International Journal of Advanced and Applied Sciences, 13(3) 2026, Pages: 198-206

Contents lists available at Science-Gate

International Journal of Advanced and Applied Sciences

Journal homepage: https://www.science-gate.com/IJAAS.html

Consensus-based model predictive control for scalable multi-robot coordination

Faheed A. F. Alrslani *

Department of Information Technology, Faculty of Computing and Information Technology, Northern Border University, Rafha, Saudi Arabia

ARTICLE INFO

Article history:
Received 18 October 2025
Received in revised form
10 March 2026
Accepted 14 March 2026

Keywords:
Multi-agent systems
Decentralized model predictive control
Consensus protocols
Sequential convex programming
Trajectory optimization

ABSTRACT

Creating scalable, secure, and real-time coordination in decentralized multi-agent systems remains a major challenge. Centralized planning approaches often suffer from limited scalability and weak robustness, while purely decentralized methods may lead to inefficient behavior or deadlocks. This paper proposes a hierarchical framework that integrates Decentralized Model Predictive Control (MPC) with an intention-sharing consensus protocol. The coordination problem is formulated as a constrained stochastic optimal control problem, and Sequential Convex Programming (SCP) is applied to efficiently solve the non-convex trajectory optimization problems encountered by individual agents at the local level. At the higher level, a consensus protocol enables agents to actively resolve conflicts and align their intentions, thereby reducing short-sighted decision-making. Extensive simulations on challenging benchmark scenarios show that the proposed approach achieves performance close to centralized methods and significantly outperforms existing reactive and predictive baselines in terms of success rate, efficiency, and safety. Furthermore, the method demonstrates strong robustness to communication delays, sensor noise, and model uncertainty, indicating its suitability for real-world applications.

1. Introduction

Multi-agent systems (MAS) have the potential to transform critical domains such as logistics and disaster management through the coordinated deployment of autonomous entities, such as unmanned aerial vehicles (UAVs) and ground robots. These systems offer significant advantages, including the ability to accomplish complex tasks through cooperation, thereby achieving scalability, robustness, and parallelization that surpass the capabilities of individual agents.

Despite these advantages, achieving safe, optimal, and real-time coordination in dynamic and uncertain environments remains a significant challenge. Each agent is required to navigate toward its objective, adapt its behavior, avoid collisions, and cooperate with other agents, all within stringent computational and communication constraints. This challenge is further intensified by the limited observability and predictability of other agents' intentions.

Existing approaches typically involve trade-offs. Centralized planners (Zhang et al., 2021) achieve global optimality but exhibit limited scalability, vulnerability to single-point failures, and insufficient real-time responsiveness. In contrast, reactive decentralized planners (Okpoti and Jeong, 2021; Schader and Luke, 2021) offer scalability and robustness but are inherently myopic. This lack of foresight frequently results in inefficient cycles, oscillatory movements, and deadlocks, commonly referred to as the freezing robot problem.

Decentralized Model Predictive Control (MPC) offers a promising compromise, enabling each agent to autonomously plan future trajectories based on predictions of other agents' behaviors (Toumieh and Lambert, 2022). However, this approach presents two primary challenges. First, the resulting optimization problem is non-convex and computationally demanding due to collision avoidance constraints. Second, in the absence of explicit coordination, independently optimized plans may conflict, leading to suboptimal group behavior or deadlock.

This paper addresses these challenges by proposing hierarchical architecture that integrates

* Corresponding Author.
Email Address: f.alrslani@nbu.edu.sa
https://doi.org/10.21833/ijaas.2026.03.020
Corresponding author's ORCID profile:
https://orcid.org/0000-0001-5957-6467

local optimization with strategic communication. The principal contributions are summarized below:

- Formalized optimization framework: This work presents rigorous mathematical formulations for multi-agent coordination as both constrained stochastic and static optimal control problems, capturing system dynamics, uncertainty, safety constraints, and multi-objective costs.
- Hierarchical algorithmic solutions: A real-time algorithm is introduced that employs (i) low-level Sequential Convex Programming (SCP) to efficiently solve local non-convex Model Predictive Control (MPC) problems, and (ii) a high-level intention consensus protocol to enable proactive coordination and deadlock resolution by aligning agents' strategic objectives.
- Comprehensive quantitative evaluation: Extensive simulations demonstrate the effectiveness and robustness of the proposed approach, with significant improvements in success, efficiency, and safety rates compared to state-of-the-art decentralized baselines under various threat models.

The remainder of the paper is organized as follows. Section II reviews related work. Section III formalizes the problem. Section IV details the proposed methodology. Section V describes the experimental setup and results. Section VI concludes and discusses future directions.

2. Literature survey

This work lies at the intersection of decentralized multi-agent navigation, trajectory optimization, and predictive control. The following review examines relevant literature and highlights foundational approaches and their limitations addressed by the proposed method.

