JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com



Decentralized Collision Avoidance and Formation Recovery for Multi-Robot Systems in Dynamic Environments: An AnyLogic-Based Evaluation

Kenneth Kayock Attans, Ahmed Abubakar Aliyu, Gabriel Lazarus Dams, Nehemiah Musa, Rimamtari David Nyatse
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria

ABSTRACT
Multi-agent robots can improve robustness, redundancy, and mission efficiency, but coordinated navigation becomes difficult when robots must preserve a formation while responding to static and moving obstacles. This study presents a decentralized framework for collision avoidance and formation control in a dynamic environment. The framework uses a leader–follower V-shaped formation in which the leader plans a route to the target with the A algorithm, while each robot independently maintains its assigned bearing, spacing, and velocity relative to a preceding robot. Potential collisions are detected using Minimum Predicted Distance (MPD), and collision-risk regions are handled using a Minimum Enclosing Rectangle (MER) maneuver. Following avoidance, affected followers adjust speed, bearing, and spacing to return to the prescribed formation using the Proportional Integral Derivative (PID) control. The framework was implemented and evaluated in AnyLogic using an agent-based simulation containing 7, 10, 20, and 40 static and dynamic obstacles. Centralized and decentralized control were compared using mission duration, collision rate, mean formation error, path efficiency, obstacle-avoidance time, and command count. The results indicate that centralized control can be advantageous for smaller formations, whereas decentralized control becomes more scalable as the number of robots increases. The proposed architecture removes dependence on a central controller, reduces communication hops, and supports autonomous local responses in dynamic environments.*

ARTICLE INFO
Article History
Received: April, 2026
Received in revised form: May, 2026
Accepted: August, 2026
Published online: September, 2026

KEYWORDS
Multi-Agent Systems, Mobile Robots, Collision Avoidance, Formation Control, Decentralized Control, Dynamic Environment

INTRODUCTION

Multi-agent systems (MASs) have been investigated since the 1980s and are widely regarded as more robust and efficient than single-agent systems (Medhi et al., 2023; Shi et al., 2024). As a subfield of Artificial Intelligence, MASs provide a foundation for constructing complex systems in which sub-problems that satisfy constraints can be delegated to different problem-solving agents, each with its own local interests and goals. Compared with single-agent systems, multi-agent systems offer greater flexibility, stronger adaptation capacity, and higher reliability in mission completion, which explains their adoption in complex and hazardous environments (Chang et al., 2025; Lei & Pei, 2025). Cooperation among multiple agents also yields advantages associated with strength in numbers, including improved efficiency, robustness, redundancy, flexibility, and cost reduction.

In recent years, researchers have extensively studied multi-agent robotic systems across disciplines such as physics, biology, cybernetics, and automatic control. Substantial work has been conducted separately on formation control (Butler et al., 2024; Chang et al., 2025; Solano-Castellanos et al., 2025; Pham & Nguyen, 2023) and on collision avoidance (Dong et al.,

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

2024; Martinez-Baselga et al., 2026; Vielmetti et al., 2026; Wang & Zhao, 2024; Xiong et al., 2023), with applications proposed in diverse domains. Recent studies increasingly address collision avoidance and formation control as core challenges in autonomous robotics, particularly for Unmanned Surface Vehicles (USVs). A collision occurs when vehicles attempt to occupy the same coordinate or node simultaneously, or when a robot encounters a static obstacle.

To prevent this, three primary strategies are used:

1. Head-On Avoidance (*Figure 1 Scenario a*)**:** When two robots approach each other at a 180° angular distance, each swerve to the right or left.
2. Static Obstacle Avoidance (*Figure 1 Scenario b*)**:** When a robot approaches a stationary obstacle, only that robot swerves to bypass it.
3. Junction Negotiation **(***Figure 1 Scenario c***):** When moving paths cross, robots negotiate right-of-way to determine who passes first.

The third negotiation method is unsuitable for head-on (180°) collisions because both robots would remain on each other's intersecting paths despite negotiating. Therefore, strategy selection depends entirely on the encounter geometry.

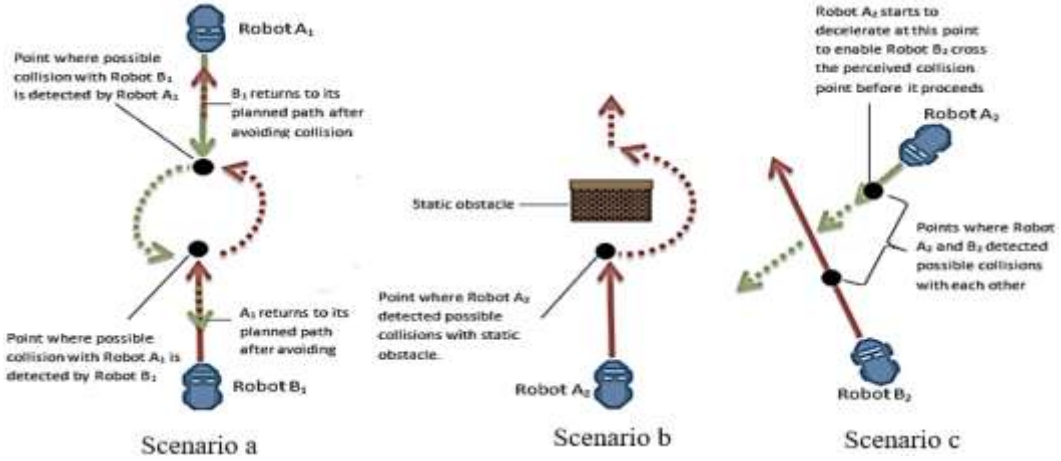


Figure 1: Collision avoidance scenarios.

RELATED WORK

Research on multi-robot formation control centers on the coupled challenge of maintaining group coordination while avoiding static and moving obstacles under real-world constraints such as limited communication, state uncertainty, and actuator limits. Modern approaches seek to balance formation accuracy and safe motion using Velocity Obstacles, Control Barrier Functions (CBFs), artificial potential fields, decentralized estimators, reinforcement learning, and fixed-time control.

