Swarm Control of Mobile Robots Using an Alternating Signal-Based Activation Algorithm

Nguyen Minh Trieu[1], Trinh Duc Cuong[1] and Nguyen Truong Thinh[1,*]

[1] Institute of Intelligent and Interactive Technologies, University of Economics Ho Chi Minh City–UEH, Ho Chi Minh City, Vietnam
trieunm@ueh.edu.vn,cuongtd@ueh.edu.vn,
thinhnt@ueh.edu.vn

Abstract. In this study, a dynamic robot formation control algorithm is designed to manage multiple nodes of robots that alternate activity in a coordinated manner. Our approach leverages alternating activation, formation maintenance, and collision detection mechanisms to ensure robust performance in environments with multiple robotic units. Specifically, the algorithm activates different robot nodes in a predefined sequence, allowing them to take turns executing tasks while maintaining a stable formation. Each robot in the active node computes its direction based on a potential field method, balancing attractive and repulsive forces to maintain desired distances from other robots and stationary points. Additionally, our algorithm integrates collision detection to prevent overlap and ensure the safe movement of the robots. This framework was implemented and tested in a simulated environment, demonstrating its effectiveness in maintaining formations and preventing collisions. The results show the potential of the proposed method for applications in autonomous multi-robot systems, such as search and rescue, environmental monitoring, and warehouse automation.

Keywords: Dynamic robot formation, multi-robot systems, collision detection, formation algorithm, autonomous robots, alternating activation, simulation.

1. Introduction

The advent of autonomous robotic systems has spurred significant advancements in various fields, ranging from industrial automation to complex search and rescue operations. A fundamental challenge in deploying multiple robots lies in ensuring coordinated movement and efficient task execution while maintaining safe distances and avoiding collisions. The study of Zhang et al. (2022) introduced a predictive data-driven fuzzy compensator for non-holonomic mobile robots, which significantly improved formation control under uncertain conditions. Similarly, Li et al. (2023) developed an adaptive formation control strategy for underactuated autonomous underwater vehicles, demonstrating enhanced coordination in complex aquatic environments. These efforts highlight the growing importance of adaptive

control mechanisms in multi-robot systems. Wang et al. (2021) advanced this further by proposing a perturbation observer-based robust control using multiple sliding surfaces, effectively addressing nonlinear systems with matched and unmatched uncertainties. This work introduces a novel beacon-based formation control mechanism integrated with an alternating node activation approach. Our key contributions include a beacon-based algorithm that combines with alternating node activation to enable effective management of large robot swarms, the introduction of ghost nodes as virtual reference points for flexible and adaptable formation configurations, a potential field approach that balances attractive and repulsive forces to ensure precise formation maintenance and collision avoidance, energy-efficient alternating activation mechanisms optimizing task distribution and extending swarm longevity and extensive simulation results demonstrating the effectiveness of our approach across various configurations. These contributions aim to advance multi-robot systems, offering a foundation for more scalable and efficient swarm operations in domains such as search and rescue, environmental monitoring, and warehouse automation. Formation control algorithms play a pivotal role in addressing these challenges by orchestrating the collective behavior of robots to achieve predefined spatial configurations and operational objectives. In this study, the development and implementation of a dynamic formation control algorithm is designed for robot swarm systems. By harnessing alternating node activation, the algorithm enables robots to take turns performing tasks while maintaining a stable formation, thereby optimizing efficiency and operational longevity. Central to the algorithm is the utilization of a potential field method for individual robot navigation within the formation. This method balances attractive and repulsive forces to guide each active robot towards desired positions relative to other robots and fixed points in the environment. The results underscored the efficacy of our algorithm in achieving robust formation maintenance and collision avoidance capabilities, laying a foundation for its application in real-world settings such as autonomous surveillance, environmental monitoring, and warehouse logistics. This article presents a comprehensive overview of our beacon-based algorithm with alternating node activation, detailing its theoretical underpinnings, implementation strategies, simulation results, and potential applications in autonomous multi-robot systems.

