Collaborative Transport Using Warehouse Robots with Model Predictive Control

Azamat Abdikarimov, Mitchell Hornak, Nathan Lau

Abstract

This paper presents a comparison of two Model Predictive Control (MPC) strategies for multi-robot formation control along predefined warehouse paths: a single-leader approach and a dual-leader approach. A team of four autonomous differential-drive robots is tasked with maintaining a square formation while tracking a global centerline trajectory. In the single-leader strategy, one robot follows the path directly while the others follow their own offset path but modify controls to maintain relative positions to the leader; in the dual-leader strategy, both front robots independently track offset paths while maintaining formation to each other and the rear robots track the same offset paths while only maintaining formation to their respective leader. Each robot solves its own linearized MPC problem at every timestep, optimizing for path tracking, formation maintenance, and control smoothness. Testing was performed in simulation on three different trajectories to closely simulate actual maneuvers that would be performed in a warehouse: drive straight, avoid small obstacles, and make a 90 degree turn. Simulation results show that the dual-leader approach provides superior formation stability and tracking performance in the first two scenarios, while the single-leader approach performs better during turns. Based on these findings, we propose that a hybrid strategy combining both methods, using dual-leader control for straighter regions and single-leader control for turns, would achieve the best overall formation performance. However, implementation of dynamic hybrid switching is left for future work.

I. Introduction

The demand for greater efficiency in the processing and storage industries has accelerated the development of robotic automation. Large corporations increasingly rely on fleets of autonomous mobile robots to transport inventory within their warehouses, gaining significant logistical advantages. Companies such as Amazon have acquired specialized robotics firms, including Kiva Systems and Cloostermans, to build extensive in-house robotic operations under the Amazon Robotics brand [1]. However, despite rapid advances at the top end of the market, access to automation remains highly uneven across the industry. Smaller storage facilities and independent warehouses often struggle to adopt robotic systems due to high upfront investment costs and the difficulty of reconfiguring limited floor space for traditional automation architectures. This growing technological gap risks creating monopolistic market conditions, restricting innovation and widening the efficiency divide between large and small enterprises. To address these challenges, more flexible and scalable robotic solutions are needed. One promising approach involves the use of smaller, lower-cost robots operating collaboratively in a swarm to transport larger or heavier inventory items. By distributing load-sharing across multiple robots, such systems offer better adaptability to constrained environments and reduce the cost barriers associated with traditional large-scale automation. Advancing coordinated multi-robot control strategies is therefore a critical step toward democratizing access to warehouse automation and ensuring that efficiency gains are broadly shared across the storage and logistics industries.

II. Background

In warehouse environments, smaller collaborative robots offer distinct advantages over larger industrial robots. Their compact size and lightweight design make them ideal for facilities with limited space, as they can operate without the need for protective fencing or extensive safety barriers. This not only conserves valuable floor space but also allows for more flexible workspace configurations. Their versatility, adaptability, and low relative cost make them well-suited for completing small tasks alone while also being able to work together to collaboratively transport larger packages [2]. Overall, the integration of smaller collaborative robots in warehouse settings leads to increased efficiency and, most of all, flexibility.

Formation control has become an increasingly important and widely used technique in multi-robot systems, enabling coordinated behaviors such as collaborative transport, surveillance, and area coverage [3]. Several classical approaches to formation control have been developed, each offering a distinct strategy for managing group behavior. In the leader-follower approach, one robot is designated as the leader and follows a predefined trajectory, while the remaining robots maintain fixed relative positions with respect to the leader [4]. Behavior-based formation control relies on local rules and reactive behaviors, where each robot adjusts its motion based on nearby teammates and environmental cues [4]. In the virtual structure approach, the entire formation is treated as a single rigid body, with each robot occupying a fixed position within the virtual frame [5].

While these approaches define how the formation is organized, an effective control method is required to ensure that robots can track their positions accurately and maintain formation in dynamic environments under real-world constraints. MPC has emerged as a promising method to address this challenge by predicting future states and solving an optimization problem that explicitly respects system dynamics and physical constraints [6]. MPC can be applied within any of these classical formation frameworks, optimizing control inputs not only to follow the global trajectory but also to maintain the formation structure. Among these strategies, the leader-follower approach has gained particular attention within MPC-based systems due to its simplicity, scalability, and reduced communication requirements, making it especially practical for multi-robot coordination tasks [7].

III. Problem Description

In this work, we address the problem of coordinating a team of autonomous differential-drive robots to travel in formation along a predefined warehouse path with a speed profile with the intent of the robots cooperatively transporting packages resting on top of them. Each robot is modeled as a discrete-time unicycle system with linear acceleration and angular velocity control inputs where their behavior is governed by a decentralized MPC framework, each robot solving for it's own path tracking and formation handling. The objective is to maintain a geometric formation while tracking along the given path, despite variations in curvature and speed along the way.

The formation consists of four identical robots of arbitrary size arranged in a square 0.5 meters on a side, with the robot centers of mass starting at each corner of the formation. Two different coordination strategies are considered to maintain the 0.5 meters of separation:

- **Single Leader Strategy:** One robot acts as the sole leader following their path unimpeded, while all other robots follow their given path but adjust speed and trajectory relative to the leader to maintain formation.
- **Dual Leader Strategy:** Both the left and right front robots serve as leaders of their respective side of the square, following an offset path from the centerline while also adjusting speed and trajectory to keep in formation with each other. In this strategy, the rear robots are following the same paths but adjust their controls based only on their respective side leader to keep formation.