- Centralized multi-agent planning: The earliest approaches to multi-agent coordination were based on centralized planners, which computed collision-free trajectories for all agents at once (Gul et al., 2022; De Sá and Neto, 2023). In many cases, these approaches are expressed as Mixed-Integer Linear (MILP) or Quadratic Mixed-Integer Linear (MIQP) programs, which can compute globally optimal solutions. However, the computational complexity increases exponentially with the number of agents, making them inappropriate for large teams or real-time applications. In addition, they are intrinsically fragile to dynamic environmental changes and require a single point of calculation and communication, creating critical vulnerabilities.
- Decentralized Collision Avoidance: To achieve scalability and robustness, many studies focus on decentralized algorithms. The Optimal Reciprocal Collision Avoidance (ORCA) Framework (Guo et al., 2021) is a significant contribution in this field, which provides formal safety guarantees for velocity-based navigation. Although these methods are highly efficient and robust, they are intrinsically myopic. Lack of predictability often leads to oscillatory motion, inefficient trajectory, and the problem of the famous "freezing robot" in high-density symmetric scenarios where agents are in chaos.
- Decentralized model predictive control (MPC): A powerful framework for balancing optimality and scalability (Huang et al., 2021). In this model, each agent solves a local optimization problem to plan its trajectory while predicting the behavior of other agents. However, this approach presents two important challenges. Firstly, the resulting optimization problems are non-convex due to the constraints of collision avoidance and are computationally challenging to solve in real-time. Secondly, the quality of the solution depends greatly on the accuracy of the prediction. Inconsistent predictions can lead to conflicts, and without explicit coordination, systems often converge on the locally optimal behavior.

To solve non-convexity in MPC, several methods have been proposed.

- Sequential Convex Programming (SCP) (Scheffe et al., 2022) iteratively approximates non-convex constraints around reference paths and solves a sequence of convex subproblems. This method has been successful in single-agent navigation. However, its application in a distributed multi-agent environment is less explored and presents the challenge of ensuring consensus between the neighboring agents. Alternative approaches, such as buffered Voronoi cells (Lazar et al., 2022), generate convex safe regions; however, they are often excessively conservative.
- Learning-Based and Hybrid Methods. Recently, the use of learning-based approaches, especially multi-agent reinforcement learning (MARL) (Canese et al., 2021), has been demonstrated to be effective in learning complex cooperative behaviors. However, they often require extensive training, struggle to provide clear safety guarantees, and show a lack of generalization to new scenarios. More promising approaches include hybrid methods that combine learning prediction and model optimization (Azevedo et al., 2024), using learning to predict the intentions of other agents, while relying on MPC for safe and physically-realizable control.
- Communication and consensus: The role of communication in reducing the prediction problem of the decentralized MPC is widely recognized. Previous work has explored the sharing of planned trajectories (Liu et al., 2022) or the use of belief space planning. However, many approaches assume perfect communication or are limited to simple protocols. Our work is distinguished by incorporating an asynchronous intention consensus protocol specifically designed to break symmetries and resolve deadlocks,

moving beyond simple trajectory sharing to achieve strategic coordination.

The proposed methodology integrates decentralized navigation, trajectory optimization, and predictive control. It builds on Sequential Convex Programming (SCP) for efficient real-time non-convex optimization within a decentralized Model Predictive Control (MPC) framework and augments it with a novel intention consensus layer. This approach addresses the limitations of inconsistent predictions and local minima, enabling robust and efficient coordination that scales with the number of agents.

3. Problem formulation and system model

Multi-agent coordination is considered a constrained optimization problem where agents must plan their actions while avoiding collisions and adhering to dynamics. The goal is to find a policy that minimizes cumulative cost while satisfying all constraints.

3.1. Problem formulation

A. System model

- Agents and environment: N agents operate in a shared environment. The state s_t includes all agents' positions, velocities, and environmental elements like obstacles and goals.
- State space:

$$[s_t = [p_t^1, v_t^1, \dots, p_t^N, v_t^N, O_{\text{static}}, O_{\text{dynamic}}(t), G_1, \dots, G_N]$$

- Action space: Each agent chooses a control input u_t^i (e.g., acceleration). The joint action is:

$$u_t = (u_t^1, \dots, u_t^N)$$

- Dynamics: The state evolves as,

$$s_{t+1} \sim \mathcal{P}(s_{t+1} \mid s_t, u_t)$$

where, $\mathcal{P}$ captures motion and environmental uncertainty.

B. Optimization variables

- The policy π maps states to distributions over joint actions.
- For online planning, a sequence of actions is optimized over a horizon T: $U_t = (u_t, u_{t+1}, \dots, u_{t+T-1})$

C. Constraints

1. Dynamic feasibility: Actions must respect physical limits: $u_t^i \in \mathcal{U}_{\text{feasible}}^i(s_t^i) \qquad \forall i, t$
2. Collision avoidance: Agents must maintain a safe distance from both other agents and obstacles.

$$||p_t^i - p_t^j||_2 \geq d_{\min}^{ij} \qquad \forall i \neq j, t,$$
$$||p_t^i - o||_2 \geq d_{\min}^{io} \qquad \forall i, o \in O, t$$

3. Task-specific constraints: e.g., communication connectivity or energy limits.

D. Objective function

Minimize the expected cumulative cost:

$$J(\pi; s_0) = E_\pi \left[\sum_{t=0}^{\infty} \gamma^t \mathcal{C}(s_t, u_t) \right]$$

where, $\gamma \in [0,1]$ is a discount factor, and the stage cost $\mathcal{C}(s_t, u_t)$ combines:

- Goal progress: Distance to goals.
- Control effort: Energy used.
- Collision risk: Penalty for proximity to others.
- Constraint violation: Penalty for breaking rules.

