TELKOMNIKA Telecommunication Computing Electronics and Control
Vol. 24, No. 3, June 2026, pp. 1037~1047
ISSN: 1693-6930, DOI: 10.12928/TELKOMNIKA.v24i3.27589

Layered virtual-leader DMPC with SQP for scalable V formation tracking of omni-robots in cluttered maps

Tuan Phu Duong[1], Vinh Quang Nguyen[1], Minh Tuan Nguyen[2]
[1]Academy of Military Science and Technology, Ha Noi, Viet Nam
[2]Department of Engineering and Technology, Faculty of International Training, Thai Nguyen University of Technology, Thai Nguyen, Viet Nam

Article Info

Article history:

Received Oct 6, 2025
Revised Feb 7, 2026
Accepted Mar 29, 2026

Keywords:

Distributed model predictive control
Formation control
Omnidirectional mobile robots
Sequential quadratic programming
Virtual-leader

ABSTRACT

This paper proposes a hierarchical virtual leader based distributed model predictive control (DMPC) framework for V-formation control of omnidirectional mobile robots in static obstacle environments. The physical leader follows a predesigned reference trajectory, while the followers maintain the desired formation through distributed optimization. A layered communication topology is established, where only a subset of robots receives the leader's predicted states and acts as virtual leaders for downstream followers. Each robot independently optimizes its control sequence using local neighbor information, enabling fully distributed coordination without centralized synchronization. The unified cost function considers formation maintenance, leader or virtual leader tracking, obstacle avoidance, and control effort. Static obstacles are represented on a grid map to ensure collision-free motion. Simulation results demonstrate that the proposed framework achieves accurate formation keeping, smooth trajectory tracking, and effective obstacle avoidance. The hierarchical virtual-leader architecture enhances scalability, coordination efficiency, and robustness for multi-robot formation systems.

Corresponding Author:

Minh Tuan Nguyen
Department of Engineering and Technology, Faculty of International Training
Thai Nguyen University of Technology
No. 666, 3-2 Street, Tich Luong Ward, Thai Nguyen Province, Viet Nam
Email: nguyentuanminh@tnut.edu.vn

1. INTRODUCTION

Model predictive control (MPC) is an optimization-based control strategy that utilizes a system's dynamic model to compute control inputs, allowing effective handling of system constraints. When applied to multi-robot or distributed systems, distributed model predictive control (DMPC) enables each robot to optimize its behavior based on local information and limited communication with neighbors, thus enhancing the scalability and flexibility of the overall system [1]-[3].

In the context of robotic formation control, DMPC facilitates formation maintenance, trajectory tracking, obstacle avoidance, and performance optimization in environments with complex constraints and uncertainties [2], [4]. However, the optimization problem in DMPC is often computationally challenging and prone to feasibility issues, especially when nonlinear constraints such as obstacle avoidance and sensor noise are present [5], [6]. Sequential quadratic programming (SQP) is one of the most effective nonlinear optimization techniques, and has been employed to improve convergence speed and ensure feasibility in DMPC problems, even under complex constraints [5]. Recent studies have also integrated noise filtering techniques such as the Kalman filter and dynamic occupancy grids to enhance system reliability and safety in

real-world deployments [6], [7]. In summary, combining DMPC and SQP offers a promising approach for mobile robot formation control, simultaneously addressing the requirements of decentralization, optimization, feasibility, and reliability in complex environments.

This paper proposes a method that integrates DMPC with SQP to effectively solve the problem of omnidirectional mobile robot formation control with static obstacle avoidance in noisy environments. The approach leverages the distributed constraint-handling capability of DMPC and the fast, stable convergence of SQP, while employing a multi-objective cost function to simultaneously maintain formation, avoid obstacles, and track the virtual leader trajectory efficiently. Experimental results demonstrate that the proposed method significantly improves control performance, reduces collisions, and maintains formation accuracy in complex environments opening up new directions for future mobile robotic systems.

The remainder of this paper is organized as follows. Section 2 presents the system model, including robot kinematics and the hierarchical virtual-leader communication structure. Section 3 describes the formulation of the DMPC cost function and the SQP-based optimization framework and implementation details. Section 4 provides simulation results that evaluate the proposed method in terms of formation accuracy and obstacle avoidance under measurement noise. Finally, section 5 includes conclusions and discussion of future research directions involving dynamic obstacle scenarios.

2. PRELIMINARIES

2.1. Robot model

In this study, we consider omnidirectional mobile robots operating in a two-dimensional plane. Each robot is equipped with omni or mecanum wheels, which allow instantaneous motion in any direction without the need to reorient the body. The configuration (pose) of robot i is described by the state vector:

$$z_i = [x_i \; y_i \; \theta_i]^\intercal \tag{1}$$

where $(x_i, y_i) \in \mathbb{R}$ denotes the position coordinates in the global reference frame, and $\theta_i \in \mathbb{R}$ is the heading angle of the robot expressed in radians.

The corresponding control input vector is given by:

$$v_i = \left[v_{x,i} \; v_{y,i} \; w_i\right]^\intercal \tag{2}$$

where $v_{x,i}$, $v_{y,i}$ are the linear velocities along the body-frame axes of the robot, and w_i is the angular velocity around the vertical axis.