Fan et al. (2024) combined virtual-spring forces with an improved Velocity Obstacle-based (VO) method, demonstrating that dynamic conflicts severely increase maximum position errors in larger formations (e.g., 20 robots). Roncero et al. (2025) integrated velocity obstacles and control barrier functions into a unified quadratic programming framework for multi-agent collision avoidance (Roncero et al., 2025). The approach handles second-order integrator and car-like dynamics while guaranteeing safety bounds, aligning closely with the combined VO-CBF methods discussed in Fan et al. (2024). Similarly, Estevez et al. (2025) extended Hybrid Reciprocal Velocity Obstacles (HRVO) for paired agents and triads, achieving smooth, collision-free navigation, though validation was limited to small simulation groups. Butler et al. (2024) developed a collaborative Control Barrier Functions (CBFs) safety filter for tree-structured networks, while Wang et al. (2025) integrated distributed

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

observers with relaxation parameters to handle input saturation and pose uncertainties. While principled, these methods depend heavily on reliable state estimation and feasible constraint resolution.

Bohara and Farzan (2025) proposed a decentralized, CBF-based controller that estimates neighbor velocities under communication limits. Liu et al. (2023) combined the virtual structure method for precise multi-USV formation geometry maintenance with an artificial potential field for real-time obstacle and collision avoidance, offering the strength of high trajectory accuracy and robust cooperative navigation, but remaining limited by the classic artificial potential field drawback of local minima entrapment, where vehicles can become stuck near obstacles. Comparing architectures, Jamshidpey et al. (2024) found that centralized multi-robot sweep coverage excelled in completion time and uniformity, whereas decentralized systems favored scalability, fault tolerance, and local autonomy.

Xie et al. (2024) demonstrated a two-stage reinforcement learning pipeline for multi-UAV obstacle evasion that outperformed standard baselines. To handle unknown disturbances, Xue et al. (2026) combined an extended-state observer, fixed-time strategies, and event-triggered updates to minimize communication overhead and manage high initial control speeds. Despite significant advancements, existing multi-agent collision and formation control methods predominantly rely on centralized architectures or static environment assumptions. This creates vulnerabilities such as single points of failure, high communication overhead, and limited adaptability. Furthermore, current decentralized techniques frequently struggle to handle unpredictable dynamic obstacles while simultaneously maintaining strict spatial formations during real-time evasion without external intervention.

Problem Definition

Collision avoidance and formation control are coupled problems. A robot that deviates to avoid an obstacle changes its position, bearing, and distance relative to neighboring robots. Unless these variables are subsequently corrected, the formation may become unstable or fail to complete the mission as a coordinated group. The problem is more demanding in dynamic environments, where obstacles can move unpredictably and where a robot cannot assume that other agents will negotiate or communicate before changing their trajectories.

Conventional centralized architectures allow a controller to collect information, compute decisions, and distribute commands. Although this can simplify coordination, it introduces communication overhead and a single point of failure. A decentralized architecture instead equips each robot with local sensing and decision-making capabilities. The central research problem addressed in this study is therefore how a group of robots can plan path to a target, form a V-shaped arrangement (geometric pattern/formation), detect static or moving obstacles, avoid collisions autonomously, and return to the formation without depending on a central controller, all along considering the dynamic nature of the environment, where static and dynamic obstacles exist.

METHODOLOGY

In the proposed approach, a robot agent at any time instance is either a leader or a follower. As a leader, it will utilize path planning, collision detection, and collision avoidance, without a need for formation control. As a follower, it utilizes formation control, collision detection, and collision avoidance, without a need for path planning.

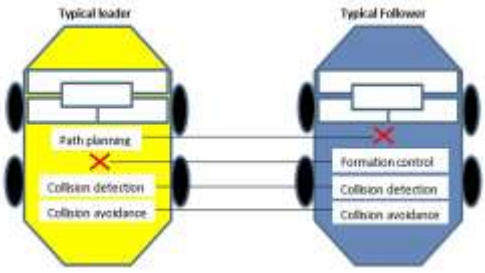


Figure 2. Instances of actions performed by a typical leader and a typical follower in the formation during movement.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