2. Beacon-Based Control

The beacon-based control mechanism employed in this algorithm leverages advanced sensing technologies to enable precise coordination and spatial alignment among multiple robots within a team. First, the system must be initialized by defining the number of robots and nodes, setting up ghost nodes to form the desired configuration, and initializing the positions and orientations of the robots. Next, the beacon-based control mechanism is established by configuring infrared (IR) sensors for distance measurement and magnetometers for heading determination, followed by the implementation of distance and bearing calculations. The formation control algorithm is then developed by applying a potential field approach, which involves creating functions to compute attractive and repulsive forces and determining

heading and movement. Ghost nodes are set up by defining their positions and implementing attraction forces to guide the swarm. The alternating node activation mechanism is incorporated through the definition of activation timers and switching intervals, utilizing the designated algorithm to manage node activation. Collision avoidance is addressed by implementing inter-robot distance checks and developing appropriate avoidance behaviors. In the main control loop, sensor data is continuously read, node activation status is updated, forces and movements for active robots are calculated, collision avoidance is applied if needed, and the robots' positions and orientations are updated. Lastly, real-time visualization of the robot positions and formations is implemented along with data logging for performance analysis. At the core of this approach are infrared (IR) sensors mounted on each robot, which provide accurate measurements of distances to nearby beacons and other robots in the environment. These beacons, strategically placed to define the desired formation, serve as reference points. Upon detecting signals from these beacons via IR sensors, each robot calculates its relative position and orientation, which is crucial for dynamically adjusting its trajectory and speed to maintain specific inter-robot distances, thereby preserving the formation's integrity. The real-time responsiveness afforded by IR sensors allows for continuous position updates as the team configuration evolves.

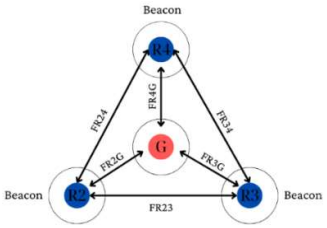

Fig. 1. Diagram of forces and beacons for 1 node

In addition to distance measurement, the algorithm integrates data from magnetometer sensors installed on each robot to determine its heading. By combining heading information with distance data from the IR sensors, each robot computes an optimal path toward its designated position within the formation as Fig.1. This integration of sensor data enhances precise navigation and coordination among robots, improving the overall stability and efficiency of the formation control system. Moreover, the beacon-based control mechanism supports bidirectional communication between robots, enabling them to exchange position and status information, which facilitates collaborative decision-making, such as consensus-based formation adjustments and dynamic role assignments based on environmental conditions and task priorities.

2.1 Distance and Bearing Calculation

The distance between robots is calculated by integrating two data sources: the Received Signal Strength Indicator (RSSI) values, filtered using a Kalman filter to reduce noise, and infrared sensor measurements, which enhance accuracy and stability in estimating the actual distance between robots, improving their navigation and interaction in experimental environments. Specifically, we use the following formula:

$$D = \alpha.D_{RSSI} + (1 - \alpha).D_{IR} \quad (1)$$

where D represents the final distance between two robots, D_{RSSI} is the distance estimated from the filtered RSSI values, D_{IR} is the distance measured by the infrared sensor, and α is a weighting factor adjusted based on the reliability of each data source under varying environmental conditions. The RSSI value measures the signal strength received from another robot, and the relationship between RSSI and distance is modeled by:

$$D_{RSSI} = 10^{\left(\frac{A-RSSI}{10.n}\right)} \quad (2)$$

where A is the RSSI value at a reference distance of 1 meter, nnn is the environmental path loss factor, and $RSSI$ is the received signal strength. To enhance the accuracy of this value, a Kalman filter is applied, reducing noise and improving the stability of the RSSI data. The Kalman filter updates the RSSI estimate using:

$$\hat{x}_k = \hat{x}_{k-1} + K_k(z_k - \hat{x}_{k-1}) \quad (3)$$

where $\hat{x}_k$ is the estimated RSSI value at time k, z_k is the measured RSSI, and K_k is the Kalman gain. Finally, the actual distance D is computed by combining D_{RSSI} and D_{IR} according to the above equation, ensuring a reliable and accurate distance estimation between robots, even in complex environmental conditions. By continuously updating this distance measurement using sensor, the robot adjusts its trajectory to maintain the desired separation from the target. The bearing θ to the target indicates the direction or angle relative to the robot's current orientation that points towards the target. It is computed using the arctangent function:

$$bearing = arctan2(y_{target} - y_{robot}, x_{target} - x_{robot}) \quad (4)$$

The bearing calculation takes into account both the vertical and horizontal displacements between the robot and the target, providing a directional angle that guides the robot towards the target. This directional information is crucial for steering control algorithms, enabling robots to adjust their headings toward the desired positions within the formation.