The kinematic model of each omnidirectional mobile robot is then given by:

$$\dot{z}\begin{bmatrix}\dot{x}_i\\ \dot{y}_i\\ \dot{\theta}_i\end{bmatrix} = \underbrace{\begin{bmatrix}cos\theta_i & -sin\theta_i & 0\\ sin\theta_i & cos\theta_i & 0\\ 0 & 0 & 1\end{bmatrix}}_{R(\theta_i)}\begin{bmatrix}v_{x,i}\\ v_{y,i}\\ w_i\end{bmatrix} \tag{3}$$

where $R(\theta_i)$ is the rotation matrix that maps the control inputs from the body frame to the world frame. The first two rows describe the translational motion in the global *x*-*y* plane, while the third row directly relates the angular rate w_i to the heading dynamics $\dot{\theta}_i$, widely used in omni-directional robot modeling [8]-[10].

For the subsequent predictive control formulation, the continuous-time model (3) is discretized with a sampling time T_s using a forward Euler scheme:

$$z_i(h+1) = z_i(h) + T_s f\big(z_i(h), v_i(h)\big), h = 0,1, \dots, H-1 \tag{4}$$

where $f(\cdot)$ is obtained from the right-hand side of (3), and *H* denotes the prediction horizon. Throughout the paper, all internal computations use θ_i in radians; heading errors shown in the simulation figures are converted to degrees only for visualization. The signed heading error is wrapped to the interval $[-\pi, \pi]$ using a standard wrap-to-π operation and is later used in the DMPC cost function.

2.2. V-formation and communication structure

In the proposed model, the omnidirectional mobile robots are organized into a V-shaped formation, consisting of a single leading robot and multiple follower robots symmetrically distributed on both sides. The V-formation is organized with a leader at the apex following a predefined trajectory, and followers

symmetrically positioned along two branches at fixed offsets. A hierarchical virtual-leader structure is established: the leader shares its state $(x_L, y_L\theta_L)$ only with two immediate followers, which act as virtual leaders for downstream robots. Each robot maintains relative positioning based on local neighbor information, ensuring formation stability under a fully distributed control scheme.

This communication structure can be represented by a directed graph, where edges indicate the direction of state information flow [11]-[13]:

$$\mathcal{G} = (\mathcal{V}, \mathcal{E}, A) \tag{5}$$

Let $\mathcal{V} = \{1,2 \dots, M\}$ denote the set of nodes, representing the robots in the formation. $\mathcal{E} = \{(i,j), i,j \in \mathcal{V}, i \neq j\}$ be the set of directed edges, where an edge (i,j), indicates that robot i receives information from robot j.

The communication topology is characterized by an adjacency matrix $A = [a_{ij}] \in \mathbb{R}^{M\times M}$, where $a_{ij} = 1$, $(i,j) \in \mathcal{E}$ if i receives information from robot j, otherwise 0. In this communication topology, each robot exchanges only its predicted state with neighboring robots rather than full control inputs. State-only communication significantly reduces bandwidth and power consumption, which is particularly advantageous for mobile robots operating under limited communication resources. Similar observations were also reported in distributed MPC studies such as [14], where minimizing exchanged variables improves communication efficiency and energy usage

The communication structure is designed to satisfy the following criteria: ensure global connectivity of the network; minimize computational and communication load, which is essential for distributed control schemes; support the implementation of DMPC, where each robot solves a local optimization problem based on information from its neighbors. This sparse and directed topology reduces dependence on centralized or fully-connected communication, which may help maintain coordinated behavior even when only local neighbor information is available.

3. PROPOSED METHOD

3.1. DMPC cost function

In the DMPC framework, each robot in the system independently optimizes a local objective function at every time step, based on its current state and information received from neighboring robots. The optimization problem is solved in a receding horizon manner, ensuring both feasibility and stability throughout the control process.

Based on the exchanged state information between robots, we construct a cost function comprising the following components: formation-keeping relative to the leader or virtual leader; consensus on position and heading with neighboring robots; collision avoidance; control effort minimization, and terminal cost to ensure long-term stability [11], [15]-[17]. The general form of the cost function is defined as:

$$J_i = \sum_{h=0}^{H-1}\left[J_i^{form}[h] + J_i^{cons}[h] + J_i^{obs}[h] + J_i^{ctr}[h] + J_i^{ter}\right] \tag{6}$$

3.1.1. Formation-keeping cost

If the leader (or virtual leader) L is included in the neighbor set $\mathcal{N}_i$, robot i maintains a desired relative offset $r_{rel,i}$ expressed in the leader's body frame [14]:

$$J_i^{form}[h] = \begin{cases} \lambda_f . \left\| p_i[h] - (p_L[h] + R(\theta_L[h].r_{rel,i})) \right\|^2 \\ 0, \end{cases} \tag{7}$$

where $p_i = [x_i, y_i]^{\mathrm{T}}$ is the robot position, and $\lambda_f > 0$ is the formation. If the leader is not a neighbor (e.g., second -layer follower), we set $J_i^{form} = 0$.