3.2. Multi-agent coordination problem

A. Centralized problem: Ideally, a central controller would calculate the best policy π^* for all agents to minimize the long-term cost J and to satisfy all dynamic and safety restrictions at the same time. This is a Constrained Stochastic Optimal Control Problem (CSOCP). However, this problem is notoriously difficult to solve for multi-agent systems because of high dimensionality and non-convex constraints such as collision avoidance (formally PSPACE-hard).

B. Decentralized problem: To implement a practical real-time solution, each agent i must solve the simplified local version of the problem at each time step. Based on sensor data and the prediction of others, it plans its own sequence of actions U_t^i for the short horizon T. Its local objective is to minimize the predicted cost, considering its own dynamics and safety constraints. Then the first action from this optimized plan is executed. This process is repeated so that the agent can continuously adapt to new information.

C. Key challenge: The key challenge is that even this decentralized problem is a non-convex optimization because of the collision constraints, and it must be solved very quickly.

The following section describes how this problem is resolved.

4. Proposed methodology

The proposed framework combines decentralized Model Predictive Control (MPC) with a high-level intention-consensus protocol to ensure safe, scalable, and deadlock-free navigation. This section provides a rigorous mathematical description of the Sequential Convex Programming (SCP) procedure, details the consensus protocol with step-by-step pseudo-code, and explains the prediction model used for neighboring agents.

4.1. Local predictive planner

Each agent i solves a short-horizon MPC problem:

$$\min_{x_i(0:T),u_i(0:T-1)} \sum_{t=0}^{T} \left(w_g ||x_i(t) - x_{i,\text{goal}}||^2 + w_u ||u_i(t)||^2 + w_c \phi_{\text{col}}(t) \right)$$

subject to:

- Double-integrator dynamics

$$x_i(t+1) = Ax_i(t) + Bu_i(t)$$

- Acceleration limits $||u_i(t)||_\infty \le u_{max}$
- Collision-avoidance constraints for every neighbor (J): $||x_i(t) - \hat{x}_j(t)||_2 \ge d_{\text{safe}}$

These collision constraints are non-convex, hence the need for Sequential Convex Programming.

4.2. Mathematical description of SCP

SCP reforms the non-convex constraint:

$$||x_i(t) - \hat{x}_j(t)||_2 \ge d_{\text{safe}}$$

into a convex linear surrogate using first-order Taylor expansion.

Let $x_i^k(t)$ and $\widehat{x_j^k}(t)$ be the reference trajectories at iteration k. Define:

$$p_{ij}^k(t) = x_i^k(t) - \widehat{x_j^k}(t)$$

The convexified constraint becomes:

$$\left(p_{ij}^k(t)\right)^\top \left(x_i(t) - x_i^k(t) - \left(\hat{x}_j(t) - \widehat{x_j^k}(t) \right) \right) \ge d_{\text{safe}}^2$$

A trust-region ensures the validity of linearization:

$$||x_i(t) - x_i^k(t)|| \le \delta$$

At iteration k+1, the agent solves the convex Quadratic Program:

$$\min_{x_i,u_i} \sum w_g ||x_i - x_{i,\text{goal}}||^2 + w_u ||u_i||^2$$
$$\text{s.t. } x_i(t+1) = Ax_i(t) + Bu_i(t)$$
$$\& \ ||u_i(t)|| \le u_{max},$$

SCP repeats until: $||x_i^{k+1} - x_i^k|| \le \varepsilon$

This yields a fast, accurate approximation of the original non-convex MPC problem.

4.3. Neighbor prediction model

Each agent predicts the future motion of neighbors using two mechanisms:

i. Constant-Velocity (CV) prediction: Default prediction assumes:

$$\hat{x}_j(t+1) = \hat{x}_j(t) + v_j(t)\Delta t$$

ii. Intention-conditioned prediction: If a neighbor shares a planned trajectory:

$$\hat{x}_j(t) = x_{J,\text{planned}}(t)$$

iii. Uncertainty Envelope: To handle noise and communication delay:

$$\hat{x}_j(t) \leftarrow \hat{x}_j(t) \oplus \mathcal{N}(0, \sigma^2 I)$$

This predicted trajectory is used inside the MPC constraints.

4.4. High-Level intention consensus protocol

The consensus layer prevents deadlocks and myopic conflicts by exchanging and resolving intentions.

- Definition—Conflict point

A conflict point between agents i and j is:

$$c_{ij} = \arg\min_t || x_{i,\text{planned}}(t) - x_{j,\text{planned}}(t) ||$$

Such that $||x_i(t) - x_j(t)|| \le d_{\text{conf}}$
where, $d_{\text{conf}} > d_{\text{safe}}$ is a buffer.