Proper tracking of the given path is defined by the robots keeping the centerline path in the formation center of mass (see Figure 1), so the left and right robots are assigned an individual offset path derived from the global centerline to account for their desired position within the formation. The reference paths for the robots are generated by shifting the centerline laterally according to the formation. Additionally, the speed profile along the path is adjusted based on local curvature to ensure feasible turning behavior and limit acceleration demands. At every time step, each robot solves a local MPC problem over a finite prediction horizon. The cost function aims to minimize deviation from the desired state (position, orientation, and speed errors), deviation from formation, control effort and control input smoothness by modifying the input linear acceleration and angular velocity.

Simulation scenarios are conducted in Python using CVXPY-based MPC solvers. The robots are tested on three different trajectories that closely emulate maneuvers that would be performed in a typical warehouse: drive straight, avoid small obstacles, and make a 90 degree turn. These paths introduce natural challenges in maintaining formation fidelity, particularly when different robots experience different curvature-induced speed demands. The goal of the study is to evaluate the effectiveness of each coordination strategy in achieving accurate path tracking while maintaining a robust formation. Performance is assessed quantitatively through metrics such as velocity profiles, tracking error, formation error, time to goal, and qualitatively through visualization of robot trajectories and formation evolution over time.

IV. Approach

In this project, we compared the two MPC strategies laid out in Section III to maintain a four-robot square-shaped formation while tracking a preplanned path in a warehouse environment. Each robot has a decentralized MPC controller to predict and optimize its trajectory over a finite horizon, minimizing deviations from a reference trajectory while satisfying dynamic constraints and formation constraints. The global simulation procedure is laid out in Algorithm 1.

A. Robot Kinematic Model

Each robot is modeled as a discrete-time unicycle system (differential-drive robot) with four state variables and two control inputs. The state vector for each robot is defined as:

$$\mathbf{x} = \begin{bmatrix} x & y & \theta & v \end{bmatrix}^\top \tag{1}$$

where x and y denote the global position coordinates, θ is the robot's heading angle (yaw) measured from the positive x-axis, and v is the linear velocity.

The control input vector is given by:

$$\mathbf{u} = \begin{bmatrix} a & \omega \end{bmatrix}^\top \tag{2}$$

where a is the linear acceleration and ω is the angular velocity (yaw rate).

The continuous-time kinematic equations governing the motion of each robot are:

$$\begin{bmatrix} \dot{x} & \dot{y} & \dot{\theta} & \dot{v} \end{bmatrix}^\top = \begin{bmatrix} v\cos(\theta) & v\sin(\theta) & \omega & a \end{bmatrix}^\top \tag{3}$$

Algorithm 1 Formation Control Simulation Procedure

Input: Global centerline path $\{(x, y, yaw, \kappa)\}$, formation offsets, MPC parameters
Initialize: Robot states positioned according to formation offsets
Compute: Speed profile based on path curvature
Set simulation time $t \leftarrow 0$
while $t \leq$ maximum allowed simulation time **do**
    for each robot i in the formation **do**
        Generate offset point for robot i based on formation offset
        Calculate reference trajectory x_{ref} from nearest path point
        for each linearization iteration (up to max allowed) **do**
            Predict future motion using current control inputs
            Linearize robot dynamics around predicted trajectory
            Solve MPC optimization to optimize acceleration and angular velocity
        end for
        Apply first control input to update robot state
        Store predicted future trajectory for use by follower MPC
    end for
    Increment simulation time $t \leftarrow t + \Delta t$
    if all robots have reached the goal **then**
        break
    end if
end while

For implementation within the MPC framework, these equations are discretized using forward Euler integration with a fixed sampling interval Δt. The discrete-time state update equations become:

$$x_{k+1} = x_k + v_k \cos(\theta_k)\Delta t \tag{4}$$
$$y_{k+1} = y_k + v_k \sin(\theta_k)\Delta t \tag{5}$$
$$\theta_{k+1} = \theta_k + \omega_k \Delta t \tag{6}$$
$$v_{k+1} = v_k + a_k \Delta t \tag{7}$$

The kinematic model captures the non-holonomic constraint inherent to differential-drive robots, which prevents instantaneous lateral movement. It provides a balance between simplicity and accuracy that is suitable for trajectory planning and formation maintenance tasks at moderate speeds. This model is linearized at each time step around the current predicted trajectory to generate local linear approximations, which are then used within the MPC optimization problem. The linearization accounts for variations in heading and velocity, enabling the robots to robustly handle curved paths and formation adjustments in real-time.