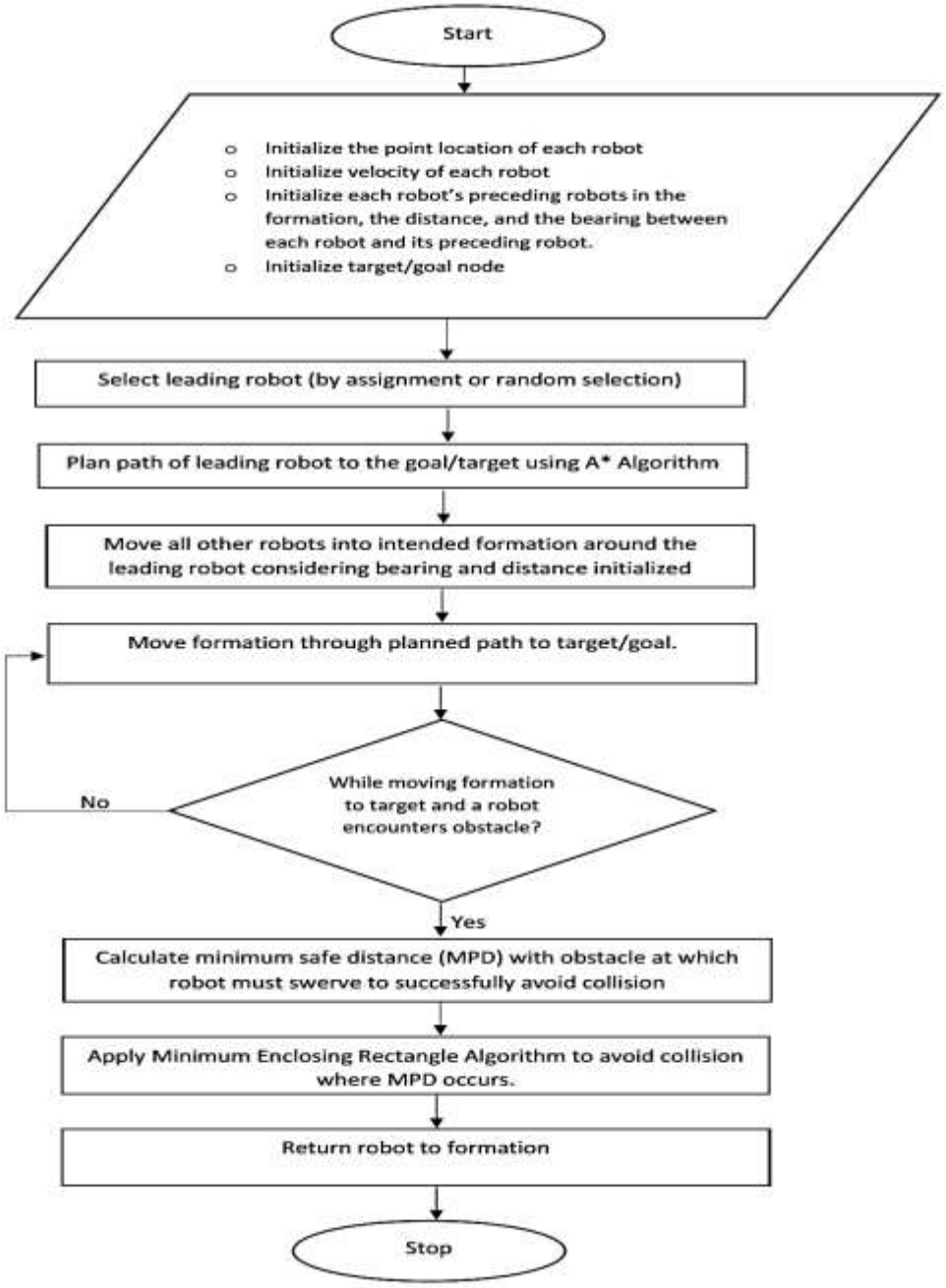


Figure 3. Overall model of the proposed system

It is noteworthy that although all the robots are loaded with the path planning algorithm, only the leading robot uses it at a time. If the leading robot is incapacitated, the next designated leader takes over and plans the best path to the target/goal. Below is the control instance for each robot that makes up the system.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