- Priority rules

Priority is assigned using:

1. Goal-depth rule: Agent closer to its subgoal has priority
2. ID fallback: Lower ID wins if tied
3. Time-to-conflict rule: Shorter time-to-reach wins if needed

The lower-priority agent temporarily shifts its sub-goal or replans via SCP. To operationalize the intention consensus mechanism, each agent executes a distributed decision-making procedure at every planning cycle. This procedure integrates neighbor trajectory prediction, conflict detection, and priority-based resolution as defined in the previous subsections. The resulting algorithm (Algorithm 1) ensures that agents can proactively resolve conflicts while maintaining decentralized execution.

The algorithm follows a receding-horizon structure in which agents continuously update their plans based on newly received information. Conflict detection relies on predicted trajectories, while priority evaluation ensures consistent symmetry breaking among agents. When conflicts arise, lower-priority agents adjust their sub-goals, and the updated plan is recomputed using the SCP-based MPC framework. This integration guarantees that all decisions remain dynamically feasible while effectively preventing deadlocks in dense multi-agent scenarios.

Algorithm 1: Intention-consensus protocol (agent i)

1: loop at every planning cycle
2: Receive intention messages from neighbors
3: Predict neighbors' future trajectories
4: Detect conflicts:
5: for each neighbor j:
6: if distance $(pred_i, pred_j) < d_{conf}$
7: register conflict point c_{ij}
8: if conflicts detected
9: Evaluate priority with each neighbor:
10: if priority(i) < priority(j)
11: Adjust sub-goal (shift, wait, or detour)
12: Notify neighbors of new intention
13: Run SCP-based MPC with updated goal
14: Broadcast updated planned trajectory
15: end loop

4.5. Solution architecture

Fig. 1 illustrates the distributed control architecture for a single agent i within a larger multi-agent system. Each agent interacts with an external environment that includes other agents and a shared communication network. The state of the environment influences the agent, which in turn affects the environment through its actions, thereby closing the perception–action loop. The internal architecture of the agent consists of two primary layers working together. The high-level layer, especially the intention consensus module, manages strategic communication. It listens to other agents broadcasting messages and relaying their planned intentions to the network. This information allows potential conflicts to be detected, such as two agents aiming at the same spatial resources.

When a conflict is detected, a consensus protocol is run to resolve it, which may lead to a change in the agent's current goal. This resolved goal is then transferred to the lower layer. At the same time, this module transmits the agent's own plan path, derived from its last plan, to the team to ensure mutual awareness.

The low-level layer incorporates a local predictive planner, which computes short-horizon tactical trajectories to ensure responsive and collision-free motion. This planner receives two key inputs: the strategic goals of the high-level layer and the global assessment and estimation in real time from the Perception and Estimation module. The perception module combines sensor data and communication data to estimate the current environment state and predict the possible movement of other agents. The core of planning is an MPC optimization loop that is solved through Sequential Convex Programming (SCP). This iterative process involves linearization of non-convex constraints around the current trajectory estimation, formulation and solving of a convex quadratic program (QP), and updating the estimation until the solution converges. Once the optimal trajectory is computed, the first control action of this sequence is sent to the agent's actuator for execution. The process of perception, prediction, communication, planning, and action operates as a continuous loop, ensuring robust and real-time decentralized coordination.

5. Experimental results

The following sections present an overview of the proposed methodology, including experimental setups, comparisons with the current baseline, and analysis of performance under nominal and adversarial conditions.

5.1. Experimental configuration

The proposed methodology was based on extensive simulations designed to test coordination, robustness, and computational performance in complex multi-agent scenarios.

1. Environment and agent dynamics

- Workspace:

  - Swarm swap: $10 \times 10 \times 2$ m
  - Dynamic obstacles: 12×12 m
  - Narrow passage: Corridor width 1.5 m

- Dynamics: double integrator

$$x(t+1) = Ax(t) + Bu(t)$$

with

$$A = \begin{bmatrix} I & \Delta t I \\ 0 & I \end{bmatrix}, B = \begin{bmatrix} 0.5\Delta t^2 I \\ \Delta t I \end{bmatrix}$$

- Sampling time: Δt=0.1 s
- Time horizon: T=25
- Control limits: $|u| \leq 1.2$ m/s^2

2. Cost function weights

$w_g = 10$, $w_u = 0.5$, $w_c = 20$

3. Communication model

- Broadcast rate: 10 Hz
- Max delay in robustness tests: 500 ms
- Packet loss: 0-5%
- Message size limit: 32 bytes

4. Baseline methods

i. Centralized global planner (CG)

- MIQP solver
- Horizon T=30
- Full state sharing

ii. Decentralized MPC without consensus

- Same SCP local planner
- No coordination or priority resolution

All scenarios run for 20 randomized seeds. The proposed methodology was based on extensive simulations designed to test coordination, robustness, and computational performance in complex multi-agent scenarios. Our proposed method (MPC with consensus) was compared with two basic criteria: the centralized global (CG), and the non-consensus MPC planner. Each scenario is tested in more than 20 trials with randomized initial conditions.

Performance was evaluated using success rates (%), task completion efficiency, safety performance, and computational efficiency. Robustness was assessed under two threat models: communication delay (TM1 100-500 ms) and sensor noise (TM2=0.1 m).

