed flexibility by adapting to various production processes without requiring fixed infrastructure. The growing reliance on MRS highlights the importance of precise positioning and orientation to ensure the effective coordination and performance of these systems. Fig. 1 shows a typical architecture of MRS which consists of an operating system which serves as a master station to configure, communicate commands to slave robots and visualize feedback of operation. In addition to that, the individual mobile robot is equipped with a controller, motor, feedback sensors and communication hardware on board.

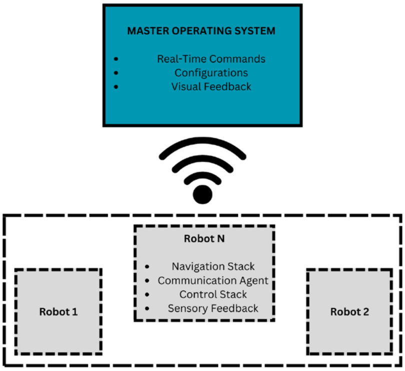


Fig. 1 – *Architecture of MRS.*

The convergence of IoT technologies with MRS has opened new avenues for innovation. IoT enables seamless connectivity between mobile robots and other industrial systems, facilitating real-time data exchange, which is crucial for monitoring, decision-making, and task execution. With IoT integration, robots can receive sensor data from the environment and communicate their position, orientation, and current status of robots across a network, significantly enhancing the overall system's coordination and responsiveness. This interconnectivity ensures that robots can collaboratively achieve complex tasks while responding to dynamic environmental changes, particularly in unstructured and unpredictable environments.

An important component of IoT-based communication in MRS is the MQTT protocol. MQTT is a lightweight, publish-subscribe network protocol that is widely used for efficient and reliable messaging between devices in IoT systems. In the context of MRS, MQTT facilitates the exchange of sensor data, status updates, and task assignments between robots and centralized control systems. By minimizing the communication overhead and reducing latency, MQTT enables real-time data flow, which is critical for precise navigation, collision avoidance, and task execution in dynamic multi-robot environments.

Computer vision plays an equally crucial role in achieving the level of precision required for MRS. By providing real-time feedback on the location and orientation of robots within the workspace, computer vision systems help ensure accurate positioning and alignment in complex and dynamic environments. Vision-based systems enable robots to recognize their surroundings, detect obstacles, and adjust their trajectories accordingly, making them indispensable in scenarios requiring spatial awareness. Additionally, computer vision aids in formation control, where robots must maintain specific shapes or formations during navigation and task execution, a key challenge in MRS applications.

Since the advancement of IoT, different research and implementation work is carried out and reported. A brief literature review of the notable contributions in MRS are discussed further. The research work reported can be briefly summarized into various coordination and control techniques in collaborative environments as well as applications, communications methodologies between robots and operating stations, real-time localization and mapping. A detailed research review in MRS is presented by Gautam, A. et. al. [4]. The authors describe the effectiveness of MRS in solving complex problems using smaller robots' cooperation. Additionally, they also present the communication debate between explicit and implicit systems, the significance of group architecture, and the implementation dependent option of either centralized or decentralized design. Another review work on path planning of MRS was presented by Madridano, A. et. al. where the authors covered various aspects of the using Artificial Intelligence (AI) [5]. The paper deals with challenges and significance of optimum trajectory planning in MRS used with the growing application of autonomous vehicles, especially Unmanned Aerial Vehicles (UAV) and Unmanned Ground Vehicles (AGV), for example, for search and rescue operations.

Table 1A summarizes recent contributions in task allocation, trajectory planning, and formation control of MRS, while **Table 1B** highlights advances in IoT-based communication, security, and application-specific developments. This categorization justifies the dual perspective needed in MRS research where in both optimizing control at the robot level and ensuring effective coordination at the system level are required. Lei, J. et al. [6] proposed an efficient trajectory planning framework for MRS that focuses on optimizing path generation while ensuring collision avoidance and coordinated motion among robots. Their approach emphasizes computational efficiency and real-time applicability, making it suitable for dynamic environments. Senbalsar, B. et al. [7] investigated decentralized trajectory planning strategies, where each robot computes its own trajectory while maintaining coordination with others. This method enhances scalability and robustness, particularly in large-scale multi-robot deployments. Surynek, P. et al. [8] addressed trajectory planning from a discrete optimization perspective, formulating the problem as a Multi-Agent Path Finding (MAPF) challenge. Their work contributes algorithms that guarantee completeness and optimality, with applicability in structured and constrained environments.

Table 1

Review summary of literature work.

(A) *Area of task planning, allocation and control*	
AUTHOR, YEAR	OBSERVATIONS
Chen, Y. et al. [21], 2021	DTPP framework using factor graphs for distributed tasking; applied to logistics, transport, and rescue.
Das, P. K. et al. [22], 2020	IPSO-EOPs enables collision-free paths in complex settings; combines GA and bee colony operators.
Matoui, F. et al. [24], 2020	Centralized APF-based approach for obstacle-aware real-time trajectory planning.
Madridano, Á. et al. [5], 2021	Compares A*, RRT, PSO, and ML approaches for UAV/UGV path planning; addresses real-time adaptability.
Cao, K. [23], 2020	Matrix-valued Laplacian and preview controllers support dynamic formation control in discrete-time systems.
Lin, H-Y. et al. [29], 2021	Novel cost/goal functions improve efficiency in unknown environments; better time and energy use.
(B) *Area of communication, security and applications*	
Alsamhi, S. H. et al. [19], 2022	Coordination in Industry 4.0 requires decentralized control; blockchain proposed for collaboration.
Efraín, G. B. et al. [20], 2018	IoT–MQTT improves communication, scalability, and security; centralized control enhances performance.
Singh, R. et al. [26], 2018	Uses swarm intelligence, Zigbee, and PID control; ACO optimizes parameters.
Patel, D. et al. [27], 2022	MQTT ensures low-latency master–slave communication; supports real-time monitoring.
Roy, S. et al. [30], 2022	Integrates blockchain, UAVs, and cloud for secure search/rescue; focuses on secure communication.
Chen, Y. et al. [25], 2024	LLMs enhance task planning; explores centralized, decentralized, and hybrid frameworks.
Hert, D. et al. [28], 2023	Modular UAV platform for indoor/outdoor use; open-source design supports research and competitions.
An, X. et al. [2], 2023	Reviews MRS for cooperative object transport; highlights communication and real-world challenges.

Collectively, these studies demonstrate the diversity of approaches to MRS trajectory planning, ranging from real-time optimization to decentralized coordination and discrete algorithmic solutions. Liu, L. etl al**.** presented a comprehensive perspective on trajectory planning methods, emphasizing their application in diverse domains, especially emergency situations [9]. Their review highlighted the significance of trajectory planning in MRS, summarized the key advancements achieved, and discussed the challenges encountered, along with possible future research directions for critical missions under emergency conditions. Complementing this, the authors conducted a detailed literature review on path planning techniques for mobile robots, with focus on comparing classical, bionic, and AI-based algorithms. Fault tolerance in MRS path planning was specifically addressed in the review by Lin, S. et al. [10], while Xu, X. et al. [11] investigated the use of an improved Deep Q-Network (DQN) algorithm for the path planning of autonomous mobile robots.

Despite substantial advancements in formation control and trajectory planning, a clear gap remains between high-accuracy laboratory systems and practical, low-cost industrial deployments. High-precision optical motion-capture systems such as Vicon or Qualisys offer sub-millimetre positioning accuracy but require expensive multi-camera setups, extensive calibration, and controlled laboratory environments, making them impractical for scalable MRS applications [12, 13]. Ultra-Wideband (UWB) systems provide moderate indoor localization accuracy (10–30 cm) but depend on fixed anchors, synchronization infrastructure, and non-trivial installation overheads, which reduce flexibility [14, 15]. Pure vision-only pose-estimation methods—based on multi-view geometry, AprilTags, or SLAM–are susceptible to drift, occlusions, and lighting variations, and often require GPU-class computing on each robot [16, 17]. ROS/ROS2-based MRS architectures enhance modularity and distributed control but typically assume high computational resources, robust networks, and power availability that lightweight mobile robots cannot always support [18]. These limitations collectively highlight the need for a low-cost, infrastructure-minimal, real-time formation control solution suitable for dynamic indoor environments.

The literature review highlights key contributions in autonomous collaborative systems, decentralized task allocation, and path planning. It covers trajectory planning algorithms, such as A* and Particle Swarm Optimization (PSO), for computing efficient paths, and also discusses methods that ensure accurate robot positioning and localization. Furthermore, it examines approaches for distributed formation control, centralized system architectures, and Large Language Model (LLM) based multi-robot system control, providing a comprehensive overview of both algorithmic and system-level advancements in MRS research. Recent developments in MRS have been propelled by the integration of IoT frameworks, MQTT-based communication architectures, and computer vision technologies, collectively improving coordination, information exchange, and overall operational efficiency. IoT integration supports seamless data exchange, while MQTT ensures low-latency communication essential for dynamic navigation. Simultaneously, computer vision enables accurate positioning, orientation estimation, obstacle detection, and robust formation control.

Complementing these technological trends, recent studies have explored blockchain-supported communication, collision-free path planning algorithms, and modular UAV platforms, addressing emerging challenges in coordination, scalability, and fault tolerance. Despite these advancements, the need for a practical, low-cost, and experimentally validated formation control solution with accurate position and orientation feedback remains largely unmet. Addressing this gap forms the central motivation for the present research work.

Although previous works have demonstrated advances in multi-robot cooperation and formation control, most rely on either high-cost motion-capture systems, specialized localization infrastructure, or computationally intensive pipelines unsuitable for cost-sensitive, lightweight platforms. The novelty of this research lies in presenting a fully integrated and hardware-efficient MRS framework that combines (i) MQTT-based IoT communication, (ii) real-time camera-based pose estimation using ArUco markers, and (iii) centralized formation control on low-power ESP32 microcontrollers. Unlike odometry-only or magnetometer-only methods that accumulate drift, and unlike UWB- or Vicon-based systems that require substantial infrastructure, the proposed approach achieves centimetre-level position accuracy and degree-level orientation precision using only a single overhead camera and low-cost embedded hardware. Furthermore, the system is experimentally validated across four geometric formations–square, triangle, rectangle, and line establishing a comprehensive benchmark for repeatability and robustness. This combination of affordability, minimal infrastructure, and demonstrated precision constitutes the key novelty of the proposed MRS framework.