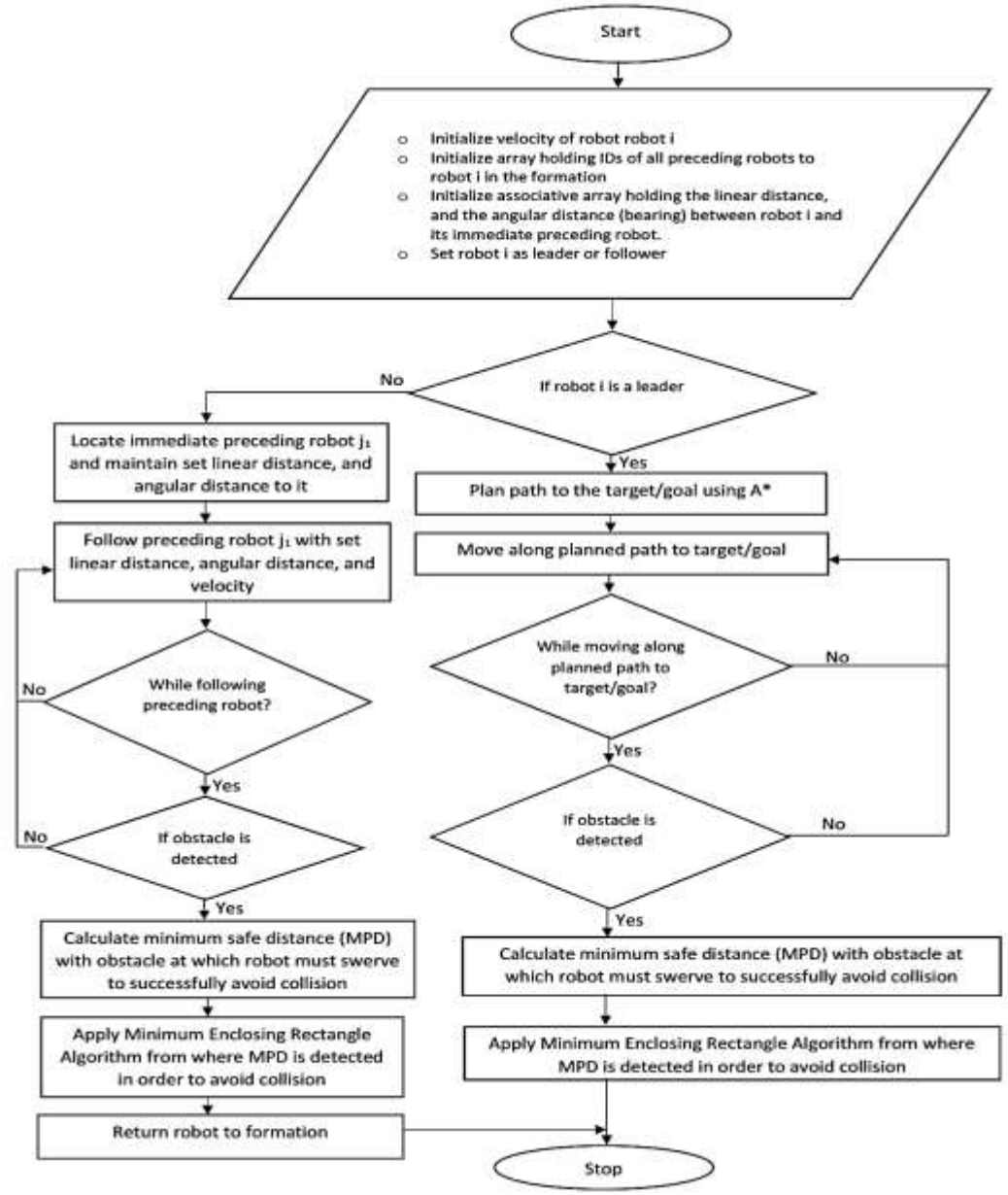


Figure 4. Model of collision avoidance and formation control in each robot

Path Planning Using the A* Algorithm

We used the A* algorithm (pronounced A-star), an extension of Dijkstra algorithm, and a well-known pathfinding algorithm, to get the shortest path from the leader's start location to its target. The algorithm, as explained in *Figure 4.4,* works by traversing nodes that lead from a robot's current position to its target with the goal of finding the shortest path. It uses a heuristic to improve performance with respect to time. The algorithm has an evaluation function given as:

$f(n) = g(n) + h(n)$

Where

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

g(n) = is the cost of the best path currently known from the start node to node n (which is the distance from start position to the current expanded node),

h(n) = is a heuristic estimated cost to reach from this current node to the goal node or target, and ***f(n)*** = sum of *g(n)* and *h(n)*.

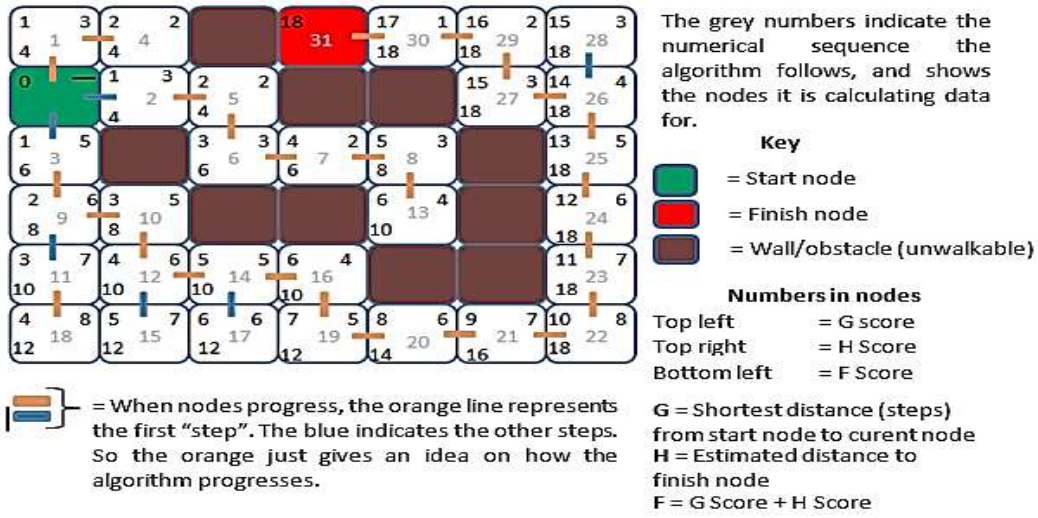


Figure 5: A grid showing how a robot uses A* algorithm to plan its path to a target in an environment with obstacles.

The A* is used to estimate the lowest-cost path from the initial position to the goal/target position. The cost so far, *G,* and the Heuristic cost ahead, *H,* are needed to calculate the total cost *F*, Where F = G + H

To approximate the value of H, the Euclidean Distance is used. The Euclidean Distance would be calculated between the start cell and the goal/target cell using the distance formula as shown below.

$H = \sqrt{((current_cell.x - goal.x)^2 + (current_cell.y - goal.y)^2)}$

Collision Detection and Avoidance

Each robot uses MPD to determine whether a collision is likely assuming it maintains its current trajectory. MPD represents the minimum future separation predicted between two moving entities. A robot begins an avoidance maneuver when the predicted separation falls below the permitted safety distance.

Let $r = P_o - P_r$ be the relative position vector between the obstacle P_o and robot P_r

Let $V_{rel} = V_o - V_r$ be the relative velocity vector.

Assuming constant velocity over prediction horizon T, the time to minimum distance t^* is:

$$t^* = max\left(0, min\left(-\frac{r.Vrel}{||Vrel||^2}, T\right)\right)$$

The predicted minimum distance d_{min} is therefore given as:

$d_{min} = ||r + t^*V_{rel}||$

An avoidance maneuver is triggered when $d_{min} < d_{safe}$ and $t^* > 0$.

Joining and Maintaining Formation

To join and maintain a formation, a robot maintains the angular distance and linear distance it received during initialization. Constant speed is also maintained if the assigned distance is observed to be achieved; otherwise, a robot adjusts its speed in order to maintain its assigned distance to its preceding robot.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