B. MPC Formulation with Formation Constraints

At each control cycle, each robot solves a finite-horizon optimization problem to compute its control inputs. The problem is formulated as follows over a prediction horizon T:

$$\min_{\{\mathbf{u}_k\}_{k=0}^{T-1}} \quad J = \sum_{k=0}^{T-1} \left(\|\mathbf{x}_k - \mathbf{x}_k^{\text{ref}}\|_Q^2 + \|\mathbf{u}_k\|_R^2 + \|\mathbf{u}_k - \mathbf{u}_{k-1}\|_{R_d}^2 + \text{FormationCost}_k \right) + \|\mathbf{x}_T - \mathbf{x}_T^{\text{ref}}\|_{Q_f}^2, \tag{8}$$

subject to:

$$\mathbf{x}_{k+1} = A_k \mathbf{x}_k + B_k \mathbf{u}_k + \mathbf{C}_k, \tag{9}$$
$$v_{\text{min}} \leq v_k \leq v_{\text{max}}, \tag{10}$$
$$|a_k| \leq a_{\text{max}}, \tag{11}$$
$$|\omega_k| \leq \omega_{\text{max}}, \tag{12}$$
$$|\omega_{k+1} - \omega_k| \leq \Delta\omega_{\text{max}}. \tag{13}$$

Here:

- $\mathbf{x}_k = [x_k, y_k, \theta_k, v_k]^T$ is the state vector at timestep k.

- $\mathbf{u}_k = [a_k, \omega_k]^T$ is the control input vector at timestep k, consisting of linear acceleration a_k and angular velocity ω_k.
- $\mathbf{x}_k^{\text{ref}}$ is the desired reference state, derived from the speed profile and centerline path with lateral offsets applied for formation maintenance.
- Q, R, and R_d are positive semi-definite weight matrices that penalize state error, control effort, and control input rate-of-change, respectively.
- Q_f is the terminal cost weight matrix applied at the final prediction step.
- FormationCost_k encodes additional penalties at timestep k that enforce formation maintenance relative to leader robots.

The constraints ensure the robot's velocity v_k remains between minimum and maximum bounds, the acceleration a_k and angular velocity ω_k stay within prescribed limits, and the angular velocity does not change too abruptly between successive timesteps.

The matrices A_k, B_k, and vector $\mathbf{C}_k$ represent the linearized discrete-time kinematic model of the robot around the predicted trajectory, updated at each timestep. Specifically, the model is linearized using the current predicted heading θ_k and velocity v_k:

$$A_k = \begin{bmatrix} 1 & 0 & -v_k \sin(\theta_k)\Delta t & \cos(\theta_k)\Delta t \\ 0 & 1 & v_k \cos(\theta_k)\Delta t & \sin(\theta_k)\Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{14}$$

$$B_k = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & \Delta t \\ \Delta t & 0 \end{bmatrix}, \quad \mathbf{C}_k = \mathbf{0}. \tag{15}$$

Formation Cost Term:

Depending on the formation structure, FormationCost_k is defined differently:

- **Single Leader Model:** The front-left robot (Robot 0) follows the global centerline directly without accommodating for the position of its fellow robots, while the follower robots (Robots 1-3) minimize deviations from desired relative positions to Robot 0. For each follower robot i, at each timestep:

$$\text{FormationCost}_k = w_x(x_i - x_0 - \Delta x_{i0})^2 + w_y(y_i - y_0 - \Delta y_{i0})^2 + w_\theta(\theta_i - \theta_0)^2, \tag{16}$$

  where $(\Delta x_{i0}, \Delta y_{i0})$ are the desired relative offsets in the global frame.
- **Dual Leader Model:** The front robots (Robot 0 and Robot 1) independently track left and right offset paths relative to the global centerline. Rear robots (Robot 2 and Robot 3) maintain relative distances to their respective front-side leaders. For each rear robot i (where $i \in \{2, 3\}$), the formation maintenance cost at timestep k is:

$$\text{FormationCost}_k = w_x(x_i - x_{\text{leader}(i)} - \Delta x_i)^2 + w_y(y_i - y_{\text{leader}(i)} - \Delta y_i)^2, \tag{17}$$

  where $\text{leader}(i)$ denotes the corresponding front leader robot (Robot 0 for Robot 2, Robot 1 for Robot 3). Additionally, a symmetric constraint is imposed between the two front robots themselves (Robot 0 and Robot 1) to maintain a consistent lateral separation. A cost term penalizes deviations from the desired lateral offset between their predicted positions, ensuring that the front formation remains cohesive during curved maneuvers:

$$\text{FormationCost}_k = w_y(y_0 - y_1 - \Delta y_{01})^2, \tag{18}$$

  where Δy_{01} is the desired lateral separation between the front-left and front-right robots in the global frame.
  where $\text{leader}(i)$ denotes the corresponding leader for robot i.

The complete optimization problem is formulated as a convex Quadratic Program (QP) and solved using the `CVXPY` modeling language with the `Clarabel` solver backend. The full list of parameters used to get results can be found in Table I.

C. Iterative Linearization and Warm-Starting

Due to the robot kinematics being nonlinear in heading angle, a simple one-shot linearization would not be sufficiently accurate for long prediction horizons. Therefore, an iterative linear MPC approach is employed. At each control cycle, the robot's future trajectory is first predicted based on previous control inputs. The robot's dynamics are then re-linearized around this predicted motion, and the MPC problem is solved again to update the control sequence. This process is repeated for a fixed number of iterations or until convergence. Warm-starting the control inputs with previous solutions is also employed to significantly accelerate convergence and improve numerical stability within the solver.