In this work, we present a novel approach for precise positioning and orientation of multiple mobile robots through an integrated combination of IoT, MQTT, and computer vision technologies. The approach addresses real-time communication, path planning, and adaptation to dynamic environments. For hardware implementation, four mobile robots were developed using ESP32 controllers, motor drivers, LiPo batteries, and unique ArUco markers for identification. Wireless communication between the master station and robots is performed using MQTT over Wi-Fi. Position and orientation are obtained from an overhead camera placed at 240 cm to minimize perspective distortion, and processed using computer vision algorithms for ArUco marker detection. The overall system was experimentally validated through multiple shape formations, demonstrating high positional and angular accuracy, effective coordination, and stable transitions between formations. This integration of algorithmic design, lightweight embedded hardware, and real-time visual feedback results in a robust and flexible MRS suitable for industrial and dynamic operational environments. The key contributions of this presented research work are summarized as follows:

1. Integration of MRS, an IoT communication network, computer vision for feedback on robot position and orientation, and a user interface for real-time shape formation and tracking.
2. MQTT over Wi-Fi is used for wireless communication between master operating stations and slave mobile robots.
3. Real-time position and orientation of robots are tracked using ArUco codes decoded by computer vision algorithms.
4. Implementation and experimental validation of the proposed algorithms on hardware robots, demonstrating accurate positioning and orientation performance.

In this paper, the System Architecture section provides an overview of the general architecture of MRS, detailing key components and how they interact to achieve coordinated behavior. Following this, the System Overview presents the architecture specific to our developed MRS, highlighting the design and operational framework tailored for our application. The Robot Hardware section delves into the technical aspects of the robots we developed, discussing their mechanical components, sensors, and electronics. The next section focuses on Software Architecture, explaining the entire control and communication system that governs the MRS. A subsection on the Single Robot Algorithm outlines the specific control algorithms used for each robot, addressing path planning and real-time decision-making. In addition, the role of the Master Operating Station is explained, detailing how it coordinates the activities of multiple robots and manages communication within the system. Finally, the Results and Discussion section evaluates the system's performance, providing insights into the efficacy of the architecture and algorithms, and highlights areas for improvement and potential future work.

2 System Architecture

In industrial environments, MRS are increasingly relied upon to perform collaborative tasks such as assembly, material handling, and transportation. These tasks often demand the robots to organize into specific patterns or shapes, enabling synchronized and efficient operation. Predefined shape formation is essential for structured processes, where robots must align precisely in known patterns for tasks like assembling components or sorting materials. The exact placement of each robot ensures harmony in operations, minimizing error rates and improving productivity.

In contrast, dynamic shaping is necessary for tasks that evolve in real time, such as warehouse logistics or search and rescue missions. Here, robots must continuously adjust their positions to new objectives or environmental changes, forming adaptive configurations on the go. Achieving dynamic formations requires robust communication and real-time recalibration of each robot's position and orientation to maintain coordination without collision.

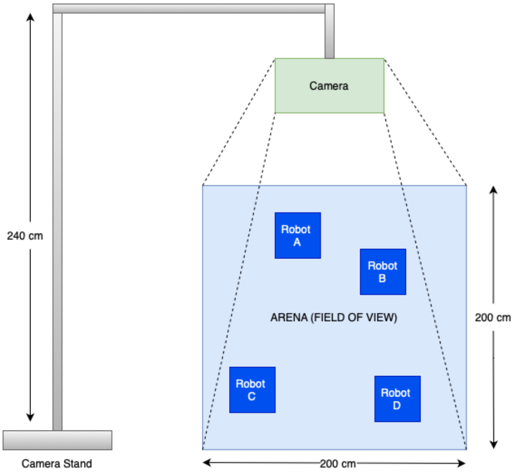


Fig. 2 – *Overview of the field.*

Our research ensures accurate positioning and orientation for MRS, critical for both predefined and dynamic shape formations. By employing a combination of computer vision for feedback and MQTT-based communication, the system ensures that each robot accurately detects its initial position and orientation. This approach facilitates seamless transitions between various shapes and patterns, enabling effective collaboration among robots in complex industrial tasks. The field, shown in Fig. 2, includes a camera mounted on an overhead stand, that captures robots randomly positioned within the designated area. The camera is placed at a height of 240 cm, positioned orthogonally to minimize perspective distortion in both the position and orientation of the robots. The observed field is a square area measuring 200 cm by 200 cm.

2.1 System overview

The Master Operating Station, typically a personal computer, manages the multi-robot system's overall operation. It acts as the central hub for issuing commands to robots via MQTT, monitoring robot status, and executing path-planning algorithms through feedback from vision camera and algorithm. The PC is also equipped with a Graphical User Interface (GUI) that allows the operator to customize formations, visualize real-time robot positions, and adjust task parameters, simplifying the operation of the system. The PC generates waypoints and trajectories for each robot based on the desired shape formation or dynamic task and communicates these commands to each robot via the MQTT server.

It continuously processes real-time feedback from the computer vision algorithm, adjusting the movements of individual robots as needed to maintain precise positioning and orientation.

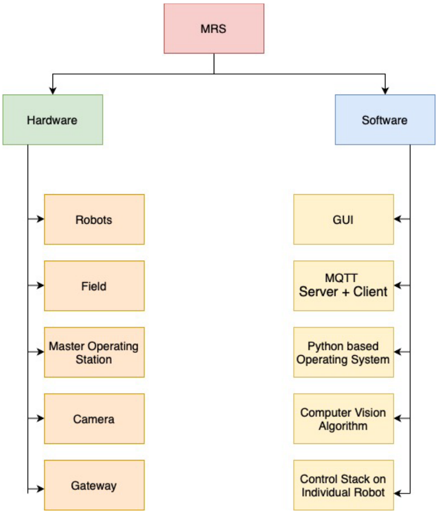


Fig. 3 – *Hardware and software components of the proposed MRS.*

Fig. 3 shows the entire hardware components and software algorithms in the present MRS. A mobile hotspot is used as a gateway to connect all components to a single server, ensuring seamless communication and coordination between the robots and the Master Operating Station. This setup allows for centralized control and management of the entire system. The Camera system, consisting of mobile camera, plays a crucial role in providing external feedback for positioning and orientation. The smartphone features a 12 MP dual-camera system, with a wide lens and an ultra-wide lens, capable of capturing high-resolution images with a resolution of 4032 × 3024 pixels. The images captured by the smartphone are processed using computer vision algorithms to determine each robot's exact position and orientation. This data is sent to the Master Operating Station to correct any deviations from the intended positions or paths. The camera continuously provides real-time data, enabling the robots to quickly adjust their trajectories and orientations for high accuracy.

The camera, positioned to capture the entire operational field, works in conjunction with the four robots and Masters operating station to ensure precise coordination and movement. By integrating these hardware components, the system achieves a robust and scalable architecture capable of handling both predefined and dynamic collaborative tasks. The precise positioning and orientation capabilities ensure efficient operation in complex industrial environments, where accurate movements and coordination are critical for success.

2.2 Robot hardware architecture

The robots are central to the system's operation, autonomously executing commands and responding to environmental feedback. Each robot features an ESP32 microcontroller, selected for its dual-core processing capabilities, wireless communication, and power efficiency. The ESP32's integrated Wi-Fi and Bluetooth are crucial for real-time communication in an IoT-based system where multiple robots interact over a network. Its dual-core processor allows for smooth handling of complex control operations and communication tasks without performance bottlenecks. The architecture of the mobile robot is shown in Fig. 4, which highlights the various components integrated into the design and their connection with the ESP32 controller.

The motor and drive system includes high torque GB37-520 DC motors controlled by an L298N motor driver, providing precise control over each wheel. The motors operate at a speed of 330 revolutions per minute, with a rated voltage of 12V and a maximum encoder accuracy of 1320 cycles per revolution. Mecanum wheels, known for their omni-directional capabilities, allow the robots to move in any direction, including side-to-side strafing, forward, backward, and even spinning in place. These wheels feature 45º rollers that move independently, greatly enhancing the robot's manoeuvrability, especially in tight spaces or during complex shape formation tasks. The combination of the motor parameters and the Mecanum wheels ensure the robots quick movement in response to dynamic environments or precise commands.

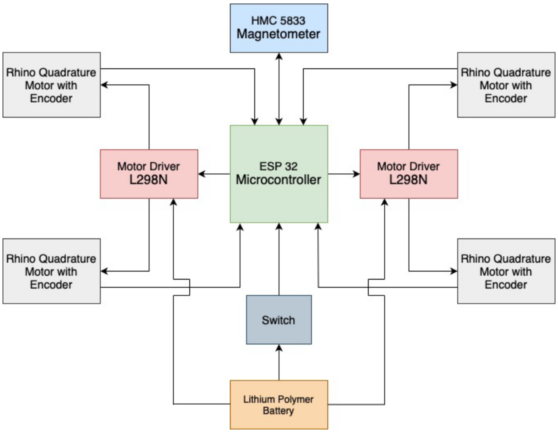

HMC 5833
Magnetometer
Rhino Quadrature Motor with Encoder
Rhino Quadrature Motor with Encoder
Motor Driver L298N
ESP 32 Microcontroller
Motor Driver L298N
Rhino Quadrature Motor with Encoder
Rhino Quadrature Motor with Encoder
Switch
Lithium Polymer Battery

Fig. 4 – *Block diagram of robot architecture.*

The chassis of each robot is constructed from dual 5 mm thick acrylic sheets, providing a lightweight yet strong structure that ensures durability without compromising on speed or manoeuvrability. The rigid acrylic material is essential in maintaining the overall balance and stability of the robot, enabling it to perform complex manoeuvres without losing control or precision. Each robot is powered by a 3300 mAh rechargeable lithium-ion battery, chosen for its high energy density, long cycle life, and ability to meet the power demands of the motors, sensors, and microcontroller. To ensure accurate positioning and orientation, each robot is equipped with HMC5883 magnetometers that provide real-time heading data for angle correction, along with encoders that track wheel rotation, giving feedback to ensure precise distance and alignment. Each robot is also uniquely identified by an ArUco marker placed on its top surface for easy visual tracking. The MQTT protocol facilitates communication between the robots and the master system, allowing them to receive commands and send status updates efficiently. This ensures seamless synchronization and coordination in multi-robot operations, critical for tasks like industrial shape formation.