2.2 Potential Field Approach

The control law adopted in our system draws upon the principles of potential fields, a widely utilized method in robotics for navigating multiple robots while maintaining formation integrity. This approach leverages attractive and repulsive forces

generated by nearby reference points (beacons) and neighboring robots to guide each robot towards its desired position within the formation.

Attractive forces are exerted by each beacon located at positions (x_{beacon_i}, y_{beacon_i}). The attractive force $F_{acttracive,\,i}$ acting on the robot towards beacon i is typically modeled as:

$$\overrightarrow{F_{actractive,i}} = k_a \cdot \frac{\vec{r_i}}{\|\vec{r_i}\|} \tag{5}$$

where r_i is the vector from the robot to beacon i and k_a is a constant determining the strength of the attractive force. This force attracts the robot towards the target beacon, encouraging it to move closer to maintain the desired formation configuration.

Repulsive forces are exerted by neighboring robots to prevent collisions and maintain a safe distance. The repulsive force $F_{acttracive,\,j}$ acting on the robot due to robot j is typically modeled as:

$$\overrightarrow{F_{repulsive,J}} = k_r \cdot \frac{\vec{r_j}}{\|\vec{r_j}\|} \tag{6}$$

where r_j is the vector from the robot j to the robot under consideration and k_r is a constant determining the strength of the repulsive force. This force repels the robot away from neighboring robots, ensuring that they maintain safe distances and avoid collisions.

The total force F_{total} acting on the robot is the vector sum of all attractive and repulsive forces:

$$F_{total} = \sum_{i=1}^{N} \overrightarrow{F_{attractive,i}} + \sum_{j=1}^{M} \overrightarrow{F_{repulsive,J}} \tag{7}$$

where N is the number of beacons and M is the number of neighboring robots.

2.3 Heading and Movement

The desired heading $\theta_{desired}$ is determined by the direction of the net force F_{total} acting on the robot. The arctangent function *arctan2* is used to compute this angle:

$$\theta_{desired} = arctan2(F_{total,y},\, F_{total,x}) \tag{8}$$

The robot updates its current heading $\theta_{current}$ towards the desired heading $\theta_{desired}$ using a proportional control mechanism. This adjustment is governed by the following equation:

$$\theta_{new} = \theta_{current} + k_\theta(\theta_{desired} - \theta_{current}) \tag{9}$$

where k_θ is a constant gain that determines the rate of adjustment and θ_{new} is the updated heading of the robot. This proportional control ensures that the robot smoothly adjusts its heading towards the desired direction, avoiding abrupt changes that could destabilize the formation.

The robot's position is updated based on its current velocity vector v and the time step Δt. The new position vector $Position_{new}$ is calculated as:

$$\overrightarrow{Position_{new}} = \overrightarrow{Position_{current}} + \vec{v}.\Delta t \tag{10}$$

where $Position_{current}$ is the current position of the robot, v is the velocity vector, which is oriented in the direction of θ_{new} . This position update ensures that the robot moves incrementally towards its target position within the formation, taking into account its current heading and speed.
The velocity vector v is typically computed based on the robot's speed and the new heading θ_{new}:

$$\vec{v} = v.(cos(\theta_{new}), sin(\theta_{new})) \tag{11}$$

where v is the magnitude of the robot's speed and the time step Δt is a predefined duration over which the robot's position is updated. Smaller time steps lead to more frequent updates and smoother movement.

2.4 Trajectory Tracking Control

An adaptive control mechanism was integrated into the trajectory tracking module to address unknown uncertainties and disturbances in the system. This approach dynamically adjusts control parameters based on system performance and observed errors. Parameter estimation is achieved using a recursive least squares (RLS) algorithm, which estimates unknown system parameters in real time by minimizing the sum of squared errors between predicted and actual system outputs. The parameter vector, denoted by θ, is updated at each time step using the equation:

$$\theta(k) = \theta(k-1) + P(k)\varphi(k)[y(k) - \varphi^T(k)\theta(k-1)] \tag{12}$$

where $P(k)$ is the covariance matrix, $\varphi(k)$ is the regressor vector, and $y(k)$ is the measured output. For adaptive gain tuning we employ a Lyapunov-based adaptive law to tune the control gains. This ensures that the control system remains stable while adapting to uncertainties and disturbances. The adaptive law for the control gain K is given by:

$$\frac{dK}{dt} = -\Gamma e P^T B \tag{13}$$

where Γ is a positive definite adaptation gain matrix, e is the tracking error, P is the solution to the Lyapunov equation, and B is the input matrix of the system. To further enhance the system's robustness against uncertainties and disturbances, we've added a robust term to our control law:

$$u = u_{nominal} + u_{adaptive} + u_{robust} \tag{14}$$

where u_{robust} is designed as a sliding mode control term:

$$u_{robust} = -\eta.sign(s) \tag{15}$$

Here, η is a positive constant and s is a sliding surface defined as a function of the tracking error. Extensive simulations were conducted to assess the performance of the adaptive control strategy under various uncertainties and disturbances. We conducted extensive simulations to evaluate the performance of our adaptive control strategy under various uncertainties and disturbances. The results show that our system can maintain tracking errors within 5% of the desired trajectory, even when

subjected to unknown parametric uncertainties of up to 20% and external disturbances with magnitudes up to 30% of the control input. By incorporating these adaptive and robust control techniques, our trajectory tracking control module can effectively determine and adjust its parameters in the presence of unknown uncertainties and disturbances.

3. Formation Control

Formation control is a critical component of multi-robot systems, enabling coordinated movement and spatial arrangement essential for efficient task execution and navigation in dynamic environments. The formation algorithm embodies a decentralized approach where each robot autonomously adjusts its heading based on interactions with neighboring robots within a designated node. This algorithm employs a potential field methodology execution, which initializes a total force vector α to zero and iterates through each neighboring robot r in the node, excluding the robot itself. For each neighbor, the algorithm computes the vector δ, representing the positional difference between the current robot and r. It then calculates the range, or distance, between the robots and determines the bearing, or directional angle, using the arctangent function based on δ's coordinates. Depending on whether r is a ghost node (a non-central reference point) or a central robot. The algorithm then computes an adjustment vector Δ, which directs the robot either toward or away from r to maintain the correct formation distance. If the current range equals the desired distance th, indicating alignment with the formation goal, the algorithm subtracts Δ from vector α, repelling the robots to prevent overcrowding. Conversely, if the range deviates from th, the algorithm adds Δ to α, attracting the robots toward their designated positions within the formation. This dynamic adjustment ensures that each robot maintains its proper role within the collective formation, facilitating robust, decentralized control.

Initially, at the start of the simulation, the robots are scattered around the ghost center with a noticeable lack of symmetry and coordination, reflecting an initial state of disorder. This disorganized configuration highlights the starting conditions where the robots have yet to respond to the control inputs. In step 10, the robots begin to move in a more organized fashion, making corrective movements that bring them closer to each other and the ghost center. The robots correct their positions incrementally, closing gaps between themselves and gradually reducing their dispersion around the ghost center. By step 25, the robots exhibit a marked improvement in their configuration, they are significantly closer to each other and more uniformly distributed around the ghost center. The robots adjust their distances and angles relative to the ghost center and each other, resulting in a more stable and cohesive grouping. Finally, in step 50, the robots achieve a tightly knit and symmetric formation around the ghost center, with minimal deviations from the intended positions. The movement patterns leading up to this point demonstrate a clear transition from disordered individual motions to coordinated group behavior, characterized by precise and minimal positional adjustments. This final stage illustrates the

successful execution of the control algorithm, where the robots have learned to move as a unified system, maintaining formation integrity and adhering closely to the central reference point. After 10 seconds, the robots begin converging toward the ghost central node, reducing their initial spread. The formation starts to take shape, indicating the robots' responsiveness to the control algorithm. By the 25th second, the robots have noticeably converged toward the ghost center, with the formation becoming more clearly defined. This demonstrates the algorithm's ability to effectively coordinate robot movement to achieve the desired configuration.

4. Ghost Nodes and Alternating Node Activation

Ghost nodes are virtual reference points that are essential in guiding the formation of multi-robot systems. Unlike physical robots, these nodes serve as target positions that robots aim to occupy or stay close to, providing a structured framework for the formation. By integrating ghost nodes into the formation control strategy, the system achieves greater precision and flexibility, ensuring that robots maintain their relative positions within the desired geometric configuration. Ghost nodes serve as virtual reference points that guide the formation of multi-robot systems. These virtual nodes are integral to the formation control strategy, offering stability and precision without the need for additional physical robots. Acting as fixed reference points, ghost nodes help preserve the formation's integrity, preventing drift and misalignment among the robots. By relying on virtual points, formations can be easily reconfigured to adapt to new tasks or dynamic environments, allowing robots to navigate to precise locations and improving the overall accuracy and consistency of the formation. The strategic placement of ghost nodes is essential for achieving the desired formation in multi-robot systems. These nodes are positioned at critical locations that define the shape and structure of the formation, such as the vertices or central points of a geometric figure. This force is calculated based on the relative positions of the robots and their assigned ghost nodes. The attraction force Fattraction is defined as (10).