TABLE I: MPC Parameter Summary for Single Leader and Dual Leader

Parameter	Single Leader	Dual Leader
State Weight Matrix Q	$\text{diag}(500, 500, 20, 1)$	$\text{diag}(1000, 1000, 20, 1)$
Input Weight Matrix R	$\text{diag}(0.1, 0.1)$	$\text{diag}(0.1, 0.1)$
Input Rate Weight Matrix R_d	$\text{diag}(0.5, 1.0)$	$\text{diag}(0.5, 1.0)$
Terminal Weight Matrix Q_f	Same as Q	Same as Q
Prediction Horizon T	5 steps	5 steps
Sampling Time Δt	0.05 s	0.05 s
Max Velocity $v_{\max}$	5.0 m/s	5.0 m/s
Max Acceleration $a_{\max}$	5.0 m/s^2	5.0 m/s^2
Max Angular Velocity $\omega_{\max}$	120°/s (≈ 2.09 rad/s)	120°/s (≈ 2.09 rad/s)
Max Angular Rate Change $\Delta\omega_{\max}$	5.0 rad/s^2	5.0 rad/s^2
Formation Cost Weight (w_x)	1000.0	8000.0
Formation Cost Weight (w_y)	3000.0	3000.0
Yaw Alignment Weight (w_θ)	500.0	1000.0
Extra Lateral Error Penalty	300.0	300.0
Velocity Tracking Weight	5.0	5.0

D. Speed Profile Adjustment

Before control begins, a nominal global speed profile $v_{\text{global}}(s)$ is generated along the entire reference path based on local path curvature. The global speed profile $v_{\text{global}}(s)$ is embedded directly into the reference trajectory $\mathbf{x}_k^{\text{ref}}$ as the target velocity component. Each robot tracks both its spatial position and the corresponding reference speed simultaneously during MPC optimization. The target speed at each path location s is computed according to:

$$v_{\text{global}}(s) = \min\left(v_{\max}, \frac{k_{\text{curv}}}{|\kappa(s)| + \epsilon}\right), \tag{19}$$

where $v_{\max}$ is the maximum allowable speed, k_{curv} is a curvature tuning constant, $\kappa(s)$ is the path curvature at distance s, and ϵ is a small regularization term to avoid division by zero. This global speed profile ensures that robots automatically slow down in sharp corners, helping maintain stability and improving tracking performance during tight maneuvers. During operation, each robot attempts to track the global target speed at its assigned reference position along the path. Minor adjustments to the nominal speed may occur dynamically in the MPC optimization, especially for formation maintenance, but the overall structure is dictated by the precomputed global speed profile.

E. Curvature-Aware Constraint Relaxation and Speed Boosting

To maintain a tight formation and improve responsiveness during curved segments of the path, two curvature-aware adjustment mechanisms were implemented: constraint relaxation and speed boosting. First, a curvature-aware constraint relaxation is applied to the formation maintenance objectives. In regions of higher curvature, the lateral separation between formation members naturally becomes distorted due to robots on the outside of the turn needing to take longer, wider turns, while the robots on the inside require a sharp, short turn. To compensate for this effect, the desired relative offsets between robots are dynamically adjusted based on local path curvature. For each robot, the curvature κ at its reference index is estimated from the spline path, and the nominal formation offsets $(\Delta x, \Delta y)$ are modified according to:

$$\Delta x' = \Delta x(1 + \kappa \cdot s_x), \tag{20}$$

$$\Delta y' = \Delta y(1 + \kappa \cdot s_y), \tag{21}$$

where s_x and s_y are tunable curvature sensitivity scaling coefficients. This adjustment allows the formation to "bend" naturally along curves rather than rigidly enforcing a flat rectangular structure, reducing lateral drift during sharp turns and preventing gridlock. Second, a curvature-aware speed boosting mechanism is introduced to help robots maintain longitudinal cohesion. In particular, a soft correction is applied to a robot's target speed when it falls behind its leader. A small gain proportional to the longitudinal distance error is used to slightly increase the robot's commanded speed, encouraging it to close gaps gently without introducing abrupt acceleration changes. Together, these two mechanisms ensure that robots remain closely aligned in both lateral and longitudinal directions, preserving formation integrity even during challenging curved maneuvers.

F. Path Unlocking Algorithm for Rear Robots

In decentralized MPC formation control, it is common for the optimization to find undesirable and unexpected local minima [8]. In the case of high curvature paths, because the front robots would slow down first, the rear robots often found it more optimal to overtake the front robots due to the low cost of the maneuver; this would result in a break of formation and lead to the package being dropped. To maintain proper formation geometry and prevent rear robots from overtaking their respective leaders, a path unlocking mechanism was incorporated into the reference trajectory generation process. The path unlocking method constrains the reference trajectories available to each rear robot based on the current progress of its designated leader robot. Specifically, a rear robot's reference trajectory is generated such that it cannot ”see” beyond a certain proximity relative to the leader's progress along the path. This is achieved by imposing an *unlocking index limit* during reference trajectory calculation.

Additionally, a curvature-based correction factor is applied to the unlocking limit. In high-curvature sections, rear robots are allowed to unlock slightly more forward to account for natural path distortion in turns, while still maintaining their intended following distance. This curvature adjustment helps prevent stalling or excessive delay of rear robots during sharp turns. Overall, the path unlocking strategy plays a critical role in ensuring that the formation remains orderly, with rear robots smoothly following the front robots without overtaking or violating the designed formation structure.