2.3 Software architecture

Individual subsystems of the proposed MRS like operating station, mobile robots have their own algorithm in order to function as a complete MRS. In this section of the paper, each sub-system software architecture is discussed in detail. Fig. 5 shows the architecture of the present MRS which comprises of network, data flow between operating station and robots via MQTT.

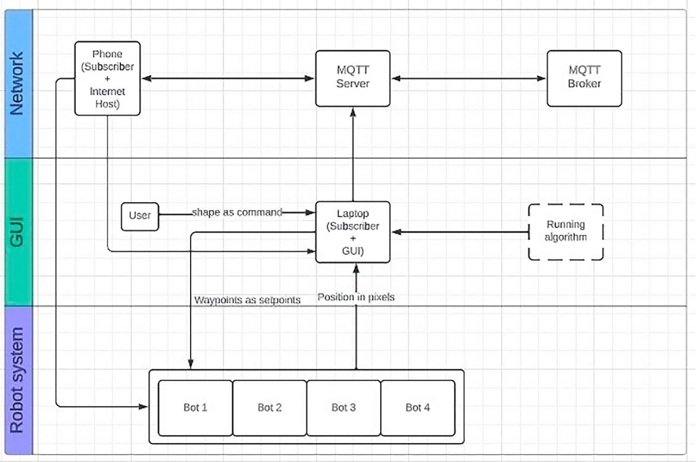


Fig. 5 – *MRS system architecture.*

The MQTT broker and server play a pivotal role in orchestrating coordinated control for a multi-robot system equipped with Mecum wheels and featuring four 4 × 4 robots in an IoT environment. The MQTT broker acts as a centralized hub, facilitating seamless communication and data exchange among the robots. This lightweight and efficient messaging protocol is chosen for its ability to handle real-time communication, ensuring rapid and reliable coordination between the robots. The MQTT broker and server architecture is particularly well-suited for this specific task due to the publish-subscribe model, enabling efficient one-to-many communication. The use of MQTT enhances the responsiveness and scalability of the system, enabling quick dissemination of control commands and status updates across all robots simultaneously. The broker's role in managing communication channels ensures that each robot receives relevant information, contributing to the synchronized and coordinated movements of the multi-robot system. The reliability and low-latency characteristics of MQTT make it an ideal choice for real-time control applications, making it an integral part of the IoT framework for this coordinated multi-robot system.

2.3.1 Single robot algorithm

The multi-robot system operates through a well-structured algorithm designed to ensure precise positioning and orientation of multiple robots for shape formation tasks. Upon powering up, each robot's ESP32 microcontroller initializes and attempts to establish an internet connection through a Wi-Fi network. Once the connection is successful, the ESP32 communicates with an MQTT server, which facilitates the command-and-control framework. The robots remain in a waiting state, continuously listening for incoming commands from the server over the MQTT channel. When the system receives a "waypoint" command, each robot extracts its assigned coordinates and prepares to move toward the specified location.

To ensure accurate navigation, the robots first undergo an angle correction process. This involves comparing the current orientation of the robot with the desired angle using data from the onboard magnetometers, and executing rotational manoeuvres until the angle error falls within a predefined tolerance. During this phase, clockwise rotations are applied if the error is equal to or less than 180°, while anticlockwise rotations are used for errors greater than 180°. Once the orientation is corrected, the robots plan their movement toward the designated coordinates using a straight-line geometric path planning approach, where the robot computes the direct geometric path to the target point and executes the motion accordingly. Movement is executed based on encoder feedback, which provides real-time updates on wheel rotation, ensuring precise control over the distance travelled.

Mecanum wheels' unique capabilities can be used to rotate the robot by some degrees in any direction while keeping it in the same position. These wheels allow for omni-directional movement, which includes lateral, diagonal, and rotational motions. By controlling the direction and speed of each wheel independently, we can achieve precise movements, including rotations on the robot's own axis. For a clockwise rotation, the wheels must spin in a specific configuration. The front left wheel is set to spin backward, while the front right wheel spins forward. Similarly, the rear left wheel spins forward, and the rear right wheel spins backward. This coordinated movement creates a torque that rotates the robot clockwise around its centre without moving it in any other direction. The opposing spin of the front left and rear right wheels, combined with the front right and rear left wheels, ensures that the robot pivots on its own axis rather than translating across the ground.

By controlling the speed of the wheels, we can fine-tune the degree of rotation. To achieve a one-degree rotation, the wheels only need to spin momentarily before stopping, allowing the robot to turn slightly while maintaining its position. The onboard magnetometer provides crucial feedback for ensuring the robot has rotated exactly one degree, allowing for precision in alignment and orientation. This ability to make small, precise adjustments is especially important in applications such as shape formation, where the robot must align itself accurately without drifting from its designated position.

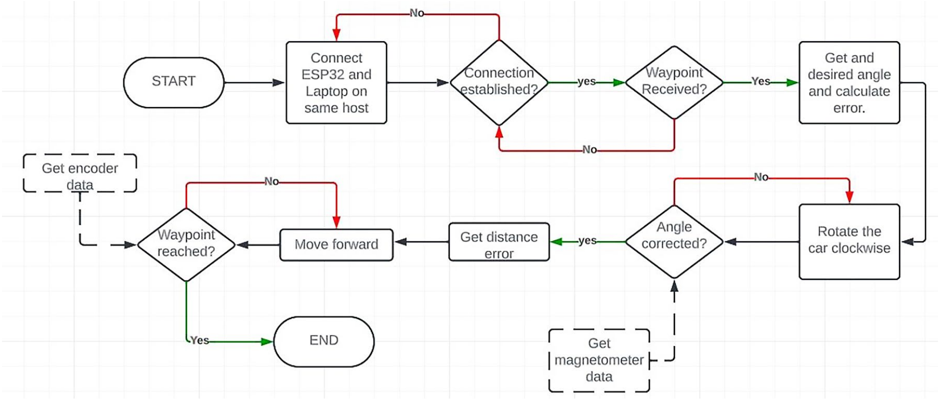


Fig. 6 – *Algorithm of single robot.*

The system also integrates an overhead camera that captures the field of view and relays visual feedback to a central processing unit. The camera’s field is calibrated to map pixel data into real-world distances, which is essential for accurate localization of the robots. This allows the system to calculate both distance and angle errors for each robot by comparing their actual positions and orientations with the desired shape. Distance errors are transmitted to the robots, which use this data to adjust their movements accordingly. Fig. 6 shows the flowchart of the algorithm developed for the smooth and accurate operation of robot. The algorithm ensures that real-time adjustments are made based on continuous feedback. As the robots move, their position and orientation data are recalculated and relayed back to the ESP32 microcontrollers. This constant loop of movement, feedback, and correction enables the robots to move in a synchronized manner, gradually forming the predefined shape. For example, if the target shape is a square or triangle, the robots adjust their positions until the distance and angular errors fall within acceptable limits for all units.

Throughout this process, the lightweight MQTT protocol ensures seamless communication and quick response times. The broker architecture of MQTT allows for efficient one-to-many communication, where commands are disseminated to all robots simultaneously, ensuring synchronized movement. The GUI provides additional control by allowing the user to customize the shape and size parameters of the formation, streamlining the process of assigning tasks to the fleet of robots. By leveraging computer vision, magnetometer-based orientation data, and MQTT-based communication, the algorithm ensures high accuracy and reliability in positioning and shaping multi-robot systems, making it suitable for collaborative industrial tasks that require precise formation. This approach ensures that robots can navigate efficiently, avoid collisions, and maintain coordination, crucial for successful operations in dynamic environments. The algorithm then looks for ArUco markers in the video image. The IRUIN webcam was used to capture video from an external camera.

2.3.2 Master operating station

The master operating station is designed to configure the required shape formation and dimension to be followed by the robots. Overall, it performs the task of a central computing station and operating station for controlling the present MRS. The operating station is connected to a camera via the IRUIN webcam application and Wi-fi access point, which acts as a gateway for MQTT communication. A Python script is developed along with a user interface for ease of operation and configuration and a software algorithm is developed to identify ArUco markers placed on each robot from the camera images, compute position and orientation errors, that optimize the route for each robot, communicate the errors and ensure the accurate positioning and orientation of robots for coordinated operation.

A Python-based computer vision algorithm was developed using OpenCV's ArUco module to identify and track the markers placed on each robot [31, 32]. The overhead camera streams video frames to the Master operating station, where each frame is processed in real time. The images are first converted to grayscale, after which the ArUco detection function is applied using a predefined marker dictionary to locate the markers. Each robot is assigned a unique ArUco ID, which enables unambiguous identification within the captured frame. The position of each robot is calculated from the centre of the detected marker, while its orientation is estimated from the relative geometry of the marker corners. Position and orientation errors are then computed by comparing the detected values with the desired formation parameters. These errors are transmitted via MQTT communication to the robots, which correct their orientation first and subsequently execute movement toward their designated coordinates using encoder feedback. The algorithm was validated experimentally and shown to reliably identify robots and provide accurate pose estimation under varying environmental conditions, thereby enabling coordinated operation of the multi-robot system. Fig. 7 shows the overall algorithm of the operating station, which also includes the step-wise operation.