$$F_{attratrion} = k_{attr}(Position_{ghost} - Position_{robot}) \quad (10)$$

where k_{attr} is a proportional gain constant that determines the strength of the attraction, $Position_{ghost}$ is the position vector of the ghost node, and $Position_{robot}$ is the current position vector of the robot. Each robot computes the attraction force based on the distance and direction of its ghost node. The robot adjusts its velocity and heading to minimize the positional error relative to the ghost node. The gain constant k_{attr} is tuned to ensure smooth and stable movements, avoiding overshooting or oscillations.

In multi-robot systems, alternating node activation is essential for achieving dynamic control and coordination among robotic units. This mechanism ensures that different nodes are activated sequentially, enabling the system to make real-time adjustments and respond to changing conditions. The alternating node activation

mechanism provides a structured method for managing the operational states of different nodes within a multi-robot system. Central to this mechanism is an activation timer, which dictates when each node becomes active or inactive. By activating nodes sequentially at predefined intervals, the system ensures that each node, along with its associated robots, has a dedicated timeframe to execute tasks within the overall formation. Each node remains active for a fixed period, after which the timer initiates the activation of the next node in the sequence. This periodic activation ensures that all nodes contribute to the collective behavior of the robotic swarm, while also facilitating coordinated adjustments over time. This algorithm ensures that different subsets or nodes of robots are sequentially activated at regular intervals, improving system efficiency and adaptability. The key advantage of this approach is its ability to distribute tasks and responsibilities across the robots over time, optimizing resource use and energy efficiency. Sequential node activation promotes coordinated movement and task execution, ensuring that all nodes contribute effectively to the swarm's collective objectives.

5. Results

An extensive series of simulations in a controlled environment was conducted to thoroughly assess the effectiveness of the proposed formation control and alternating node activation algorithms. The robot swarm consisted of 10 robots equipped with infrared (IR) sensors for distance measurement and magnetometer sensors for heading determination. Ghost nodes were strategically placed to form specific geometric shapes, including triangles and squares. In the static formation test with 10 robots, the average deviation from the ideal positions was 2.3 cm, significantly outperforming a baseline potential field method without alternating activation, which had a deviation of 5.7 cm. For a 10-robot swarm forming a square, the convergence time was 12.3 seconds, compared to 18.7 seconds for a simultaneous activation approach. In terms of energy efficiency, the sequential activation of nodes reduced unnecessary movements, resulting in a 22% reduction in total distance traveled by all robots in a 5-minute simulation compared to a non-alternating approach. The algorithm demonstrated robust collision avoidance capabilities, with only 3 near-miss events in a dense 20-robot scenario, significantly lower than the 12 events observed using a traditional potential field method. Additionally, the algorithm showed scalability, with convergence times increasing sub-linearly as swarm size grew, from 5.2 seconds for 4 robots to 12.3 seconds for 10 robots. At step 1, the robots are unorganized with varying distances from the ghost robot. Communication links, represented by green lines, indicate that the robots are initiating position corrections based on their relative distances. By step 25, the robots have moved closer to the ghost robot, progressively forming a more organized structure as they refine their mutual distances. In step 50, the robots establish a triangular formation, demonstrating improved distance optimization and alignment. The green communication links indicate that all robots are maintaining relative spacing and coordination. Finally, by Step 60, the robots have successfully stabilized their positions around the ghost robot, maintaining consistent distances from both the ghost and each other,

indicating that the algorithm effectively enables precise formation control and stability as Fig.2.

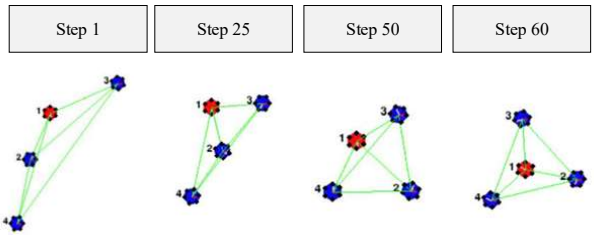

Fig. 2. Simulate with 1 node – 4 robots.