V. Experimental Setup

The multi-robot system consists of four mobile robots operating in a two-dimensional environment. Each robot is modeled using a kinematic unicycle model and is capable of controlling its linear velocity and yaw rate independently. The robots are tasked with following a given reference trajectory and speed profile while maintaining a square formation pattern seen in Figure 1.

Two leadership configurations are considered:

- **Single Leader**: All robots follow their respective offset path but Robot 0 acts as the sole formation leader; Robots 1–3 adjust controls to maintain formation relative to Robot 0.
- **Dual Leader**: All robots follow their respective offset path but Robots 0 and 1 jointly lead the formation, adjusting controls to keep in line with each other but not the rear robots; Robots 2 and 3 adjust controls to maintain formation relative to the two leaders.

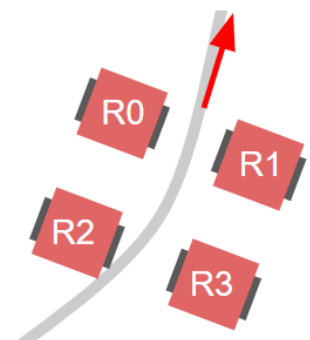

Fig. 1: Visualization of Square Robot Formation Layout with Robots Labeled.

Three distinct path types are used for evaluation:

- **Turn Path**: Consists of a sharp 90 degree turning maneuver.
- **Straight Path**: A simple straight-line trajectory.
- **Curved Path**: Alternating small left and right turns to simulate small obstacle avoidance.

Each combination of leadership configuration and path type is tested independently. For each run, robot states consisting of position, orientation, linear velocity, acceleration, and yaw rate are recorded. Three performance metrics are analyzed: trajectory tracking, formation maintenance (inter-robot distances), and individual robot speed profiles. Data collected during each run is post-processed to generate plots for qualitative and quantitative performance evaluation.

VI. Results and Discussion

This section presents and analyzes the performance of the multi-robot system under the single leader and dual leader configurations across different path types (straight, turn, and curved). Metrics such as trajectory tracking, formation maintenance, and speed profiles are evaluated.