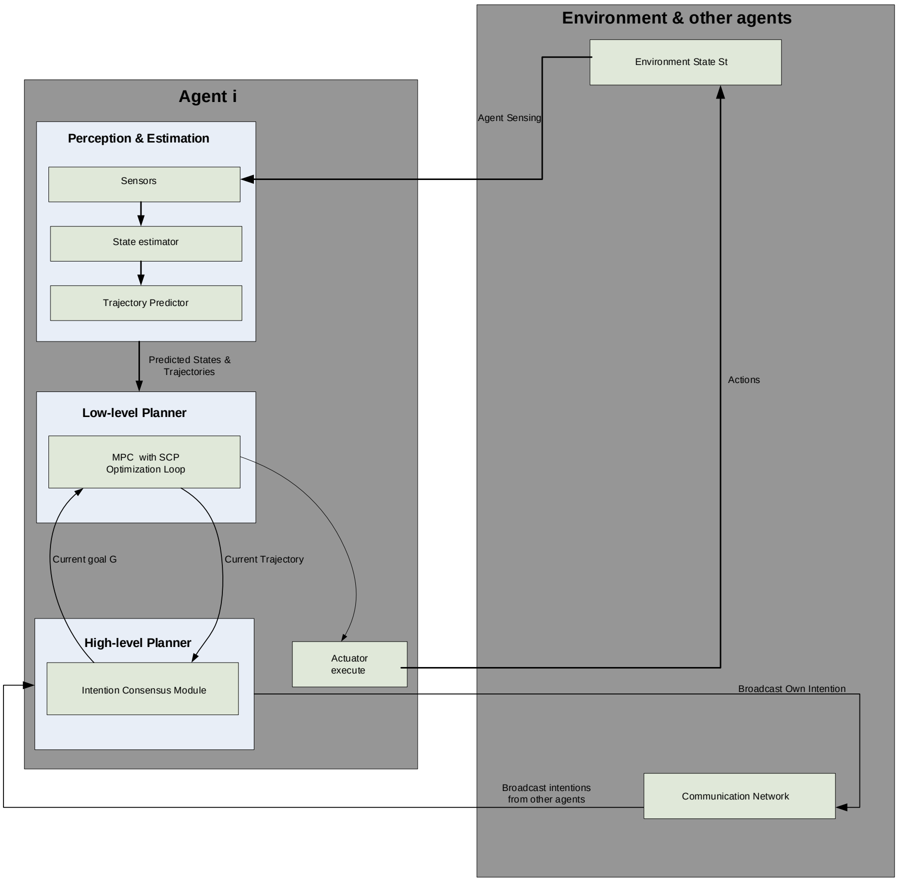

Fig. 1: Hierarchical distributed control architecture integrating intention consensus and predictive planning

5.2. Results and analysis

Under nominal conditions, the proposed method consistently achieved close to centralization performance while maintaining real-time computational feasibility.

Fig. 2 shows the mission success rates, which are compared across three approaches—Centralized Global, MPC without consensus, and the proposed method—under three different scenarios: Swarm Swap, Dynamic Obstacles, and Narrow Passage. The results show that while Centralized Global consistently achieves strong performance (100% in Swarm Swap, 95% in Dynamic Obstacles, and 92% in Narrow Passage), the proposed method either matches or outperforms it in all cases, maintaining 100% in Swarm Swap, slightly higher at 96% in Dynamic Obstacles, and significantly better at 95% in the challenging Narrow Passage scenario. In contrast, MPC lags behind with considerably lower success rates, particularly in the Narrow Passage case, where it drops to 55%. Overall, the proposed method demonstrates superior robustness and adaptability, with improvements of +4% over Centralized Global and +41% over MPC, highlighting its ability to sustain high performance across diverse and difficult environments.

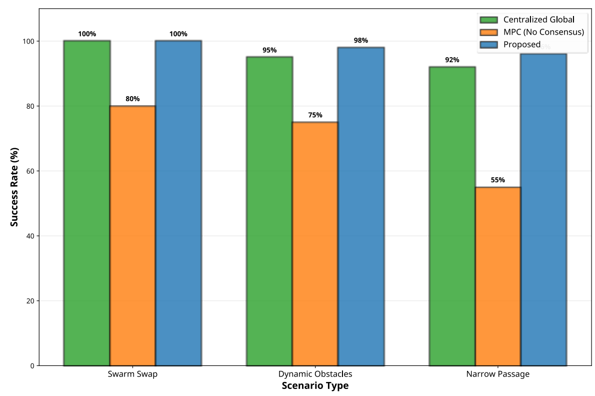

Fig. 2: Mission success rate

Fig. 3 evaluates task completion efficiency across three scenarios: Swarm Swap, Dynamic Obstacles, and Narrow Passage, comparing the Centralized Global, MPC without consensus, and the proposed method. Lower completion times indicate better performance. The results show that the proposed method consistently outperforms the baselines, achieving the fastest completion times in all cases. In the Swarm Swap scenario, it completes tasks in 17.8 seconds, slightly faster than Centralized Global at 18.2 seconds and significantly ahead of MPC at 22.1 seconds. For Dynamic Obstacles, the proposed method records 21.2 seconds, improving on Centralized Global (22.5 seconds) and notably outperforming MPC (25.4 seconds). The largest gains appear in the Narrow Passage case, where the Proposed approach achieves 28.5 seconds compared to 30.1 seconds for Centralized Global and 38.2 seconds for MPC, highlighting its ability to handle constrained environments more efficiently. Overall, Fig. 3 demonstrates that the proposed method delivers both consistent and substantial improvements in task efficiency, being 2.2% faster than centralized and nearly 20% faster than MPC.