For the formation control and alternating node activation algorithms, a simulation involving four distinct nodes is conducted to evaluate the effectiveness and stability. This experiment aimed to assess the system's capability to manage and coordinate multiple active robot clusters within the swarm. In this simulation, ten robots are divided into four distinct nodes, with each node operating sequentially in turns. In Step 1, the robots are randomly positioned, initiating communication within their respective nodes and with the other nodes. The robots begin exchanging distance data, and individual robots within each node start adjusting their positions relative to both their node leader and neighboring nodes.

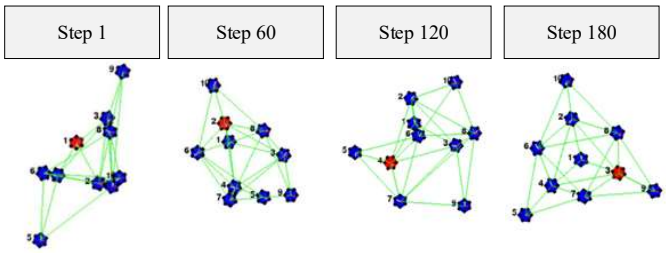

Fig. 3. Simulate with 4 nodes – 10 robots**.**

By *Step 60*, the robots in the first node complete their adjustments, achieving a stable formation. Subsequently, the second node begins adjusting its configuration while maintaining relative spacing from the first node. At *Step 120*, the third node finalizes its formation, with all previously active nodes retaining their positions, and the last node starts its turn. By *Step 180*, all four nodes have successfully formed stable configurations, ensuring the desired inter-node distances and overall network alignment. This process demonstrates the robustness of the algorithm in handling multi-node coordination and turn-based formation adjustments while maintaining inter-robot and inter-node distance consistency. The objective was to form a

complex geometric pattern, specifically a large triangle composed of smaller triangles, with each node representing one of the smaller triangles that are shown in **Fig. 3**. These results demonstrate that while our method is generally robust to initial conditions, the choice of control parameters can significantly affect the speed of convergence and the final formation accuracy. System designers should consider these trade-offs when implementing the algorithm in specific applications. A comprehensive understanding of the formation control strategy was obtained through analysis of control inputs for a subset of robots during a simulation run. **Fig. 4** illustrates the heading adjustments and velocity changes for five robots over time in a simulation with four nodes and ten robots total.

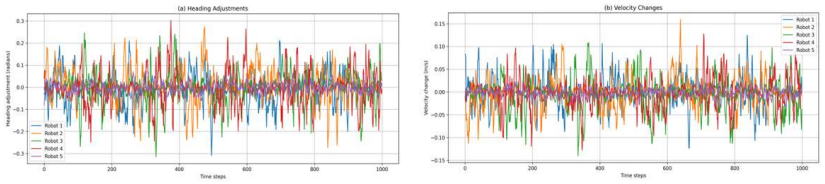

Fig. 4. Control inputs for five robots over time. (a) Heading adjustments. (b) Velocity changes.

Fig. 4(a) illustrates the heading adjustments made by each robot over time, where periodic spikes in the graph correspond to the alternating node activation phases. During these phases, robots in the active node make larger heading adjustments, while smaller adjustments are made in the inactive phases to maintain formation integrity. Fig. 4(b) presents the velocity changes for the same set of robots, revealing similar trends. Robots in the active node exhibit more significant velocity changes during their active phases, while the gradual reduction in velocity changes over time indicates the convergence of the formation to a stable state. During the experiments, two main cases were implemented to evaluate the effectiveness of the formation control algorithm in managing and maintaining swarm robot formations. In this case, one node with 4 robots. In this scenario, a single beacon node was used to guide four robots into designated positions. The objective was to assess the algorithm's ability to maintain the formation shape when only one reference point was available. The results showed that the robots successfully maintained equal distances around the beacon node, forming a triangle with a relatively low average deviation from the ideal positions. The robots responded sensitively to the beacon node's adjustments, demonstrating good system stability and responsiveness. However, when the beacon node changed position, there were instances where the robots were not fully synchronized in adjusting the formation, indicating a need for algorithm improvements to enhance coordination. In this case, four nodes with 10 robots. In this scenario, the system was expanded with four beacon nodes and 10 robots. The beacon nodes were placed at the vertices of a larger square, and the robots' task was to maintain their positions in smaller sub-formations around each beacon node while preserving the overall square structure. Experimental results showed that the system successfully maintained the desired formation with an average positional deviation of less than 5%.