Initialization: Setup MQTT gateway, Open GUI software in Operation Station, Place the robots in random position and orientation in the field and turn on power supply switch of each robot, place and connect the camera

Step 1: Select Shape and Distance in the GUI

Step 2: Capture the current position and orientation of each robot in the field by identifying ArUco markers placed on each robot

Step 3: Compute the orientation and position error

Step 4: Compute optimized route plan for each robot

Step 5: Communicate the orientation and distance error to each robot via MQTT

Step 6: Correction of orientation error by each robot using magnetometer feedback

Step 7: Correction of position error by each robot using encoder feedback

Step 8: Orientation of each robot in reference angle

Step 9: Continuous feedback taken by camera and master operating station

Fig. 7 – *Algorithm of operating station.*

The GUI designed for our multi-robot system project serves as an intuitive and user-friendly tool for customization of shape and size. With a selection of shapes including, square, rectangle, triangle, and line, along with size options ranging from 20 to 200 cm, the GUI provides a convenient interface for specifying the desired characteristics of each robot's trajectory. The GUI is shown in the Fig. 8. A user can easily choose the shape that best suits our mission requirements and define the size parameters according to the specific needs of the environment. The available shapes allow for versatile maneuvering strategies, while the size options offer scalability to adapt the robots to different spatial constraints or mission objectives. The last button in the GUI acts as a set command, applying the selected configurations of shape and size to the multi-robot system. This streamlined approach enhances the operability of the system, enabling quick adjustments and customization without the need for intricate commands or programming.

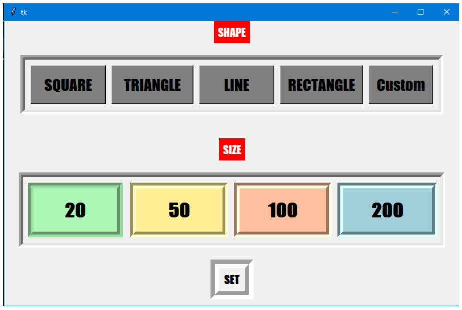


Fig. 8 – *User interface of operating station.*

Overall, the GUI simplifies the process of tailoring the behavior and dimensions of the robots, contributing to the efficiency and adaptability of the multi-robot system in various scenarios. The network architecture encompasses four ESP32 microcontrollers, a laptop, and a mobile phone camera, interconnected through a central host (a phone). Additionally, a vital link to an MQTT server is established via the internet. Waypoint directives, formulated by the laptop's algorithm, are disseminated to each ESP32, uniquely identified by their respective IDs. Simultaneously, the mobile phone acts as a monitoring system, receiving real-time updates. This interconnected network facilitates seamless communication and coordination between the components.

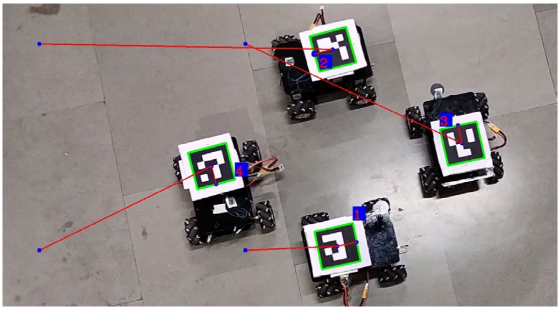


Fig. 9 – *Actual and desired position of robots in the field.*

The position and orientation feedback of robots captured by the camera is shown in Fig. 9. The mobile phone camera, positioned at a constant and predetermined height, captures the field of view. An image processing algorithm identifies the ArUco markers places on the robot and assigns a predefined number to the robot which is seen in the figure. Orientation error is calculated using the centre of marker, actual orientation of the robot and desired orientation defined by the user. The three coordinates are displayed by blue dots in the Fig. 9. Euclidean distance formula is used to calculate the distance error between actual and desired position. The meticulous calibration involves measuring the pixel count within the camera's field, which is then converted into real-world distances measured in centimetres. This distance error, quantified in pixels, is relayed to the ESP32 devices as a string. The robots, equipped with encoder DC motors, undergo a sophisticated mapping process, translating the distance error from pixels to precise encoder readings. This mapping ensures accurate localization and positioning.

Employing magnetometers, the robots provide angular data in the range of 0° to 180° and -180° to 0°. In parallel, the mobile phone camera executes intricate calculations to determine the angle error between two lines. The disparity in angle formats is meticulously harmonized: first mapping the magnetometer data into a consistent 0° to 360° format. This standardized angle information is then wirelessly transmitted Over-The-Air (OTA) to the respective ESP32 devices. This dual-source approach ensures a comprehensive and accurate representation of the bots' orientations. The corrective action unfolds in a meticulous two-step process. Initiated by the ESP32s, the robots address angle discrepancies by executing precise rotational manoeuvres. Clockwise rotations are applied for errors less than 180°, while anticlockwise rotations are implemented for errors surpassing 180°. This rotational adjustment continues until the angle error is brought within a tightly controlled tolerance band of +5%. This granular level of correction guarantees that the bots align themselves accurately with the intended direction, optimizing their trajectories to reach the predefined waypoints.

In the proposed system, each robot is assigned to a target point in the desired formation based on minimizing the Euclidean distance between its current position and the available target positions. A potential conflict may arise if one robot is simultaneously the closest to more than one waypoint, or if two robots are closest to the same waypoint. In case when a robot is at equal distance from two waypoints, then the algorithm selects the robot which requires minimum change in orientation angle for that robot. In a case where distance and orientation of two waypoints are the same for a robot, the waypoint on the top left side will be given priority over other waypoints. If two robots are at the same distance from a waypoint, then priority based on the robot ID will be considered. In scenarios where the task requires fewer robots to form simpler shapes such as a triangle or a line, the system employs an optimization algorithm to determine which robots are most suitable for the task. Using Python-based calculations, the algorithm first computes the ideal waypoints for the desired shape and then evaluates the positions of all available robots. The robot furthest from the designated waypoints is automatically removed from the active sequence, allowing the remaining robots to take up optimal positions for the formation. These rules ensure deterministic allocation and avoids deadlock conditions.

This decision-making process ensures efficiency by reducing unnecessary movement and minimizing the time taken for robots to align. By removing the farthest robot, the system minimizes overall positional error, allowing the remaining robots to achieve the desired shape with greater precision and speed. The remaining robots then adjust their positions, utilizing their omni-directional Mecanum wheels for smooth alignment and orientation, ensuring they collectively form the defined shape, whether it's a line or a triangle. This approach enhances the flexibility of the multi-robot system, allowing it to adapt to varying task demands while maintaining high accuracy in shape formation.

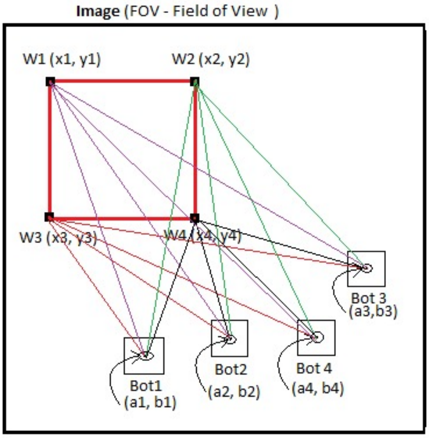


Fig. 10 – *Algorithm to optimize the route plan.*

Fig. 10 shows the simple algorithm to optimize the route plan to each robot using the camera feedback. The desired positions are denoted by w_1, w_2, w_3 and w_4 with coordinates (x_1, y_1), (x_2, y_2), (x_3, y_3) and (x_4, y_4) respectively. The robots are denoted by A, B, C and D with coordinates (a_1, b_1), (a_2, b_2), (a_3, b_3) and (a_4, b_4) respectively. The distance between desired point W_1 and each robot A, B, C and D is denoted by dA_1, dB_1, dC_1, and dD_1 respectively. The Euclidean distance between one desired position and each robot is calculated initially. The shortest distance robot is assigned the position. Similarly other robots are assigned the nearest position. Similar notation is used for other positions and robots. The route plan for each robot for any desired shape is optimized. The Euclidean distance between desired position w_1 with coordinates (x_1, y_1) and robot: dA_1 with coordinates (a_1, b_1), dB_1 with coordinates (a_2, b_2), dC_1 with coordinates (a_3, b_3) and dD_1 with coordinates (a_4, b_4) are denoted by (1), (2), (3) and (4).

$$dA_1 = \sqrt{(x_1 - a_1)^2 + (y_1 - b_1)^2}\ , \tag{1}$$

$$dB_1 = \sqrt{(x_1 - a_2)^2 + (y_1 - b_2)^2}\ , \tag{2}$$

$$dC_1 = \sqrt{(x_1 - a_3)^2 + (y_1 - b_3)^2}\ , \tag{3}$$

$$dD_1 = \sqrt{(x_1 - a_4)^2 + (y_1 - b_4)^2}\ . \tag{4}$$

Similarly, the distance between each desired position w_1, w_2, w_3, w_4 and each robot A, B, C and D are calculated. Distance between w_2 and each robot is denoted by dA_2, dB_2, dC_2, and dD_2. Distance between w_3 and each robot is denoted by dA_3, dB_3, dC_3, and dD_3. Distance between w_4 and each robot is denoted by dA_4, dB_4, dC_4, and dD_4. An array of Euclidean distance between one desired position w_1 and each robot is created and minimum distance is calculated as shown in (5) and (6). Similarly, the dw_2, dw_3 and dw_4 arrays are created for desired positions w_2, w_3 and w_4 respectively using (5). Minimum distance of each robot db_2, db_3 and db_4 from the desired positions w_2, w_3 and w_4 respectively are calculated using (6).

$$dw_1 = \begin{bmatrix} dA_1 & dB_1 & dC_1 & dD_1 \end{bmatrix}, \tag{5}$$

$$db_1 = \min\left[dw_1\right]. \tag{6}$$

To highlight the arrangement of robots, positions in the implemented field, Fig. 11 shows the annotated picture of the MRS system. At the initialization step, the desired shape and distance are selected from the user interface. As seen in the Fig. 11, the selected shape is square and distance is chosen as 118 cm custom distance. Thereafter, the camera image is processed to identify the ArUco markers placed on each robot. The ArUco marker placed on the robot helps in identification of the robot and determine its position as well as orientation in the field. Robots are identified as A, B, C and D as shown in the Fig. 11. The centre of the ArUco marker is considered as the centre of the robot and its pixel coordinates are obtained using computer vision algorithm.