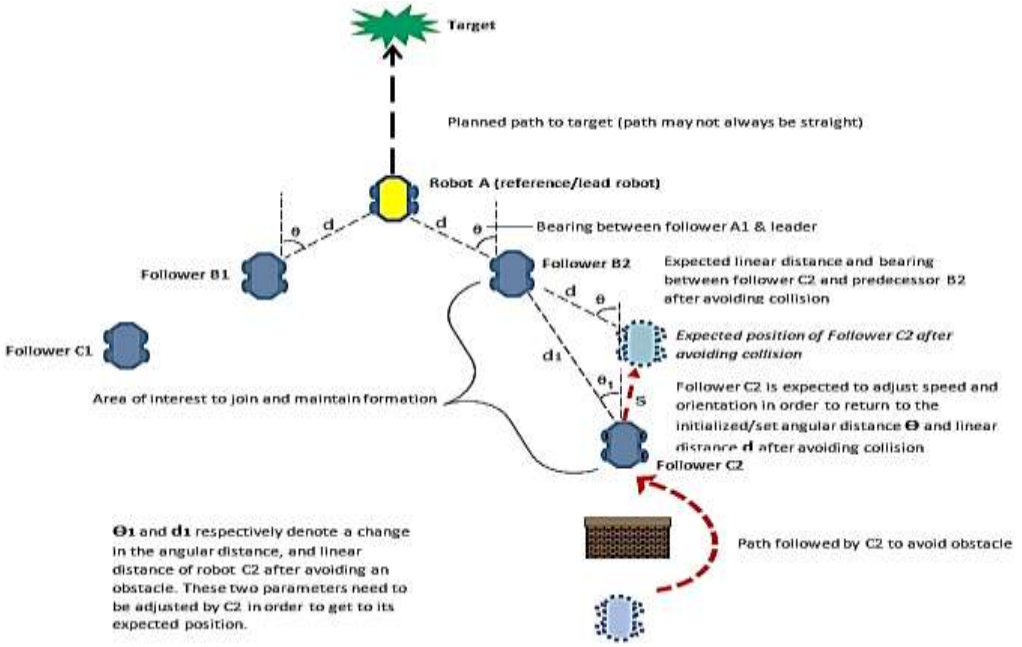


Figure 6. Altered Angular distance **Θ** and linear distance **d** in a formation due to the avoidance of an obstacle

Maintaining a formation requires robots to keep constant bearing, speed, and distance relative to one another. Even small changes in any of these variables necessitate corresponding adjustments by dependent robots, so formation maintenance depends on keeping bearing, speed, and distance constant across the group.

From *figure 4.6* above, after avoiding collision, follower C2 would need to continuously adjust d_1 and Θ_1 in order to respectively attain the desired distance ***d*** and bearing Θ to the preceding follower B2.

Furthermore, C2 would need to accelerate in order to be able to attain and maitain its desired speed ***v***, distance ***d,*** and bearing Θ to its immediately preceding robot B2. This is achieved using the Proportional Integral Derivative (PID) equation to tune each of the three variables *v*, *d*, and Θ. The PID equation is stated below as:

$$u(t) = K_p.e(t) + K_i.\int_0^t e(\tau)d\tau + K_d.\frac{de(t)}{dt}$$

Where:

u(t) = control output of the manipulated variable

K_p = proportional gain which determines the proportion of the error that contributes to the control output. It scales the error term e(t).

K_i = integral gain, which accounts for the accumulation of past errors over time. It helps eliminate steady-state errors by integrating the error term.

K_d = derivative gain, which considers the rate of change of the error. It helps dampen oscillations and reduce overshoot by taking into account how fast the error is changing.

e(t) = error at time t, calculated as the difference between the setpoint and the process variable: e(t) = setpoint – process variable. It is the error at time t calculated as the difference between the desired position and the current position

$\int_0^t e(\tau)d\tau$ = integral of the error over time, which helps in eliminating steady-state errors

$\frac{de(t)}{dt}$ = derivative term, which is the rate of change of the error with respect to time, and which helps in predicting the future error and dampening oscillations

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

The PID controller continuously computes the control output for each of the three variables of interest: speed *v*, distance *d*, and bearing *θ*. It computes each variable based on the current error *e(t)*, the integral of past errors, and the rate of change of the error. By continuously tuning the three gains (K_p, K_i, and K_d), the controller adjusts its response to achieve the desired speed, distance, and bearing.

The formation equation is given as:

For robot i,

Right side:

$x_i = x_L - di\cos(45^o)$, $yi = y_L - di\sin(45^o)$

Left side:

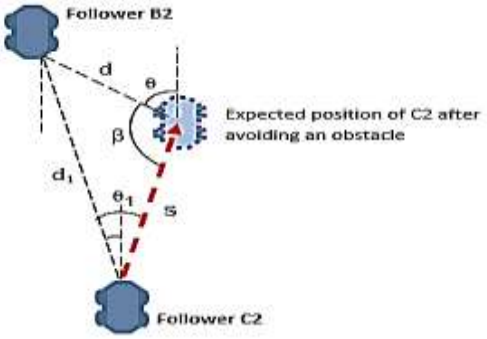
$x_i = x_L - di\cos(45^o)$, $yi = y_L + di\sin(45^o)$

Where:

x_i, y_L = leader position

d = spacing distance

Taking out the area of interest for formation control from *Figure 4.6* illustrates the linear distance, bearing, and speed that needs to be adjusted for a robot to return to its expected position in the moving formation.



d = initialized distance to be maintained by all robots in the formation

Θ = expected angle to be maintained by all robots in the formation

s = disance to be traveled by robot C2 after avoiding collision in order to reach its expected position in the formation

θ_1 = angle between a robot and its follower after avoiding collision

Figure 7: An excerpt from Figure 6

Experimental Results and Discussion

Evaluation metrics such as formation error, collision rate, path efficiency, obstacle avoidance, commands, and mission time are calculated for each robot agent, with average values displayed in real time on a time plot graph