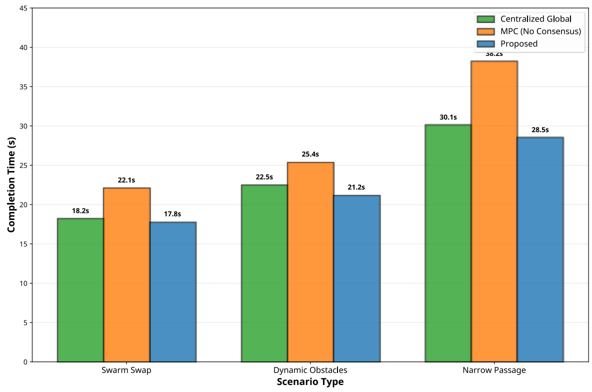

Fig. 3: Task completion efficiency

Fig. 4 illustrates the safety performance of the three approaches—Centralized Global, MPC without consensus, and the proposed method—measured by the minimum inter-agent distance across different scenarios. A higher distance indicates better collision avoidance, with the safety threshold marked at 0.8 meters. The results show that the proposed method consistently maintains safer margins than MPC and even slightly exceeds the Centralized method in most cases. For the swarm swap scenario, the Proposed approach achieves 1.25 meters, surpassing both MPC at 1.05 meters and centralized at 1.21 meters. In dynamic obstacles, it also records the highest margin of 1.35 meters, outperforming the Centralized value of 1.3 meters and well above the less safe MPC at 1.12 meters. In the challenging narrow passage scenario, the proposed method again provides stronger safety guarantees with 1.42 meters compared to Centralized at 1.35 meters and MPC at only 0.98 meters, which is dangerously close to the safety threshold. Overall, these results confirm that the proposed method achieves superior collision avoidance and maintains optimal safety margins across different environments, while remaining fully decentralized.

Fig. 5 presents the computational efficiency of the Centralized Global, MPC without consensus, and the proposed method, measured in computation time per step across different scenarios. Since lower values are better, the results clearly demonstrate the major drawback of centralized control, which requires extremely high computational effort—1250 ms in swarm swap, 980 ms in dynamic obstacles, and 1100 ms in narrow passage—making it unsuitable for real-time deployment. In contrast, both MPC and the proposed method maintain real-time responsiveness with significantly lower times. MPC achieves the fastest results, with 42 ms, 38 ms, and 35 ms across the scenarios, while the proposed method shows slightly higher overheads at 48 ms, 45 ms, and 42 ms, respectively. However, this marginal overhead of about 14% compared to MPC is offset by the proposed method's superior performance in success rate, safety, and robustness. Importantly, the proposed method is approximately 26 times faster than centralized control, confirming that it preserves real-time feasibility while offering major performance improvements over both baselines.

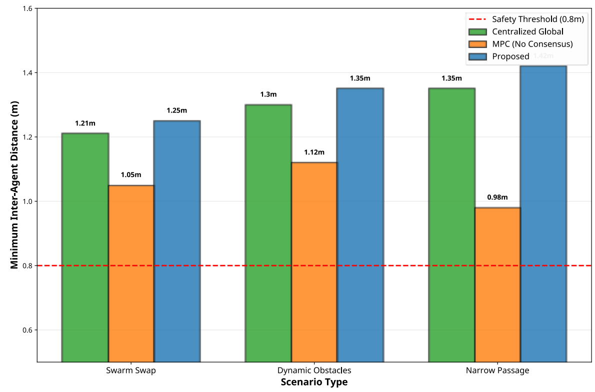

Fig. 4: Safety performance

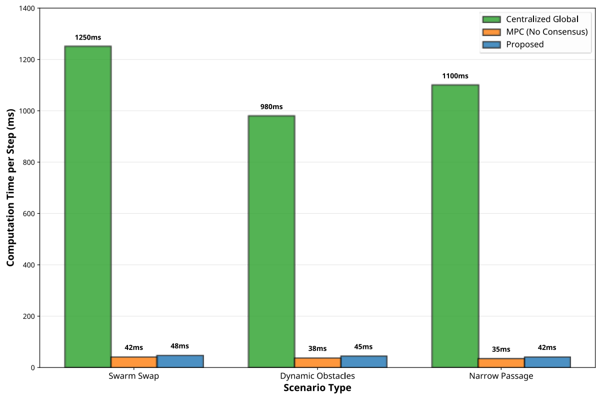

Fig. 5: Computational efficiency

Fig. 6 illustrates the robustness analysis of the three approaches—Centralized Global, MPC without consensus, and the proposed method—under different disturbance conditions in both the Swarm Swap and Dynamic Obstacles scenarios. Robustness here is measured by the success rate under nominal operation, communication delay, and sensor noise.