A. Single Leader Formation

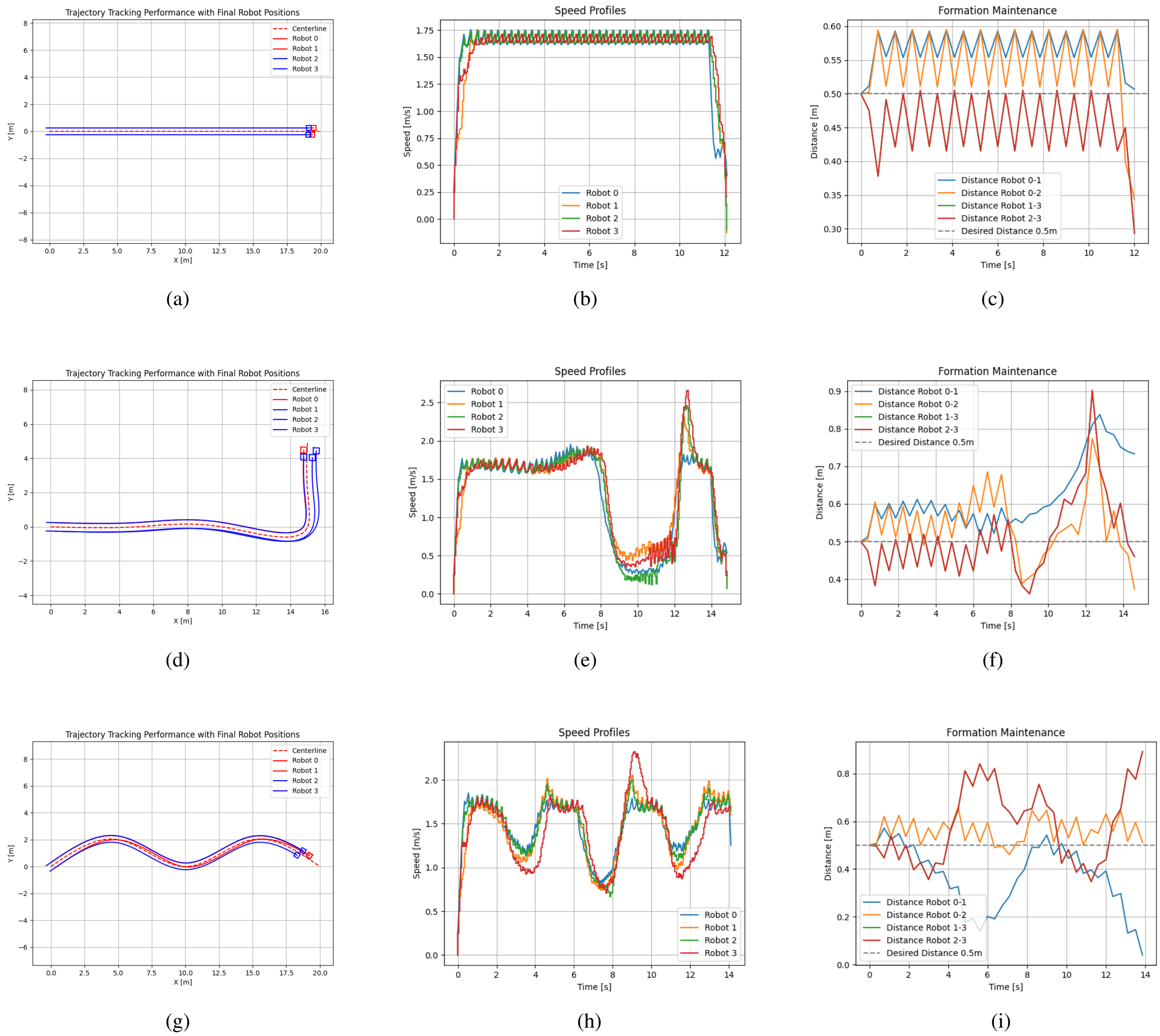

Fig. 2: Plots of Trajectory Tracking, Speed Profiles, and Formation Maintenance for the Single Leader Method. (a)-(c) - Straight Path; (d)-(f) - Turn Path; (g)-(i) - Curved Path; (a),(d),(g) - Trajectory Tracking; (b),(e),(h) - Speed Profiles; (c),(f),(i) - Formation Maintenance

1) Trajectory Tracking Performance: The single leader configuration exhibited excellent trajectory tracking on the straight path, with robots maintaining close alignment to the centerline and minimal lateral deviation. During the turn path, slight deviations appeared as the robots negotiated the turn, leading to spacing increases between the leader and followers. In the curved path, trajectory tracking performance further degraded due to frequent heading changes, causing noticeable lateral drift, which resulted in Robot 1 to be positioned beneath Robot 0.

2) Speed Profiles: In the straight path, the robots achieved a rapid and smooth convergence to the desired cruising speed, maintaining nearly constant velocities throughout the motion with minimal oscillations. During the turn path, speed profiles became more variable, with noticeable decelerations and accelerations as the robots negotiated curves, reflecting dynamic adjustments to maintain formation and trajectory tracking. In the curved path, the speed profiles exhibited frequent fluctuations, including sharper deceleration and acceleration events at each directional change, indicating greater control effort required to handle the continuous path curvature while maintaining formation coherence.

3) Formation Maintenance: During the straight path, the robots maintained relatively stable inter-robot distances around the desired 0.5 meters, with small oscillations observed across all robot pairs. The distances fluctuated slightly but remained bounded, indicating robust formation preservation in straight-line motion. In the turn path, larger deviations were observed, particularly near the sharp turning section, where distance errors temporarily increased, especially between Robot 0 and Robot 1. This behavior is expected and allowed though, due to the curvature-aware constraint relaxation and speed boosting, and it is promising to note the formation distances begin to converge back to their desired distance after the turn, indicated robust formation maintenance. In the curved path, formation ended up with more significant fluctuations, most likely due to the fact that heading is constantly changing, so the controller is finding it more optimal to trade formation maintenance for reference path following. From the turn trajectory we observed that returning to formation after a turn does take some extra time so in a state of constant turning there is expected to be some drift between the robots. Moreover, it can be seen that the distance between Robots 0 and 1 almost reached 0 meters at the end of the path, reflecting the increased difficulty in maintaining formation during frequent directional changes.