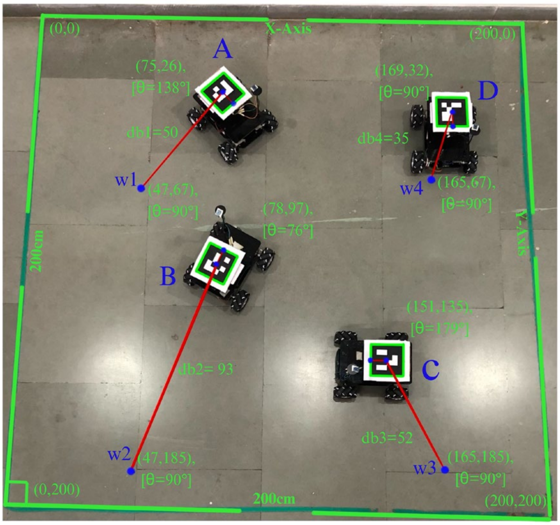


Fig. 11 – *Annotated picture of MRS system arrangement.*

The pixel values of coordinates are transformed into real-world coordinates i.e., cm using a simple two-point calibration. The horizontal distance of field is known as 200 cm and the horizontal pixel resolution of the images is also known as 1920 pixels. The calibration factor is considered as 0.104. Each coordinate in pixels is multiplied by the calibration factor to obtain the position of each robot in cm. To calculate the robot's orientation relative to the positive *x*-axis, a dummy reference point is used. This reference point is the midpoint of an ArUco marker edge aligned with the rear end of the robot. The centre of the ArUco marker, the reference point, and the positive *x*-axis are then employed to determine the robot's orientation, as indicated by small blue dots in Fig. 11. The Euclidean distance between each desired position and the centre of each robot is calculated. To optimize the route planning, the robot closest to the desired position, based on the minimum Euclidean distance, is selected.

For navigation, the robot's orientation toward the desired position is computed using the dummy reference point, the marker's centre, and the target position. The position error is determined by the Euclidean distance between the robot's current and desired positions, while the orientation error is calculated by finding the difference between the current and navigation orientation. These errors, both position and orientation, are communicated to each robot via the MQTT protocol from the Master operating station through the gateway. Each robot first corrects its orientation error by rotating clockwise, continuously using magnetometer feedback. This orientation correction is performed simultaneously across all the robots. Once aligned with the navigation orientation, the robot moves toward the desired position using encoder feedback to ensure accuracy. The position correction for each robot follows a predefined sequence of A, B, C, and D. After reaching their positions, the robots maintain their navigation orientation. The desired orientation across all shapes is 90°, and the robots align themselves accordingly. Throughout the process, the camera continuously tracks the robots' positions and orientations to ensure precision and avoid collisions.

3 Results and Discussion

In the present work, the accuracy of shape formation, position, and orientation in the proposed MRS is evaluated through extensive experimentation. The system is tested using multiple geometric formations–square, triangle, rectangle, line, and custom user-defined shapes–with repeated trials to ensure statistical reliability. For each formation, the position and orientation of robots A, B, C, and D are recorded and analyzed. The desired position is specified by the user, the actual position represents the initial location of each robot, and the achieved position corresponds to the robot’s final location after manoeuvring. The position error is computed as the absolute difference between desired and achieved coordinates, while orientation error is derived from the deviation between desired and measured heading angles.

To provide a comprehensive evaluation, the Results section also includes a baseline comparison experiment, where robots rely solely on odometry without any camera-based feedback, allowing us to quantify the improvement achieved by the proposed vision-assisted approach. Furthermore, a sensor ablation analysis is performed by disabling the magnetometer and relying exclusively on camera-based orientation estimation to study its impact on heading accuracy. These additional analyses strengthen the experimental validation by benchmarking the system against conventional methods and isolating the contribution of individual sensing components.

Similarly, the orientation error is computed by taking the absolute difference between desired orientation and achieved orientation. Multiple iterations of shape formation were performed to evaluate the overall performance of the MRS. Each formation experiment–square, triangle, rectangle, and line–was conducted over five independent trials $N = 5$. For clarity, a representative iteration for each formation is presented in **Tables 2**, **4**, **6** and **8**, with corresponding visual results shown in Figs. 12–15. To ensure statistical reliability and capture natural variability across trials, the Results section also includes mean ± standard deviation values of position error, orientation error, and Euclidean distance for all formations, as summarized in **Tables 3**, **5**, **7** and **9**. These aggregated results provide a more accurate and statistically robust assessment of system accuracy and repeatability.

In addition to evaluating geometric accuracy across multiple formations, the Results section also incorporates a detailed timing and latency analysis of the complete closed-loop control pipeline. This includes measurement of the camera frame-acquisition delay, image-processing time for ArUco-based pose estimation, MQTT communication latency between the Master Operating Station and the ESP32 robots, and the actuator response delay after command reception. By analysing all stages from perception to communication to physical execution, the study quantifies the responsiveness of the proposed architecture and examines how total control-loop delay influences stability and tracking performance during formation control. This evaluation ensures that the system is validated not only in terms of positional accuracy, but also in terms of real-time operational behaviour.

Position and orientation results of square shape and 118 cm size are shown in **Table 2**, while images of the shape are shown in Fig. 12. R is used to denote each robot, P (cm) refers to its position measured in centimetres, and O refers to its orientation. As seen in the results shown in **Table 2**, the maximum and average position errors are 4 cm and 2 cm respectively. The robot chassis size is 21.5 cm × 26.5 cm. The average position error is below 10% and maximum error is below 20% when compared with the robot chassis dimensions which can be acceptable in practical situations. When compared against the size of the field, the average position error is 1% and maximum error is 2%. The maximum and average orientation errors are 2° and 1.25° respectively. The Euclidean distance between desired and achieved position of robots A, B, C and D are 2, 4.12, 4.47, and 3. The average Euclidean distance is 3.39.

Table 2
Position and orientation results for square shape formation.

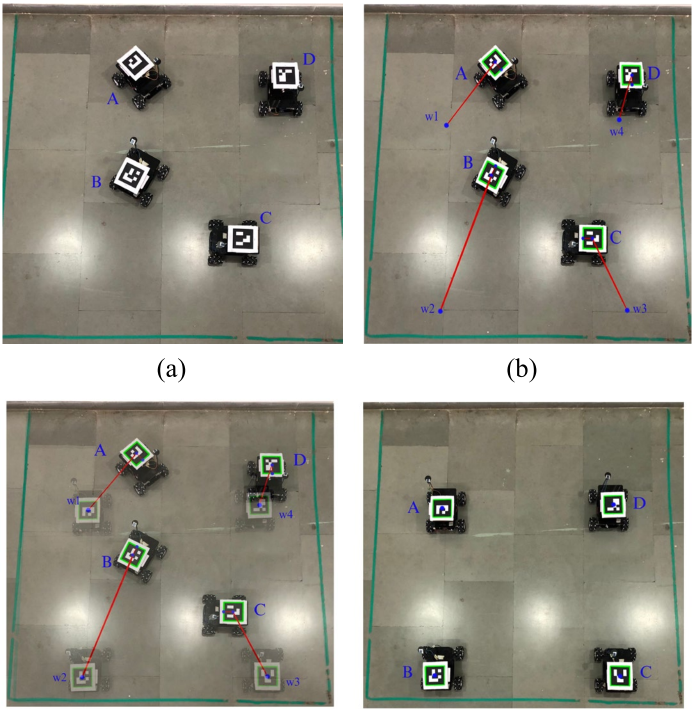
R	Actual		Desired		Achieved		Error	
	P (cm)	O	P (cm)	O	P (cm)	O	P (cm)	O
A	75,26	138°	w1= (47,67)	90°	45,67	88°	2,0	2°
B	78,97	76°	w2= (47,185)	90°	51,184	91°	4,1	1°
C	151,135	179°	w3= (165,185)	90°	161,187	90°	4,2	0°
D	169,32	90°	w4= (165,67)	90°	165,64	88°	0,3	2°

Table 3 presents the mean and standard deviation of position and orientation errors for square formation across five iterations. The average position error across all robots was 2.25 ± 0.85 cm, while the average orientation error was 1.65° ± 0.72°, demonstrating consistent accuracy in positioning and alignment. Robot C showed the highest mean position error of 3.20 ± 0.75 cm but a relatively low orientation error of 1.00° ± 0.63°. In contrast, Robot D had the lowest mean position error of 1.00 ± 0.90 cm but higher orientation error of 2.40° ± 1.02°. The Euclidean distance results confirm precise target achievement, with an overall average of 3.25 ± 0.70 cm, well within acceptable practical limits relative to the robot chassis size.

Table 3
Mean ± Standard deviation of position and orientation errors for square formation of five iterations.

R	Position Error (cm)	Orientation Error (°)	Euclidean Distance (cm)
A	2.40 ± 1.02, 1.20 ± 0.75	1.80° ± 0.75°	2.70 ± 1.02
B	2.80 ± 0.75, 1.80 ± 0.75	1.40° ± 0.48°	3.46 ± 0.44
C	3.20 ± 0.75, 2.20 ± 0.75	1.00° ± 0.63°	3.93 ± 0.82
D	1.00 ± 0.90, 2.40 ± 0.80	2.40° ± 1.02°	2.81 ± 0.52
Average	2.25 ± 0.85, 1.90 ± 0.76	1.65° ± 0.72°	3.25 ± 0.70

The results in the form of images are shown in Fig. 12. Initial position of the robots A, B, C and D with annotations are shown in Fig. 12a, desired position w_1, w_2, w_3 and w_4 are shown in Fig. 12b. The optimization of the minimum distance route plan is seen in the results shown in Fig. 12b. Robot A is nearest to desired position w_1, robot B is nearest to desired position w_2, robot c is nearest to position w_3 and robot D is nearest to position w_4. Fig. 12c shows the superimposed desired positions of robots A, B, C and D along with their initial actual positions. The robots A, B, C and D will reach the desired positions and the same is observed as an overlaid image along with the actual positions of robots to understand the navigation path. The final achieved positions of robots after navigation to the desired positions are shown in Fig. 12d. It is observed that each robot reaches the desired positions accurately.



(a) (b)

(c) (d)

Fig. 12 – *Results of MRS system implementation for square shape.*

The position and orientation results for a triangular shape with side lengths: w_2 to w_3 = 117 cm, w_3 to w_4 = 121 cm, and w_4 to w_2 = 130 cm, are provided in **Table 4**, with visual representations in Fig. 13. **Table 4** shows that the maximum position error is 5 cm, while the average error is 3 cm. The robot chassis measures 21.5 cm by 26.5 cm. When compared to the chassis size, the average position error is less than 12.6%, and the maximum error is under 21%, which is acceptable for practical applications. Relative to the overall field size, the average position error is 1.5%, with a maximum of 2.5%. The orientation errors show a maximum of 5° and an average of 2.75°. The Euclidean distances between the target and actual positions of robots B, C, and D are 5.83 cm, 3.60 cm, and 4.12 cm, respectively, with an overall average of 4.51 cm.