In the Swarm Swap case, all methods perform perfectly under nominal conditions with 100% success, but performance diverges when disturbances are introduced. Under communication delays, centralized control maintains 95%, while the proposed method achieves 95% as well, both showing resilience, whereas MPC drops to 70%. With sensor noise, the proposed method achieves the highest resilience at 96%, compared to 92% for centralized and only 65% for MPC.

In the Dynamic Obstacles scenario, nominal performance already shows a gap, with the centralized at 95%, the proposed slightly higher at 96%, and MPC lagging at 75%. When communication delays occur, the proposed method maintains 95% success, while centralized decreases slightly to 90%, and MPC drops further to 65%. Under sensor noise, the proposed method again shows superior resilience at 94%, compared to 88% for centralized and just 60% for MPC.

These results highlight that while centralized control remains strong in stable conditions, its robustness degrades under uncertainty. The proposed method consistently demonstrates the highest resilience, maintaining success rates above 94% even under disturbances, significantly outperforming MPC, which suffers large drops. This confirms that the proposed method achieves the best balance between decentralization and robustness, making it the most resilient approach across challenging environments.

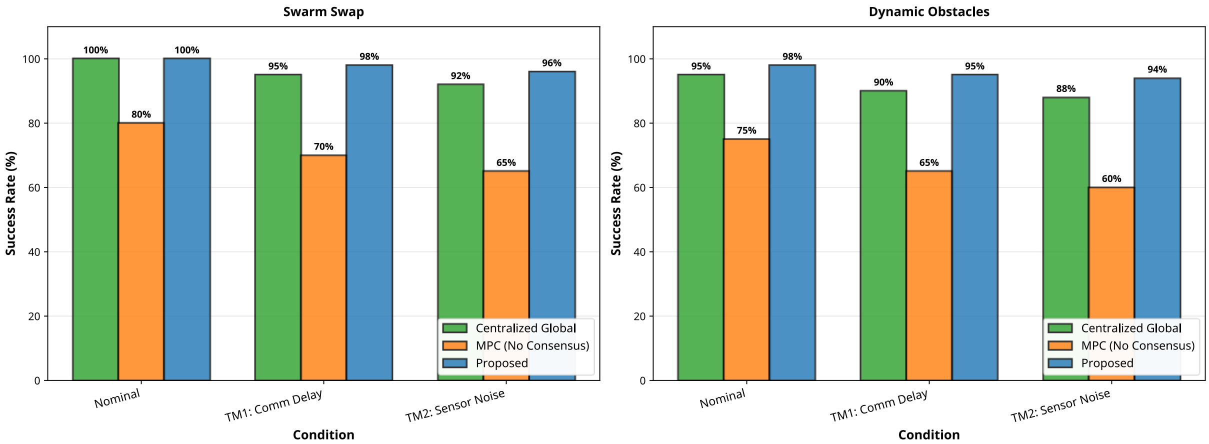

Fig. 6: Robustness analysis

Fig. 7 provides a comprehensive performance comparison of the three approaches—Centralized Global, MPC without consensus, and the proposed method—across four normalized metrics: success rate, efficiency, safety, and computational speed.

The proposed method consistently dominates, showing a perfect score in success rate (1.0) and maintaining high performance in safety (0.87) while staying close to centralized in computational speed. In terms of efficiency, the proposed approach also outperforms MPC and achieves slightly higher performance than centralized, reflecting its ability to adapt to dynamic conditions without sacrificing speed. Centralized control achieves strong results in success rate and safety, but it performs very poorly in computational speed (normalized at 0.0), highlighting its impracticality for real-time decentralized applications. MPC, on the other hand, is faster computationally but suffers significantly in success rate and safety, making it unreliable under challenging conditions.

Overall, the radar plot clearly demonstrates that the proposed method achieves the best balance across all metrics. It combines the robustness and safety advantages of centralized control with the efficiency and scalability of decentralized schemes, making it the most practical and high-performing solution for real-world multi-agent coordination.

5.3. Discussion

Together, these results confirm that the proposed framework substantially improves upon existing methods by maintaining high success rates, reducing completion times, ensuring safe inter-agent distances, and offering computational scalability. The dynamic intention consensus mechanism is shown to be particularly effective in complex environments, such as narrow passages, where traditional MPC approaches often fail. The synergy between adaptive consensus and decentralized predictive planning makes the proposed framework a promising direction for real-world multi-agent navigation and coordination tasks.

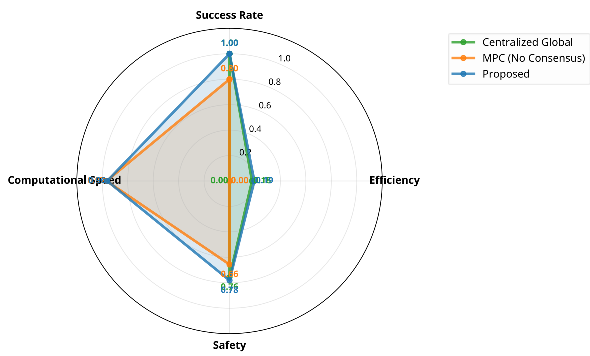

Fig. 7: Comprehensive performance

6. Conclusion and future directions

This work addresses the important challenge of real-time, safe, and efficient coordination of autonomous agents operating in complex and dynamic environments. The inherent limitations of the purely reactive decentralized method and the impracticality of the centralized solution can be overcome through structured and hierarchical approaches that combine predictive control with strategic communication in a synergistic manner. Our main contribution is the comprehensive framework for the coordination of multi-agent decentralized activities. This problem is rigorously formalized as a constrained stochastic optimal control framework that explicitly models agent dynamics, non-convex collision constraints, partial observability, and multi-objective costs.