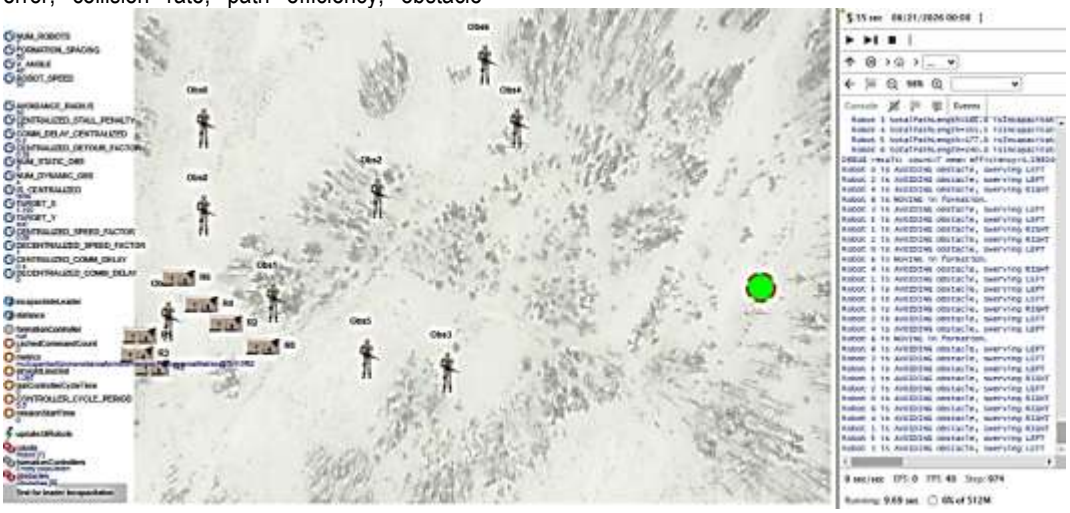


Figure 8: At initialization showing robot agents gradually making a formation around the leader. The right panel shows the activities of the robots during the experiment. Target in green.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com



Figure 9: The right panel shows the activities of robots during the experiment.

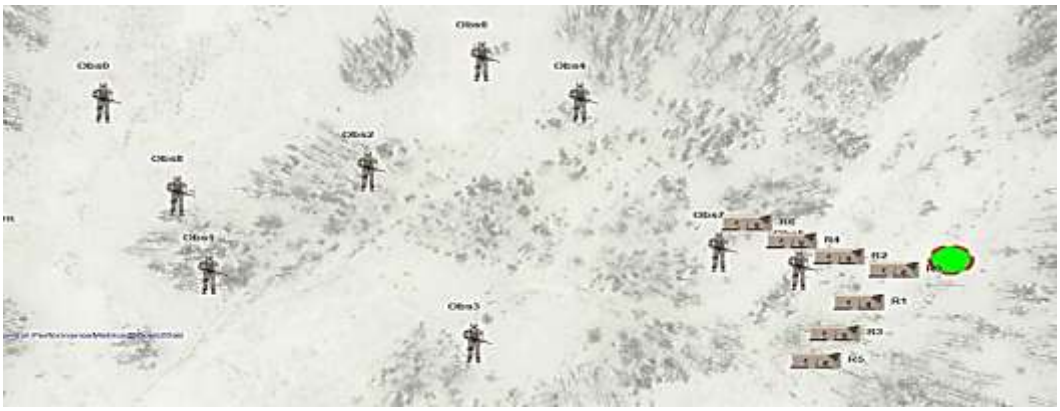


Figure 10: Arrival of the formation at the target

Time Plot for Performance Metrics

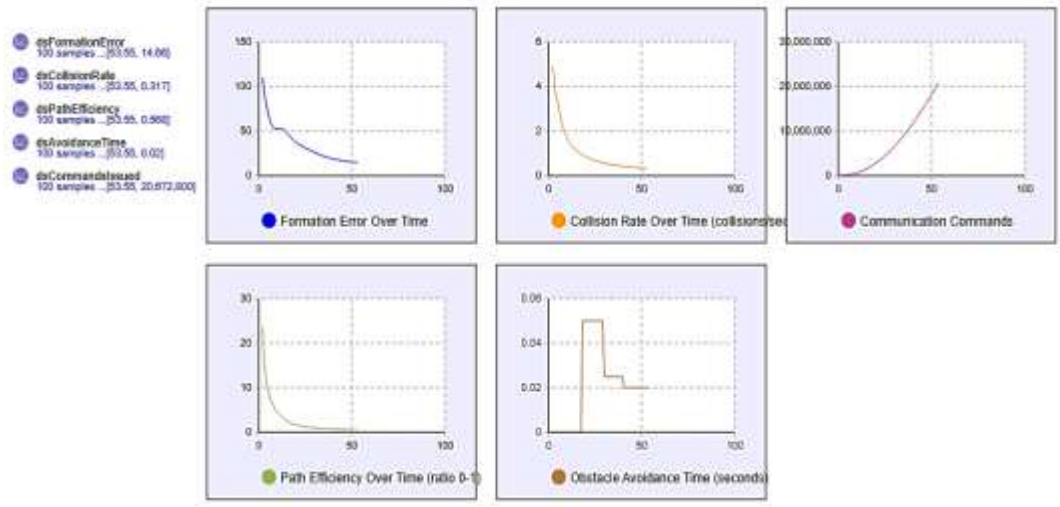


Figure 11: Time plot for performance metrics using a central controller to manage collision avoidance and formation control

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com

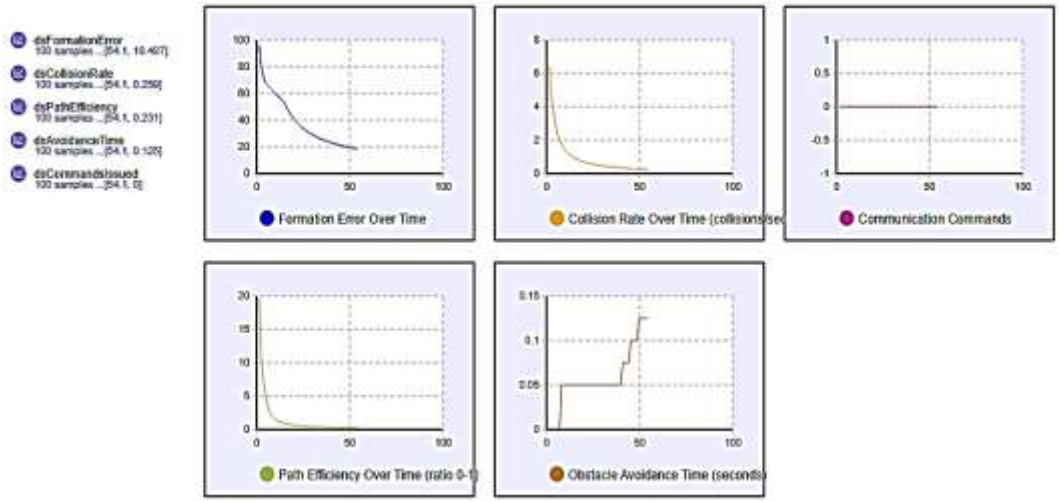


Figure 12: Time plot for performance metrics using a decentralized control

Performance metrics for experiment with centralized control

Table 1.2.1: Performance metrics for experiment with centralized control (20 robots used)

```
====== PERFORMANCE REPORT ======
Mode              : CENTRALIZED
Mission Start     : 0.0000 s
Mission Complete  : 1076.7000 s
Mission Duration  : 1076.7000 s
Collision Rate    : 0.0901 /s
Mean Form. Error  : 69.40 px
Path Efficiency   : 0.040
Avg Avoid. Time   : 0.87 s
Commands Issued   : 2102348647 (centralized ctrl)
================================
```