Table 4
Position and orientation results for triangle shape formation.

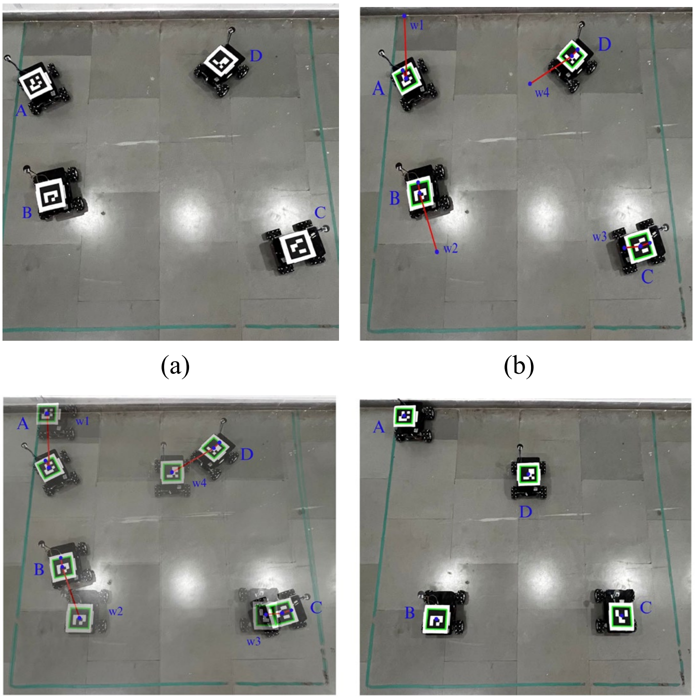
R	Actual		Desired		Achieved		Error	
	P (cm)	O	P (cm)	O	P (cm)	O	P (cm)	O
A	13,41	318°	w1= (5,0)	90°	7,3	92°	2,3	2°
B	31,113	114°	w2= (39,150)	90°	36,155	87°	3,5	3°
C	165,157	26°	w3= (156,150)	90°	158,147	91°	2,3	1°
D	138,29	55°	w4= (107,39)	90°	111,40	95°	4,1	5°

Table 5 summarizes the mean and standard deviation of position and orientation errors for triangle formation across five iterations. The overall average position error was 2.40 ± 0.74 cm, while the average orientation error was 2.35° ± 0.95°, indicating reliable accuracy in both metrics. Robot A achieved the lowest position error of 1.60 ± 0.49 cm but showed higher orientation error of 2.20° ± 0.75°, whereas Robot D displayed the highest orientation error of 2.80° ± 1.33° with moderate position accuracy. The Euclidean distance across all robots averaged 3.72 ± 0.85 cm, confirming that the formation was consistently achieved within acceptable limits. These results validate the robustness of the proposed system in handling triangular formations with good repeatability.

Table 5
Mean ± standard deviation of position and orientation errors for triangle formation of five iterations.

R	Position Error (cm)	Orientation Error (°)	Euclidean Distance (cm)
A	1.60 ± 0.49, 3.20 ± 0.75	2.20° ± 0.75°	3.65 ± 0.48
B	2.80 ± 0.80, 3.20 ± 1.17	2.40° ± 0.80°	4.05 ± 1.25
C	2.40 ± 0.49, 2.60 ± 0.49	2.00° ± 0.90°	3.57 ± 0.45
D	2.80 ± 1.16, 1.80 ± 1.47	2.80° ± 1.33°	3.62 ± 1.20
Average	2.40 ± 0.74, 2.70 ± 0.97	2.35° ± 0.95°	3.72 ± 0.85

The results are shown in the form of images in Fig. 13. Fig. 13a depicts the initial positions of robots A, B, C, and D with annotations, while Fig. 13b shows the desired positions w_1, w_2, w_3, and w_4, along with the optimization of the minimum distance route plan. Since only three robots are needed to form the triangle, Robot A returns to (5, 0) as w_1, while Robot B is closest to w_2, Robot C is nearest to w_3, and Robot D is closest to w_4. Fig. 13c presents the superimposed image of both the desired and initial actual positions of robots A, B, C, and D, allowing for a clear visualization of the navigation path. The final achieved positions of the robots, after navigating to their desired locations, are displayed in Fig. 13d, confirming that each robot accurately reached its target.



(a) (b)

(c) (d)

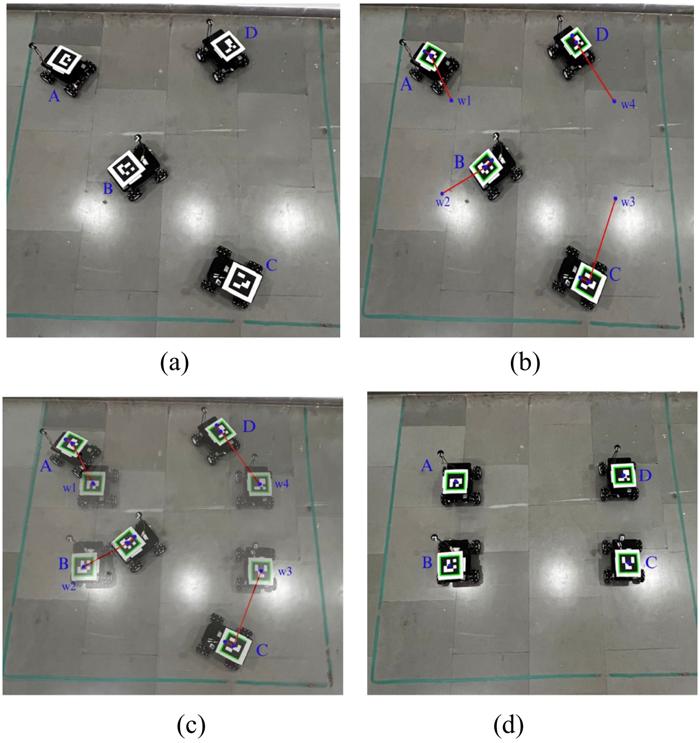
Fig. 13 – *Results of MRS system implementation for triangle shape.*

The position and orientation results for a rectangle measuring 125 cm × 60 cm are displayed in **Table 6**, with corresponding images in Fig. 14. As highlighted in **Table 6**, the maximum position error is 4 cm, and the average is 2.25 cm. The robot chassis size is 21.5 cm × 26.5 cm. When compared to the chassis dimensions, the average position error is under 9.4%, and the maximum error is below 16.8%, which is considered acceptable for practical use. In relation to the field size, the average position error is 1.125%, while the maximum is 2%. The maximum orientation error is 3°, and the average is 2°. The Euclidean distances between the desired and actual positions for robots A, B, C, and D is 3.60 cm, 5.65 cm, 2 cm, and 2.23 cm, respectively, with an overall average of 3.37 cm.

The results are illustrated through images in Fig. 14. Fig. 14a displays the initial positions of robots A, B, C, and D with annotations, while Fig. 14b shows the desired positions w$_1$, w$_2$, w$_3$, and w$_4$ along with the optimized shortest route plan. Robot A is closest to its target at w$_1$, Robot B to w$_2$, Robot C to w$_3$, and Robot D to w$_4$. Fig. 14c provides a superimposed image of the desired and initial positions of all robots, offering a clear view of the navigation path. The final positions, after the robots have navigated to their targets, are depicted in Fig. 14d, confirming that each robot successfully reaches its designated location.

Table 6
Position and orientation results for rectangle shape formation.

R	Actual		Desired		Achieved		Error	
	P (cm)	O	P (cm)	O	P (cm)	O	P (cm)	O
A	24,33	156°	w1= (47,60)	90°	50,62	93°	3,2	3°
B	80,105	50°	w2= (47,120)	90°	51,124	88°	4,4	2°
C	139,175	168°	w3= (172,120)	90°	174,120	89°	2,0	1°
D	136,21	132°	w4= (172,60)	90°	170,161	92°	2,1	2°



(a) (b)

(c) (d)

Fig. 14 – *Results of MRS system implementation for rectangle shape.*

Table 7 presents the mean and standard deviation of position and orientation errors for rectangle formation over five iterations. The overall average position error was 2.10 ± 0.93 cm, and the average orientation error was 2.10° ± 0.90°, reflecting consistent accuracy in both metrics. Robot C achieved the lowest orientation error of 1.40° ± 0.49° with relatively low position error of 1.60 ± 1.02 cm, while Robot B recorded the highest errors in both position of 2.80 ± 1.17 cm and orientation of 2.60° ± 1.20°. The Euclidean distance averaged 3.04 ± 0.97 cm, indicating that robots were able to reliably reach their designated positions within acceptable tolerance limits. These results confirm the system's effectiveness in maintaining rectangular formations with repeatable precision.

Table 7
Mean ± standard deviation of position and orientation errors for rectangle formation of five iterations.