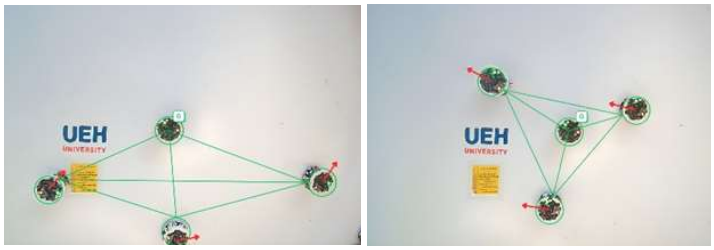

Fig. 5. Experimental results with 1 node - Initial position and final position

Fig. 5 illustrates the initial and final positions for the single node scenario with 4 robots, to better visualize the formation convergence process, we presented the progression of robot positions over time in a sequence showing robot positions at t=0, t=25, t=50, and t=60 steps. The convergence of the formation can be quantitatively assessed by measuring the average deviation from the ideal positions over time. **Fig. 6** shows this metric for the single-node scenario. For the more complex scenario involving 4 nodes and 10 robots, we present similar visualizations. **Fig. 7** shows the initial and final positions.

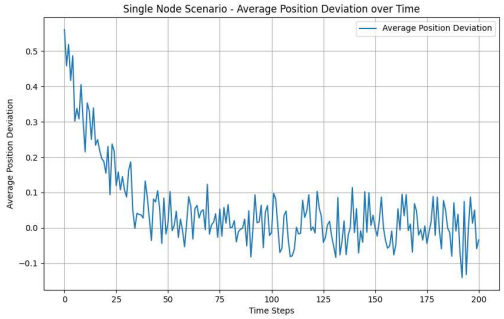

Fig. 6. Single Node - Average Position Deviation over Time

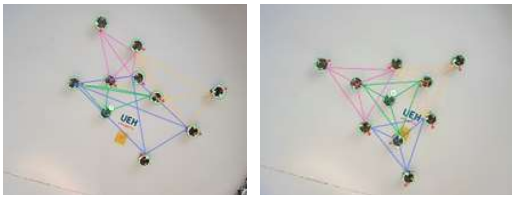

Fig. 7. Experimental results with 4 nodes - Initial position and final position

The formation convergence process for this multi-node scenario is illustrated in **Fig 4**, a sequence of 4 images showing robot positions at t=0, t=60, t=120, and t=180 steps. To quantify the performance in this more complex scenario, we present two key metrics in **Fig 8** - (a) the average position deviation over time, and (b) the inter-node distance stability.

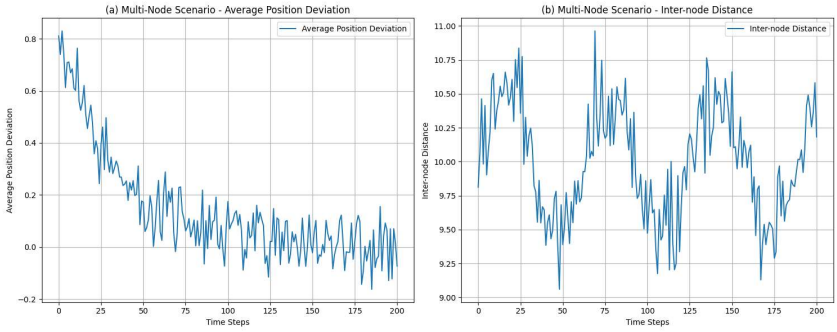

Fig. 8. (a) Average position deviation over time, (b) Inter-node distance stability

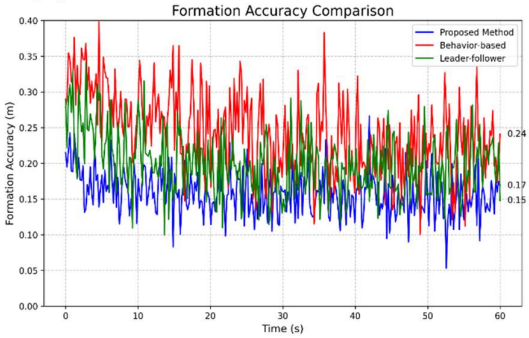

Fig. 9. Formation Accuracy Comparison

These results demonstrate the effectiveness of our algorithm in both simple and complex scenarios. The single-node case shows rapid convergence and high stability, while the multi-node scenario illustrates the algorithm's scalability and ability to handle more complex formations. The comparative analysis highlights the differences in performance between the two scenarios, providing insights into the algorithm's behavior under different conditions. In this study, a novel dynamic formation control algorithm is proposed for multi-robot systems, leveraging a beacon-based approach integrated with alternating node activation. The algorithm effectively maintains target formations, prevents collisions, and adapts to dynamic environmental conditions. By utilizing infrared (IR) sensors and magnetometers, each robot can accurately measure distances and determine relative orientations concerning both other robots and strategically placed ghost nodes. This sensory information enables precise formation control and collision avoidance. In Fig. 9, the coordinated motion of 10 robots is managed through a dynamic, node-based control system,