Table 1.2.2: Performance metrics for experiment with centralized control (40 robots used)

```
====== PERFORMANCE REPORT ======
Mode              : CENTRALIZED
Mission Start     : 0.0000 s
Mission Complete  : 1029.7500 s
Mission Duration  : 1029.7500 s
Collision Rate    : 3.3010 /s
Mean Form. Error  : 156.95 px
Path Efficiency   : 0.045
Avg Avoid. Time   : 3.75 s
Commands Issued   : 1867176738 (centralized ctrl)
================================
```

Performance metrics for experiment with decentralized control

Table 1.3.1: Performance metrics for experiment with decentralized control (20 robots used)

```
====== PERFORMANCE REPORT ======
Mode              : DECENTRALIZED
Mission Start     : 0.0000 s
Mission Complete  : 45.9000 s
Mission Duration  : 45.9000 s
Collision Rate    : 1.0893 /s
Mean Form. Error  : 68.31 px
Path Efficiency   : 0.280
Avg Avoid. Time   : 0.14 s
Commands Issued   : 0 (decentralized ctrl)
================================
```

Table 1.3.2: Performance metrics for experiment with decentralized control (40 robots used)

```
====== PERFORMANCE REPORT ======
Mode              : DECENTRALIZED
Mission Start     : 0.0000 s
Mission Complete  : 47.5000 s
Mission Duration  : 47.5000 s
Collision Rate    : 2.8211 /s
Mean Form. Error  : 208.11 px
Path Efficiency   : 1.754
Avg Avoid. Time   : 0.98 s
Commands Issued   : 0 (decentralized ctrl)
================================
```

We calculate these performance metrics as follows:

Mission duration = Time leader reached target - Time experiment started

$$\textit{Mean Collision rate} = \frac{\Sigma\left(\frac{\text{Total collisions}}{\text{Simulation time}}\right)}{N}$$

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved

JOURNAL OF SCIENCE TECHNOLOGY AND EDUCATION 14(3), SEPTEMBER, 2026
E-ISSN: 3093-0898, PRINT ISSN: 2277-0011; Journal homepage: www.atbufstejoste.com



$$\text{Mean Formation error} = \frac{\sum(\text{Actual position} - \text{Desired position})}{N}$$

$$\text{Mean Path efficiency} = \frac{\sum(\frac{\text{Shortest path length}}{\text{Actual path length}})}{N}$$

$$\text{Mean Obstacle avoidance time} = \frac{\sum(\text{Time avoidance ends} - \text{Time avoidance starts})}{N}$$

Commands Issued = All communication commands exchanged between robots and central controller when control is centralized

Where N is the total number of robots defined in an experiment.

Table 1.4: Comparison of Performance Metrics for Centralized and Decentralized Controls

	Centralized				**Decentralized**			
Number of robots used	**7**	**10**	**20**	**40**	**7**	**10**	**20**	**40**
Mission duration (s)	985.7	951.3	1076.7	1029.75	52.95	44.8	45.9	47.5
Collision Rate / s	0.0619	0.114	0.0901	3.301	0.26	0.45	1.089	2.82
Mean Formation Error	74.01	79.45	69.40	156.95	25.14	32.24	68.31	208.11
Path Efficiency	0.058	0.077	0.040	0.045	0.236	0.287	0.280	1.754
Avg. Avoidance Time (s)	0.92	0.87	0.87	3.75	0.06	0.08	0.14	0.98
Commands Issued	779628938	911110311	1181746894	1867176738	0	0	0	0

CONCLUSION

This study demonstrates that while centralized control suits small robot formations, decentralized control offers superior scalability, fault tolerance, and communication efficiency. By separating global path planning from localized collision detection and avoidance, individual robots autonomously handle local disturbances while preserving mission objectives, allowing them to easily recover from collision detection and avoidance to continue in the formation.

Although evaluated solely through simulations in AnyLogic with certain motion and sensor assumptions, the framework successfully eliminates single points of failure and supports robust leader replacement. Future work will validate these findings on physical hardware, and benchmark the architecture against contemporary velocity-obstacle and safety-filter methods.

REFERENCES

Bohara, V., & Farzan, S. (2025). *Safety-Critical Formation Control of Non-Holonomic Multi-Robot Systems in Communication-Limited Environments*. https://doi.org/https://doi.org/10.48550/arXiv.2406.13707

Butler, B.A., Leung, C.H., & Pare, P.E. (2024). *Collaborative Safety-Critical Formation Control with Obstacle Avoidance*. https://www.researchgate.net/publication/384698652_Collaborative_Safety-Critical_Formation_Control_with_Obstacle_Avoidance/fulltext/6704a435b753fa724d649322/Collaborative-Safety-Critical-Formation-Control-with-Obstacle-Avoidance.pdf

Chang, X., Yang, Y., Zhang, Z., Jiao, J., Cheng, H., & Fu, W. (2025). Consensus-Based Formation Control for Heterogeneous Multi-Agent Systems in Complex Environments. *Drones*, *9*(3). https://doi.org/https://doi.org/10.3390/drones9030175