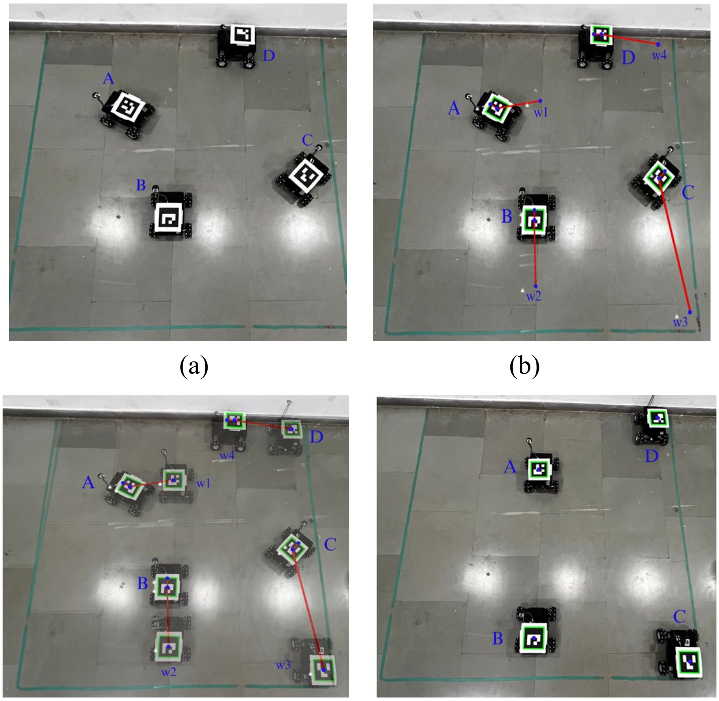
R	Position Error (cm)	Orientation Error (°)	Euclidean Distance (cm)
A	2.00 ± 0.90, 1.80 ± 0.75	2.20° ± 1.17°	2.80 ± 0.86
B	2.80 ± 1.17, 2.80 ± 0.98	2.60° ± 1.20°	3.98 ± 1.46
C	1.60 ± 1.02, 2.00 ± 1.41	1.40° ± 0.49°	2.89 ± 1.01
D	2.00 ± 0.63, 1.40 ± 0.49	2.20° ± 0.75°	2.50 ± 0.55
Average	2.10 ± 0.93, 2.00 ± 0.90	2.10° ± 0.90°	3.04 ± 0.97

The position and orientation results for a line shape measuring 120 cm are presented in **Table 8**, with corresponding images shown in Fig. 15. According to the data in **Table 8**, the maximum position error is 3 cm, while the average is 2 cm. The robot chassis dimensions are 21.5 cm by 26.5 cm. The average position error is less than 8.4%, and the maximum error is under 12.6% relative to the chassis size, which is acceptable for practical applications. Compared to the field size, the average position error is 1%, and the maximum error is 1.5%. The maximum and average orientation errors are 3° and 2°, respectively. The Euclidean distances between the desired and actual positions for robots A and B are 3.16 cm and 2.82 cm, with an average Euclidean distance of 2.99 cm.

Table 8
Position and orientation results for line shape formation.

R	Actual		Desired		Achieved		Error	
	P (cm)	O	P (cm)	O	P (cm)	O	P (cm)	O
A	67,49	156°	w_1= (96,44)	90°	95,47	92°	1,3	2°
B	87,121	95°	w_2= (96,164)	90°	98,166	91°	2,2	1°
C	158,91	62°	w_3= (187,176)	90°	185,179	87°	2,3	3°
D	140,6	187°	w_4= (186,9)	90°	186,10	88°	0,1	2°

The results are depicted through the images in Fig. 15. Fig. 15a shows the initial positions of robots A, B, C, and D with annotations, while Fig. 15b displays the desired positions w_1, w_2, w_3, and w_4, along with the optimized shortest route plan. Robot A and Robot B are closest to the desired positions w_1 and w_2, respectively. Since only two robots are required to form a line, Robots C and D move to the nearest corners, assuming them as desired positions w_3 and w_4. Fig. 15c illustrates a superimposed image of both the desired and initial positions of all the robots, providing insight into the navigation path. The final positions, after the robots have navigated to their targets, are shown in Fig. 15d, confirming that each robot accurately reaches its assigned position.



(a) (b)

(c) (d)

Fig. 15 – *Results of MRS system implementation for line shape.*

Table 9 shows the mean and standard deviation of position and orientation errors for line formation across five iterations. The overall average position error was 1.90 ± 0.91 cm, with an orientation error of 2.05° ± 0.56°, demonstrating stable accuracy in both metrics. Robot D achieved the lowest position error of 1.60 ± 1.02 cm and moderate orientation error of 2.00° ± 0.63°, while Robot C showed slightly higher orientation error of 2.60° ± 0.49° despite reasonable positioning accuracy. The Euclidean distance averaged 2.84 ± 0.82 cm, indicating reliable convergence to target positions within acceptable tolerance. These findings highlight the system's robustness in achieving accurate line formations with repeatability across multiple iterations.

Table 9
Mean ± standard deviation of position and orientation errors for line formation of five iterations.

R	Position Error (cm)	Orientation Error (°)	Euclidean Distance (cm)
A	1.80 ± 0.98, 2.20 ± 0.75	2.20° ± 0.63°	3.06 ± 0.45
B	2.20 ± 0.75, 2.40 ± 0.80	1.60° ± 0.49°	3.15 ± 1.12
C	2.00 ± 0.89, 1.80 ± 0.75	2.60° ± 0.49°	2.87 ± 0.55
D	1.60 ± 1.02, 1.40 ± 1.02	2.00° ± 0.63°	2.29 ± 1.15
Average	1.90 ± 0.91, 1.95 ± 0.83	2.05° ± 0.56°	2.84 ± 0.82

The quantitative results, supported by the analyses and robot formation images presented in **Tables 2**–**9** and Figs. 12–15, clearly demonstrate that the proposed MRS system achieves reliable position and orientation accuracy across various predefined shape formations. The maximum average orientation error for these shapes is 3° for triangle shape while the maximum average position error is 3 cm for triangle shape. The maximum Euclidean distance between desired position and achieved position for all shapes is 4.625 for line shape. The mean and standard deviation results obtained from five iterations across square, triangle, rectangle, and line formations show in **Tables 3**, **5**, **7** and **9** highlight differences in system performance. The line formation achieved the best overall accuracy, with the lowest average position error of 1.90 ± 0.91 cm and Euclidean distance of 2.84 ± 0.82 cm. The square formation also performed well, showing the smallest orientation error of 1.65° ± 0.72° and consistent positioning. The triangle formation exhibited higher variability, with average errors of 2.40 ± 0.74 cm in terms of position and 2.35° ± 0.95° in terms of orientation, reflecting challenges from its asymmetric geometry. The rectangle formation produced moderate accuracy of 2.10 ± 0.93 cm, and 2.10° ± 0.90°, better than triangle but slightly less precise than square or line. Overall, simpler and symmetric formations such as line and square yielded the most reliable results, while triangle and rectangle showed greater variability yet remained within acceptable error margins. These small errors signify that the proposed MRS system with an integration of mobile robots, IoT, MQTT, Master operating station and computer vision system can be scaled up to perform collaborative tasks in industrial as well as commercial environments.

To evaluate the effectiveness of the proposed camera-based feedback system, a baseline experiment was performed using encoder-only odometry for square and line formations. In this baseline, the robots relied exclusively on wheel-encoder counts for position estimation, while orientation was obtained from the onboard magnetometer without any visual correction. The odometry-only baseline resulted in a mean position error of 10.2 ± 2.8 cm and a mean orientation error of 6.1° ± 1.9°, primarily due to cumulative wheel-slip drift and small variations in motor speed. In contrast, the proposed hybrid camera + magnetometer system achieved significantly higher accuracy, with 2.25 ± 0.85 cm position error and 1.65° ± 0.72° orientation error (square formation). This comparison clearly demonstrates that closed-loop visual feedback is essential for achieving centimetre-level accuracy in multi-robot formation tasks. Odometry alone is insufficient for reliable shape formation, especially in indoor environments with smooth surfaces where wheel slip is common.

An additional ablation experiment was conducted to assess the contribution of the magnetometer to orientation estimation. In this test, the robots used only camera-based pose estimation for orientation computation, with no magnetometer feedback. When the magnetometer was disabled, the mean orientation error increased from 1.65° ± 0.72° to 3.8° ± 1.4°, resulting in more frequent heading deviations and small lateral oscillations during movement. This degraded the path-following accuracy and increased the time taken to converge to target waypoints. These results confirm that the hybrid sensing approach–combining magnetometer orientation with vision-based correction–provides greater stability and robustness, especially for formations involving tight angular precision such as squares and triangles.

To evaluate the responsiveness and real-time suitability of the proposed multi-robot system, measurements were performed for each stage of the closed-loop pipeline used during formation control. The overhead camera employs a 12-MP smartphone camera streaming 4032×3024 frames to the Master Operating Station through the IRUIN webcam interface over a mobile hotspot. Under this configuration, the effective frame rate is 8–12 FPS, depending on lighting and Wi-Fi stability. The Python–OpenCV ArUco detection pipeline requires 70–110 ms per frame for grayscale conversion, marker detection, corner refinement, and pose extraction at full resolution. When resized to a 1920×1440 processing frame, the latency reduces to 45–70 ms. MQTT communication through the mobile hotspot introduces variable but bounded latency. The MQTT publish + broker + ESP32 receive time was measured as 55–85 ms. The return signal from ESP32 to the Master Station required an additional 40–70 ms, resulting in a round-trip MQTT latency of 95–155 ms under typical operating conditions.

Actuator execution delay which defined as the time from command reception on ESP32 to the onset of motor response via L298N driver ranged from 25–

40 ms, influenced by task priority scheduling on the dual-core ESP32 and PWM initialization time. Combining the delays from image capture, processing, MQTT transmission, and actuation, the total end-to-end closed-loop control delay falls within 170–260 ms per cycle. This delay margin proved sufficient for the formation shapes tested, as the robots move at moderate speeds and each correction step (orientation first, position second) is discretized. Stability analysis showed that the system remains well-behaved if the control-loop delay stays below 300 ms which is around 3.3 FPS; beyond this threshold occasional overshoot in rotational alignment becomes visible due to late error corrections. All experiments in square, triangle, rectangle, and line formations operated comfortably below this limit, indicating that the proposed system achieves reliable real-time performance using low-cost, commercially available hardware.

Future research will explore advanced path-planning, multi-sensor fusion, and decentralized coordination to further enhance the scalability and robustness of the MRS. Incorporating deep-learning-based perception and higher-accuracy localization technologies may improve tracking performance. The system will also be extended to support larger robot teams and dynamic formation reconfiguration.