B. Dual Leader Formation

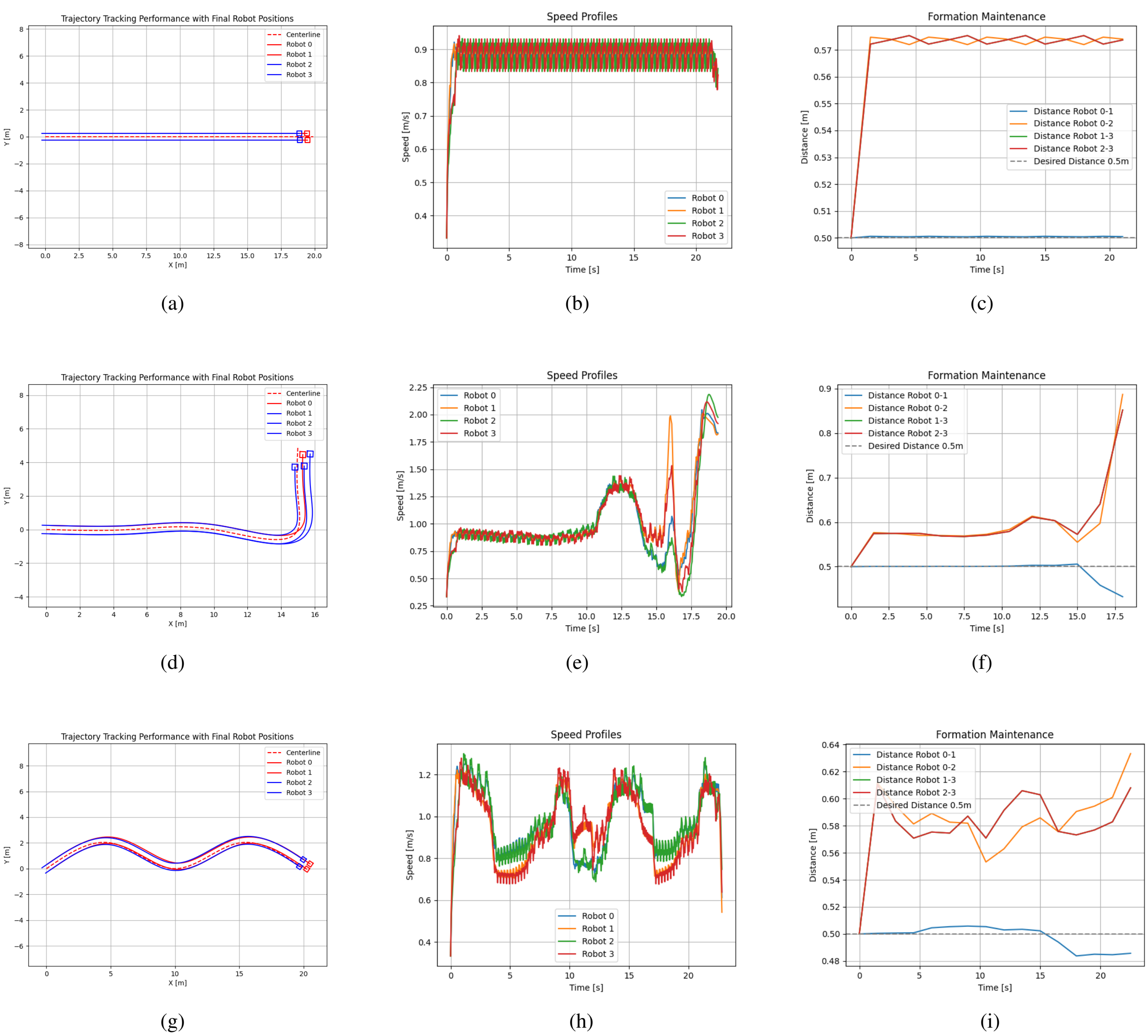

Fig. 3: Plots of Trajectory Tracking, Speed Profiles, and Formation Maintenance for the Dual Leader Method. (a)-(c) - Straight Path; (d)-(f) - Turn Path; (g)-(i) - Curved Path; (a),(d),(g) - Trajectory Tracking; (b),(e),(h) - Speed Profiles; (c),(f),(i) - Formation Maintenance

1) Trajectory Tracking Performance: In the straight path, the dual leader configuration achieved highly accurate trajectory tracking, with all robots closely aligned to their offset path and maintaining minimal deviation throughout the motion. During the turn path, the robots followed the curved segments effectively, although there were lateral drifts observed, resulting in a noticeably worse adherence to the reference compared to the single leader configuration. In the curved path, the robots were able to adhere to the reference trajectory well despite the frequent directional changes, exhibiting reduced lateral offsets and better alignment along the path when compared to the single leader case.

2) Speed Profiles: In the straight path, the robots rapidly reached the target cruising speed and maintained it steadily with minimal oscillations, reflecting effective coordination between the two leaders and their followers. During the turn path, the robots exhibited expected speed variations when navigating curves, with moderate decelerations for the interior robots and accelerations for the exterior robots, thus enabling smooth path tracking without significant disruptions. In the curved path, more pronounced speed fluctuations occurred due to the constant change in heading, but the robots managed to maintain coordinated speed adjustments and recover consistently after each transition, demonstrating robustness of the MPC.

3) Formation Maintenance: Throughout the straight path, the robots maintained very stable inter-robot distances, with deviations from the desired 0.5 meters remaining minimal and consistent. During the turn path, although the relative configuration of the formation remained consistent, the formation degraded through the sharp turn and only continued to stray from the desired formation after the turn, in contrast to the single leader, which demonstrated formation recovery after a sharp turn. In the curved path, formation maintenance became more challenging, with larger distance variations appearing, particularly among the rear robots; however, the overall structure of the formation was preserved across the course with mean and maximum deviation from formation being better than that observed in the same path for the single leader method.

C. Comparison Summary

The times for robots to reach their goal and formation maintenance deviations for the Single Leader and Dual Leader configurations are summarized in Tables II and III, respectively. Table II shows that the Single Leader configuration consistently achieved faster times to complete its path across all path types, with durations between 12 and 15 seconds. In comparison, the Dual Leader took longer to reach it's goal, ranging from 19 to 23 seconds. This behavior is due to the dual leaders influencing each others speeds and compromising control for staying in line with each other, which in turn holds back the entire formation. In the case of the single leader, the leader traverses the path uninhibited by the state of the other robots, so the entire formation is working to keep up which increases average speeds though usually sacrifices formation maintenance. Table III presents the mean and maximum deviations from the desired 0.5 meter inter-robot distance. Across all paths, the Dual Leader configuration achieved lower mean and maximum deviations than the Single Leader setup, except for the turn path, where maximum deviation for the dual leader was marginally higher. This data, along with the path tracking data, indicates the dual leader is better for formation stability during straight paths while the single leader excels at the sharp turn path.

TABLE II: Comparison of Time to Reach Goal Between Configurations

Configuration	Straight Path	Turn Path	Curved Path
Single Leader	12.1 s	14.8 s	14.1 s
Dual Leader	21.7 s	19.4 s	22.8 s

TABLE III: Formation Maintenance Deviation Results

Configuration	Path Type	Mean Deviation (m)	Max Deviation (m)
Leader-Follower	Straight	0.0469	0.2150
Leader-Follower	Turn	0.0706	0.4019
Leader-Follower	Curved	0.1007	0.4794
Dual-Leader	Straight	0.0362	0.0768
Dual-Leader	Turn	0.0621	0.4674
Dual-Leader	Curved	0.0590	0.1372

D. Discussion

The results reveal that the Single Leader configuration is more time-efficient and excels at maintaining robust formation and path following in a single sharp turn; however, it is less effective at maintaining the desired tight formations in straight and curved scenarios. In contrast, the Dual Leader configuration, while slower and more computationally intensive, consistently demonstrated superior formation maintenance in straight and curved sections, while demonstrating poor formation maintenance through sharp turns.