Dong, Z., Omidshafiei, S., & Everett, M. (2024). *Collision Avoidance Verification of Multiagent Systems with Learned Policies*. *8*, 652–657.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



https://doi.org/https://doi.org/10.1109/LCSYS.2024.3400190

Estevez, J., Jose Manuel Lopez-Guede, Javier Del Valle-Echavarri, Caballero-Martin, D., & Graña, M. (2025). Enhanced dynamic obstacle avoidance for linked multi-agent systems by an extended hybrid reciprocal velocity obstacle model. *Integrated Computer-Aided Engineering*, *33*(1). https://doi.org/https://doi.org/10.1177/10692509251368675

Fan, Y., Li, X., Liu, X., Cheng, S., & Wang, X. (2024). Multi-robot consensus formation based on virtual spring obstacle avoidance. *Mechanical Sciences*, *15*(1), 195–207. https://doi.org/https://doi.org/10.5194/ms-15-195-2024

Hellander, A., & Axehill, D. (2026). *Multi-Agent Motion Planning for Simultaneous Arrival using Time-Reversed Search and Distributed Optimal Control*. https://doi.org/https://doi.org/10.48550/arXiv.2605.02019

Jamshidpey, A., Wahby, M., M. K. Heinrich, & Allwright, M. (2024). *Centralization vs. Decentralization in multi-robot coverage: Ground robots under UAV supervision*. https://doi.org/https://doi.org/10.48550/arXiv.2408.06553

Lei, J., & Pei, Y. (2025). Recent Advances in Multi-Agent Reinforcement Learning for Intelligent Automation and Control of Water Environment Systems. *Machines*. https://doi.org/https://doi.org/10.3390/machines13060503

Liu, B., Guo, X., & Dong, C. (2025). Research on the Characteristics and Causes of Collision Accidents Between Merchant Ships and Fishing Vessels in China's Coastal Waters. *Applied Sciences*. https://doi.org/https://doi.org/10.3390/app16010268

Liu, G., Wen, N., Long, F., & Zhang, R. (2023). A Formation Control and Obstacle Avoidance Method for Multiple Unmanned Surface Vehicles. *Journal of Marine Science and Engineering*, *11*(12). https://doi.org/https://doi.org/10.3390/jmse11122346

Martinez-Baselga, D., Lanaspa, D., Riazuelo, L., & Montano, L. (2026). Adaptive Optimal Collision Avoidance of Dynamic Agents for Differential-Drive Robots. *Robotics*, *15*(4). https://doi.org/https://doi.org/10.3390/robotics15040072

Medhi, J., Liu, R., Wang, Q., & Chen, X. (2023). Robust Multiagent Reinforcement Learning for UAV Systems: Countering Byzantine Attacks. *Information*. https://doi.org/https://doi.org/10.3390/info14110623

Pham, T.V., & Nguyen, Q.T.T. (2023). *Adaptive formation control of nonlinear multi-agent systems with dynamic event-triggered communication*. https://doi.org/https://doi.org/10.1016/j.sysconle.2023.105652

Roncero, A. S., Muchacho, R. I. C., & Ögren, P. (2024). Multi-Agent Obstacle Avoidance using Velocity Obstacles and Control Barrier Functions. https://doi.org/10.48550/arxiv.2409.10117

Shi, L., Mazumdar, E., Chi, Y., & Wierman, A. (2024). Sample-Efficient Robust Multi-Agent Reinforcement Learning in the Face of Environmental Uncertainty. *Proceedings of the 41 St International Conference on Machine Learning, Vienna, Austria*, 44909–44959.

Solano-Castellanos, J.A., Peter, A.F., & Annaswamy, A. (2025). *Safe and Stable Formation Control with Autonomous Multi-Agents Using Adaptive Control*. *59*(30), 90–95. https://doi.org/https://doi.org/10.1016/j.ifacol.2025.12.218

Vielmetti, T.M., Devansh R. Agrawal, & Panagou, D. (2026). *R3R: Decentralized Multi-Agent Collision Avoidance with Infinite-Horizon Safety*. https://arxiv.org/pdf/2510.06436

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved



Vrba, P., Mařík, V., Přeučil, L., Kulich, M., & Šišlák, D. (2007). *Collision Avoidance Algorithms: Multi-agent Approach*. 348–360.

Wang, S., Wang, H., & Li, Q. (2025). Distributed observer-based formation control and obstacle avoidance of nonholonomic mobile robots with input saturation. *Knowledge Based Systems*, *319*. https://doi.org/https://doi.org/10.1016/j.knosys.2025.113542

Wang, Y., & Zhao, Y. (2024). Multiple Ships Cooperative Navigation and Collision Avoidance using Multi-agent Reinforcement Learning with Communication. *Ocean Engineering*, *320*. https://doi.org/https://doi.org/10.1016/j.oceaneng.2024.120244

Xie, Y., Yu, C., Zang, H., Gao, F., Tang, W., Huang, J., Chen, J., Xu, B., Wu, Y., & Wang, Y. (2024). Multi-UAV Behavior-based Formation with Static and Dynamic Obstacles Avoidance via Reinforcement Learning. https://doi.org/10.48550/arXiv.2410.18495

Xiong, Z., Liu, Z., & Luo, Y. (2023). Collision and obstacle avoidance strategy for multi-agent systems with velocity dynamic programming. *Measurement and Control*, *56*(1–2), 257–268. https://doi.org/https://doi.org/10.1177/00202940221122195

Xue, S., Guo, Z., Tan, J., Qu, K., Cao, H., & Chen, B. (2026). Fixed-time formation control for multiagent systems with velocity-based collision avoidance. *ISA Transactions - The Journal of Automation*, *169*, 26–38. https://doi.org/https://doi.org/10.1016/j.isatra.2025.12.029.

Corresponding author: Kenneth Kayock Attans
Department of Informatics, Faculty of Computing, Kaduna State University, Nigeria.
© 2026. Faculty of Technology Education. ATBU Bauchi. All rights reserved