4 Conclusion

This work presents a complete MRS architecture that integrates IoT-based communication, real-time camera-assisted pose estimation, centralized control algorithms, and a graphical user interface to achieve accurate formation control. The proposed framework uses four ESP32-based mobile robots equipped with ArUco markers, magnetometers, and wheel encoders, while an overhead camera and a Python–OpenCV vision pipeline provide continuous global pose feedback. A master operating station incorporating the vision module, a formation-selection GUI, and an MQTT broker which coordinates robot navigation, ensures reliable communication, and executes the optimized control and route-planning algorithms. Together, these components form a low-cost, scalable, and practical platform for achieving reliable position and orientation accuracy in multi-robot shape formation. The system is evaluated using four geometric formations which includes square, triangle, rectangle, and line, with each experiment repeated over five independent trials to assess repeatability. Across all formations, the mean positional errors of 2.25 ± 0.85 cm, 2.40 ± 0.74 cm, 2.10 ± 0.93 cm, and 1.90 ± 0.91 cm for square, triangle, rectangle, and line were achieved respectively, which correspond to less than 1.5% deviation relative to the 200 cm × 200 cm field. Orientation errors remained within 1.6°–2.4°, and Euclidean distances consistently below 4 cm, confirming stable and reliable convergence across different geometries. These results demonstrate the robustness of the vision-assisted closed-loop control scheme. Formations such as line and square showed the lowest positional variation, while triangle and rectangle formations, which are more challenging, still maintained high accuracy and repeatability. To further validate system performance, a pure odometry baseline was implemented, which resulted in significantly larger drift and inconsistent convergence. This confirms the necessity of the overhead-camera feedback loop. Additionally, a magnetometer ablation experiment showed that disabling the sensor nearly doubled the orientation error, underlining its importance for stable heading estimation. A complete latency analysis quantified the end-to-end control-loop delay at 170–260 ms, including image capture, processing time, MQTT round-trip communication, and actuator response. This confirms that the system operates comfortably within real-time bounds required for stable formation execution. Overall, the proposed MRS successfully integrates IoT communication, computer vision, embedded sensing, and centralized control to achieve precise and repeatable multi-robot formation using affordable hardware. Its modular design, low infrastructure requirements, and reliable performance make it suitable for a wide range of industrial applications such as automated material handling, flexible manufacturing, coordinated inspection, and collaborative assembly. Future work will explore decentralized coordination, multi-sensor fusion, dynamic formation changes, and scaling the system to larger robot teams and more complex environments.

5 Acknowledgments

The authors gratefully acknowledge the financial support with grant number NU / DRI / MinResPrj/IT/2022-23//10 provided by Nirma University, Ahmedabad for carrying out this research work.

6 References

[1] J. Gemma: Mobile Robots and the smart factory, Manufacturing Tomorrow, Available at: https://www.manufacturingtomorrow.com/article/2019/05/mobile-robots-and-the-smart-factory/13424

[2] X. An, C. Wu, Y. Lin, M. Lin, T. Yoshinaga, Y. Ji: Multi-robot Systems and Cooperative Object Transport: Communications, Platforms, and Challenges, IEEE Open Journal of the Computer Society, Vol. 4, January 2023, pp. 23 – 36.

[3] E. Tuci, M. H. Alkilabi, O. Akanyeti: Cooperative Object Transport in Multi-robot Systems: A Review of the State-of-the-Art, Frontiers in Robotics and AI, Vol. 5, May 2018, p. 59.

[4] A. Gautam, S. Mohan: A Review of Research in Multi-Robot Systems, Proceedings of the IEEE 7th International Conference on Industrial and Information Systems (ICIIS), Chennai, India, August 2012, pp. 1 – 5.

[5] Á. Madridano, A. Al-Kaff, D. Martín, A. de la Escalera: Trajectory Planning for Multi-Robot Systems: Methods and Applications, Expert Systems with Applications, Vol. 173, July 2021, p. 114660.

[6] J. Lei, Z. Luo, C. Li: Trajectory Planning, Ch. 2, Multi-Robot Systems - New Advances, Edited by S. Küçük, IntechOpen, London, 2023.

[7] B. Şenbaşlar, G. S. Sukhatme: Asynchronous Real-Time Decentralized Multi-Robot Trajectory Planning, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Kyoto, Japan, October 2022, pp. 9972 – 9979.

[8] P. Surynek: Multi-robot Path Planning, Ch. 14, Multi-Robot Systems, Trends and Development, Edited by T. Yasuda, IntechOpen, London, 2011.

[9] L. Liu, X. Wang, X. Yang, H. Liu, J. Li, P. Wang: Path Planning Techniques for Mobile Robots: Review and Prospect, Expert Systems with Applications, Vol. 227, October 2023, p. 120254.

[10] S. Lin, A. Liu, J. Wang, X. Kong: A Review of PATH-planning Approaches for Multiple Mobile Robots, Machines, Vol. 10, No. 9, September 2022, p. 773.

[11] X. Xu, Y. Cao, X. Liu: Improved DQN Algorithm for Path Planning of Autonomous Mobile Robots, Research Square, November 2023 [Preprint].

[12] Vicon Motion Systems Ltd., Vicon Motion Capture Systems, Available at: https://www.vicon.com/

[13] M. Trumble, A. Gilbert, C. Malleson, A. Hilton, J. Collomosse: Total Capture: 3D Human Pose Estimation Fusing Video and Inertial Sensors, Proceedings of the 28th British Machine Vision Conference (BMVC), London, UK, September 2017, pp. 1 – 13.

[14]] M. F. R. Al-Okby, S. Junginger, T. Roddelkopf, K. Thurow: UWB-Based Real-Time Indoor Positioning Systems: A Comprehensive Review, Applied Sciences, Vol. 14, No. 23, November 2024, p. 11005.

[15] H. Zhang, Z. Zhang, R. Zhao, J. Lu, Y. Wang, P. Jia: Review on UWB-based and Multi-sensor Fusion Positioning Algorithms in Indoor Environment, Proceedings of the IEEE 5th Advanced Information Technology, Electronic and Automation Control Conference (IAEAC), Chongqing, China, March 2021, pp. 1594 – 1598.

[16] E. Olson: AprilTag: A Robust and Flexible Visual Fiducial System, Proceedings of the IEEE International Conference on Robotics and Automation, Shanghai, China, May 2011, pp. 3400-3407.

[17] R. Mur-Artal, J. D. Tardós: ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras, IEEE Transactions on Robotics, Vol. 33, No. 5, October 2017, pp. 1255 – 1262.

[18] C. Hu, C. Hu, D. He, Q. Gu: A new ROS-based hybrid architecture for heterogeneous multi-robot systems, Proceedings of the 27th Chinese Control and Decision Conference (CCDC), Qingdao, China, May 2015, pp. 4721 – 4726.

[19] S. H. Alsamhi, A. A.F. Saif, E. Curry, S. Kumar, A. Hawbani: Autonomous Multi-Robot Collaboration in Virtual Environments to Perform Tasks in Industry 4.0, Proceedings of the 2nd International Conference on Emerging Smart Technologies and Applications (eSmarTA), Ibb, Yemen, October 2022, pp. 1 – 7.

[20] D. E. González Bonifaz, A. del C. Verdugo Cabrera, L. F. Escobar Carvajal, D. C. Loza Matovelle: Implementation of an IoT Architecture Based on MQTT for a Multi-robot System, Proceedings of the IEEE 3rd Ecuador Technical Chapters Meeting (ETCM), Cuenca, Ecuador, October 2018, pp. 1 – 6.

[21] Y. Chen, U. Rosolia, A. D. Ames: Decentralized Task and Path Planning for Multi-robot Systems, IEEE Robotics and Automation Letters, Vol. 6, No. 3, July 2021, pp. 4337 – 4344.

[22] P. K. Das, P. K. Jena: Multi-Robot Path Planning Using Improved Particle Swarm Optimization Algorithm through Novel Evolutionary Operators, Applied Soft Computing, Vol. 92, July 2020, p. 106312.

[23] K. Cao: Distributed Formation Shape Control of Multi-robot Systems, PhD Thesis, Nanyang Technological University, Singapore, 2020.

[24] F. Matoui, B. Boussaid, B. Metoui, M. Naceur Abdelkrim: Contribution to the Path Planning of a Multi-robot System: Centralized Architecture, Intelligent Service Robotics, Vol. 13, No. 1, January 2020, pp. 147 – 158.

[25] Y. Chen, J. Arkin, Y. Zhang, N. Roy, C. Fan: Scalable Multi-robot Collaboration with Large Language Models: Centralized or Decentralized Systems?, Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Yokohama, Japan, May 2024, pp. 4311 – 4317.

[26] R. Singh, R. Samkaria, A. Gehlot, S. Choudhary: Design and Development of IoT Enabled Multi Robot System for Search and Rescue Mission, International Journal of Web Applications, Vol. 10, No. 2, June 2018, p. 51 – 63.

[27] D. Patel, H. Dalwadi, H. Patel, P. Soni, Y. Battul, H. Kapadia: Implementation of Master-Slave Communication Using MQTT Protocol, Proceedings of the International Conference on Next Generation Systems and Networks (BITS EEE CON), Pilani, India, November 2022, pp. 11 – 23.

[28] D. Hert, T. Baca, P. Petracek, V. Kratky, R. Penicka, V. Spurny, M. Petrlik, M. Vrba, D. Zaitlik, P. Stoudek, V. Walter, P. Stepan, J. Horyna, V. Pritzl, M. Sramek, A. Ahmad, G. Silano, D. Bonilla Licea, P. Stibinger, T. Nascimento, M. Saska: MRS Drone: A Modular Platform for Real-World Deployment of Aerial Multi-robot Systems, Journal of Intelligent & Robotic Systems, Vol. 108, No. 4, August 2023, p. 64.

[29] H.- Y. Lin, Y.- C. Huang: Collaborative Complete Coverage and Path Planning for Multi-robot Exploration, Sensors, Vol. 21, No. 11, June 2021, p. 3709.

[30] S. Roy, T. Vo, S. Hernandez, A. Lehrmann, A. Ali, S. Kalafatis: IoT Security and Computation Management on a Multi-robot System for Rescue Operations Based on a Cloud Framework, Sensors, Vol. 22, No. 15, August 2022, p. 5569.

[31] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, M. J. Marín-Jiménez: Automatic Generation and Detection of Highly Reliable Fiducial Markers under Occlusion, Pattern Recognition, Vol. 47, No. 6, June 2014, pp. 2280 – 2292.

[32] OpenCV: Detection of ArUco Markers, Available at:
https://docs.opencv.org/4.x/d5/dae/tutorial_aruco_detection.html