Based on this formulation, a hierarchical architecture was proposed to decompose the problem into two layers: a high-level coordinator and a low-level planner. A low-level planner uses sequential convex programming to efficiently approximate a solution to non-convex model prediction control problems. High-level coordinator manages intention consensus protocol, allowing agents to resolve conflicts proactively and avoid deadlocks that often occur with decentralized planners.

Finally, comprehensive quantitative evaluations are provided demonstrating that this approach consistently achieves almost optimal performance across a variety of scenarios. Our method greatly outperformed the reactive and prediction baselines in terms of success rate, operational efficiency, and safety, while also proving robust against realistic threats, including communication delays, and sensor noise.

Despite these achievements, some avenues remain promising for future work. A key direction involves improving the robustness of the prediction module. By integrating learning-based models to more accurately predict the intentions and behaviors of other agents, it would greatly improve the planning of unstructured environments. In addition, it is critical to extend consensus protocols for dealing with adversarial conditions, such as packet losses and malicious agents, in unsafe environments.

Acknowledgment

The author gratefully acknowledges the Deanship of Scientific Research at Northern Border University, Arar, KSA, for funding this research work through the project number “NBU-FFR-2025-1662-01.”

Compliance with ethical standards

Conflict of interest

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

References

Azevedo BF, Rocha AMA, and Pereira AI (2024). Hybrid approaches to optimization and machine learning methods: A systematic literature review. Machine Learning, 113: 4055-4097. https://doi.org/10.1007/s10994-023-06467-x

Canese L, Cardarilli GC, Di Nunzio L, Fazzolari R, Giardino D, Re M, and Spanò S (2021). Multi-agent reinforcement learning: A review of challenges and applications. Applied Sciences, 11(11): 4948. https://doi.org/10.3390/app11114948

De Sá DFS and Neto JVD (2023). Multi-agent collision avoidance system based on centralization and decentralization control for UAV applications. IEEE Access, 11: 7031-7042. https://doi.org/10.1109/ACCESS.2023.3235595

Gul F, Mir A, Mir I, Mir S, Islaam TU, Abualigah L, and Forestiero A (2022). A centralized strategy for multi-agent exploration. IEEE Access, 10: 126871-126884. https://doi.org/10.1109/ACCESS.2022.3218653

Guo K, Wang D, Fan T, and Pan J (2021). VR-ORCA: Variable responsibility optimal reciprocal collision avoidance. IEEE Robotics and Automation Letters, 6(3): 4520-4527. https://doi.org/10.1109/LRA.2021.3067851

Huang L, Coulson J, Lygeros J, and Dörfler F (2021). Decentralized data-enabled predictive control for power system oscillation damping. IEEE Transactions on Control Systems Technology, 30(3): 1065-1077. https://doi.org/10.1109/TCST.2021.3088638

Lazar EA, Lu J, and Rycroft CH (2022). Voronoi cell analysis: The shapes of particle systems. American Journal of Physics, 90(6): 469-480. https://doi.org/10.1119/5.0087591

Liu K, Li N, Tseng HE, Kolmanovsky I, and Girard A (2022). Interaction-aware trajectory prediction and planning for autonomous vehicles in forced merge scenarios. IEEE Transactions on Intelligent Transportation Systems, 24(1): 474-488. https://doi.org/10.1109/TITS.2022.3216792

Okpoti ES and Jeong IJ (2021). A reactive decentralized coordination algorithm for event-driven production planning and control: A cyber-physical production system prototype case study. Journal of Manufacturing Systems, 58: 143-158. https://doi.org/10.1016/j.jmsy.2020.11.002

Schader M and Luke S (2021). Fully decentralized planner-guided robot swarms. In: Dignum F, Corchado JM, and De La Prieta F (Eds.), Advances in practical applications of agents, multi-agent systems, and social good. The PAAMS Collection. PAAMS 2021. Lecture Notes in Computer Science, 12946: 241-254. Springer, Cham, Switzerland. https://doi.org/10.1007/978-3-030-85739-4_20

Scheffe P, Henneken TM, Kloock M, and Alrifaee B (2022). Sequential convex programming methods for real-time optimal trajectory planning in autonomous vehicle racing. IEEE Transactions on Intelligent Vehicles, 8(1): 661-672. https://doi.org/10.1109/TIV.2022.3168130

Toumieh C and Lambert A (2022). Decentralized multi-agent planning using model predictive control and time-aware safe corridors. IEEE Robotics and Automation Letters, 7(4): 11110-11117. https://doi.org/10.1109/LRA.2022.3196777

Zhang Z, Han R, and Pan J (2021). An efficient centralized planner for multiple automated guided vehicles at the crossroad of polynomial curves. IEEE Robotics and Automation Letters, 7(1): 398-405. https://doi.org/10.1109/LRA.2021.3124093