The strengths of the Dual Leader approach are most evident during straight and curved paths, where coordination between the two leaders helps maintain better alignment and minimizes deformation of the formation at slew velocity. This configuration reduces the risk of followers lagging behind or drifting off the reference path. However, it is important to note that even when formation deviations were low, the entire formation could drift away from the centerline trajectory. Tight internal spacing does not necessarily imply that the group as a whole was accurately following the reference path, especially in dynamically changing sections.

Unexpected behaviors included occasional fluctuations in inter-robot distances during straight segments in the Dual Leader configuration, likely caused by slight synchronization delays between leaders and the controller constantly trying to perfect the formation. In the future, an important investigation would be to incorporate small windows of allowed formation deviation so the controller does not continuously oscillate trying to achieve perfection. Additionally, during curved sections, minor instability was observed where one of the follower robots temporarily overshot its intended position, though the formation recovered quickly, demonstrating robustness of the MPC. Overall, although the Dual Leader configuration imposes a higher coordination and computational burden, it provides substantial advantages in maintaining formation integrity during straighter path scenarios, making it a preferable choice when trajectory tracking and formation maintenance on straightaways are prioritized over execution speed. On the other hand, the Single Leader configuration excelled at completing its path in minimal time and formation maintenance on sharp turns, making it the preferable choice when trajectory tracking and formation maintenance are prioritized on turns or when speed is necessary.

VII. Conclusion

This paper presented an MPC framework for formation control of four autonomous warehouse robots traveling along predefined paths. Two coordination strategies were investigated: a single leader approach, where one robot led the formation, and a dual leader approach, where both front robots acted as leaders of their respective side of the formation. Each robot solved its own linearized MPC problem at each timestep, balancing objectives of path tracking, formation maintenance, velocity regulation, and control smoothness.

Simulation results demonstrated that the dual-leader strategy outperformed the single-leader approach on straight and moderately curved segments, maintaining tighter formation geometry and reducing lateral errors. Conversely, the single-leader approach proved more effective in coordinating sharp turns, ensuring cohesive rotation of the formation with fewer distortions. These findings suggest that neither strategy alone is suitable across all path types.

Based on the observed strengths and weaknesses, we propose that a hybrid coordination strategy, dynamically switching between dual leader and single leader control based on local path curvature, would achieve the best overall performance. Although hybrid switching was not implemented in the current study, it represents a promising direction for future work to enhance the robustness and flexibility of multi-robot formation control in structured environments.

A. Future Work

Based on the results observed in this study, an important direction for future work is the development and implementation of the hybrid control strategy. A controller that dynamically switches between the single-leader and dual-leader approaches depending on local path curvature would likely yield superior formation stability across a wider variety of path geometries. In such a system, dual-leader formation control would be applied during straight and gently curving segments to minimize lateral deviations between robots, while single-leader control would be used during sharp turns to better coordinate the rotation of the entire formation. Implementing real-time switching logic, along with robust transition handling between modes, remains a promising area for further investigation to enhance coordinated multi-robot navigation in warehouse environments.

References

[1] T. R. R. Staff. (2022) A decade after acquiring kiva, amazon unveils its first amr. Accessed: 2024-04-27. [Online]. Available: https://www.therobotreport.com/a-decade-after-acquiring-kiva-amazon-unveils-its-first-amr/

[2] H. Vella. (2024) Robots teach themselves to collaborate more effectively. Accessed: 27-Apr-2025. [Online]. Available: https://www.iotworldtoday.com/robotics/robots-teach-themselves-to-collaborate-more-effectively

[3] S. Liu, D. Sun, and C. Zhu, "Coordinated motion planning for multiple mobile robots along designed paths with formation requirement," *IEEE/ASME Transactions on Mechatronics*, vol. 16, no. 6, pp. 1021–1031, 2011.

[4] X. Yan, D. Xu, Y. Chen, J. Lv, and H. Hong, "Multi-robot formation based on trajectory design and model predictive control," in *2022 IEEE International Conference on Mechatronics and Automation (ICMA)*, 2022, pp. 692–697.

[5] L. Chen and M. Baoli, "A nonlinear formation control of wheeled mobile robots with virtual structure approach," in *2015 34th Chinese Control Conference (CCC)*, 2015, pp. 1080–1085.

[6] D. Q. Mayne, "Model predictive control: Recent developments and future promise," *Automatica*, vol. 50, no. 12, pp. 2967–2986, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0005109814005160

[7] Q. Lu, Y. Xu, S. Wang, C.-Y. Su, X. Wu, and F. Guo, "Model predictive control for leader-follower formation control of multiple mobile robots using a modified c/gmres method," in *2023 6th International Conference on Intelligent Robotics and Control Engineering (IRCE)*, 2023, pp. 68–72.

[8] X. Chen, "A gradient-like algorithm for decentralized formation control," *CoRR*, vol. abs/1412.6726, 2014. [Online]. Available: http://arxiv.org/abs/1412.6726