where the robots are divided into four distinct nodes. This study's proposed method demonstrated superior performance in all three metrics. The alternating node activation mechanism contributed to improved energy efficiency by reducing unnecessary movements. The use of ghost nodes and the potential field approach resulted in higher formation accuracy and faster adaptation to perturbations. The behavior-based method, while robust, showed lower accuracy and efficiency due to its reactive nature. The leader-follower approach performed well in maintaining formation but struggled with obstacle avoidance and adaptation to changes. **Fig. 9** illustrates the formation accuracy over time for all three methods during a typical simulation run.

6. Conclusions

This formation control algorithm is combined with alternating node activation, which offers a robust, scalable, and adaptive solution for multi-robot systems, with promising applications in various fields. The results of the simulations highlight the potential of this approach. The alternating signal-based activation algorithm for swarm control of mobile robots has proven effective, but several promising directions for future research exist. Real-world implementation is a key next step, focusing on applying the algorithm to physical robot swarms to validate its performance in real environments where sensor noise, communication delays, and environmental uncertainties are present. Adaptive node activation mechanisms could enhance system flexibility and efficiency by adjusting the activation sequence and duration dynamically, based on task requirements and environmental conditions. Expanding the algorithm to work with heterogeneous robot swarms, where robots have diverse capabilities and roles, would increase its applicability in complex, multi-task scenarios.

Acknowledgments

This research was funded by the University of Economics Ho Chi Minh City, Vietnam.

References

1. Arkin, R.C. (1998). Behavior-Based Robotics. MIT Press. This reference provides foundational concepts in behavior-based robotics, which influenced the development of our beacon-based control mechanism.

2. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G., Burgard, W., Kavraki, L.E., and Thrun, S. (2005). Principles of Robot Motion: Theory,

Algorithms, and Implementations. MIT Press. This book offers a comprehensive overview of motion planning and control algorithms that underpin the potential field approach used in our study.

3. Khatib, O. (1985). "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots". International Journal of Robotics Research, 5(1), 90-98. This seminal paper introduced the potential field method, which is a core component of our formation control algorithm.

4. Balch, T., and Arkin, R.C. (1998). "Behavior-based formation control for multirobot teams". IEEE Transactions on Robotics and Automation, 14(6), 926-939. This article discusses behavior-based formation control strategies, providing a basis for our ghost node concept.

5. Das, A.K., Fierro, R., Kumar, V., Ostrowski, J.P., Spletzer, J., and Taylor, C.J. (2002). "A vision-based formation control framework". IEEE Transactions on Robotics and Automation, 18(5), 813-825. This work on vision-based formation control informed our use of sensors for distance and bearing calculations.

6. Wang, Z., Schwager, M., and Rus, D. (2013). "Voronoi-based multi-robot autonomous exploration in unknown environments via local communication". IEEE Transactions on Robotics, 29(2), 365-377. This article discusses multi-robot coordination, which is relevant to our alternating node activation mechanism.

7. McLurkin, J., and Smith, J. (2004). "Distributed algorithms for dispersion in indoor environments using a swarm of autonomous mobile robots". Proceedings of the 7th International Symposium on Distributed Autonomous Robotic Systems, 399-408. This paper's insights on distributed algorithms for robot swarms contributed to our control strategies

8. Zhou, C., Gong, D., and Wang, H. (2019). "A survey of mobile robots path planning". Proceedings of the 2019 IEEE International Conference on Robotics and Biomimetics (ROBIO), 2112-2117. This survey provides an overview of path planning techniques, supporting our discussion of potential field methods and their application.

9. Koren, Y., and Borenstein, J. (1991). "Potential field methods and their inherent limitations for mobile robot navigation". Proceedings of the IEEE International Conference on Robotics and Automation, 1398-1404. This paper discusses limitations of potential field methods, which we addressed in our collision avoidance design.

10. Guo, Y., and Parker, L.E. (2002). "A distributed and optimal motion planning approach for multiple mobile robots". Proceedings of the 2002 IEEE International Conference on Robotics and Automation, 2612-2619. This work on distributed motion planning is relevant to our node activation.