3.1.2. Consensus cost with neighboring robots

Robots maintain cohesion by regulating position and heading differences relative to neighbors [18]:

$$J_i^{cons}[h] = \sum_{j\in\mathcal{N}_i}\left[\lambda_d\left(\left\|p_i[h] - p_j[h]\right\| - d_{ref,ij}\right)^2 + \lambda_\theta\left(\theta_i[h] - \theta_j[h]\right)^2\right], \tag{8}$$

where $d_{ref,ij}$ is the desired inter-robot spacing, $\lambda_d, \lambda_\theta > 0$.

3.1.3. Obstacle avoidance cost (barrier function)

Let $p_{obs,m}$ denote the position of the m-th obstacle cell in the grid map, and let $d_{im}[h] = \|p_i[h] - p_{obs,m}\|$ be the distance between robot i and obstacle m. To strongly penalize distances approaching a safety margin d_{safe}, we adopt a smooth barrier function [19]-[21]:

$$J_i^{obs}[h] = \sum_{m=1}^{N_{obs}} \lambda_{obs} \frac{1}{(d_{im}[h]-d_{safe})^2}, \quad (9)$$

where $\lambda_{obs} > 0$. This term grows rapidly as $d_{im}[h] \to d_{safe}$, ensuring collision-free trajectories.

3.1.4 Control effort minimization

Smooth input behavior is encouraged by penalizing the Euclidean norm of the control vector:

$$J_i^{ctr}[h] = \lambda_v . \|v_i[h]\|^2 \quad (10)$$

with $\lambda_v > 0$.

3.1.5. Terminal cost

A terminal-state penalty is imposed at the end of the prediction horizon to improve long-term stability:

$$J_i^{ter} = \left\|z_i(H) - z_{ref,i}(H)\right\|_{Q_f}^2, \quad (11)$$

where $Q_f \geq 0$ is a positive semidefinite terminal weight matrix, and $z_{ref,i}(H)$ is the terminal reference state.

3.2. DMPC-SQP framework

In the distributed framework, each robot *i* solves a local optimal control problem at every time step over a prediction horizon of length *H*. The variable of the decision is the control sequence:

$$V_i = \{v_i[0], v_i[1], \dots, v_i[H-1]\}$$

and the objective is to minimize the cost function $J_i(V_i)$ defined in (6)-(11), which aggregates formation-keeping, consensus, obstacle avoidance, control effort, and terminal penalties.

3.2.1. Discrete-time kinematics and input constraints

The continuous-time kinematic model of the omnidirectional mobile robot is given in (3). Over the prediction horizon, it is discretized with sampling period T_s as:

$$z_i[h+1] = z_i[h] + T_s \begin{bmatrix} \cos\theta_i[h] & -\sin\theta_i[h] & 0 \\ \sin\theta_i[h] & \cos\theta_i[h] & 0 \\ 0 & 0 & 1 \end{bmatrix} v_i[h], h = 0, \dots, H-1, \quad (12)$$

where $z_i[h] = [x_i[h], y_i[h], \theta_i[h]]^T$ and $v_i[h] = [v_{x,i}[h], v_{y,i}[h], w_i[h]]^T$. To ensure feasible and safe motion, the control inputs are bound by:

$$\left\|\left[v_{x,i}[h], v_{y,i}[h]\right]^T\right\| \leq v_{max}, |w_i[h]| \leq w_{max}, h = 0, \dots, H-1 \quad (13)$$

3.2.2. Nonlinear optimal control problem

Using the cost (6)-(11) and the prediction model (12), the local DMPC problem for robot iii can be written as:

$$\min_{V_i} J_i(V_i) \quad (14)$$

where the state trajectory $\{z_i[h]\}_{h=0}^H$ is generated by (12) starting from the current state $z_i[0]$, and the controls satisfy the bounds (13) for all *h*. Neighbor information enters $J_i(\cdot)$ through the formation and consensus terms as described in section 3.1.

3.2.3. Lagrangian and SQP formulation

To solve the nonlinear constrained problem (14) efficiently at each sampling instant, we employ SQP, a second-order method widely used in nonlinear MPC [5], [15], [16], [22]. The Lagrangian of the local problem is defined as:

$$\mathcal{L}_i(V_i, \lambda_i, \mu_i) = J_i(V_i) + \lambda_i^T c_{eq}(V_i) + \mu_i^T c_{ineq}(V_i) \quad (15)$$

where $c_{eq}(V_i)$ collects the discretized dynamics constraints (12) along the horizon, and $c_{ineq}(V_i)$ represents the input bounds (13). Vectors λ_i and μ_i are the corresponding Lagrange multipliers.

At iteration *k*, the Lagrangian is approximated around the current control sequence V_i^k using a second-order Taylor expansion:

$$\mathcal{L}_i\left(V_i^k + d_i\right) \approx \mathcal{L}_i\left(V_i^k\right) + \nabla\mathcal{L}_i\left(V_i^k\right)^T d_i + \frac{1}{2} d_i^T H_i^k d_i \quad (16)$$

where, d_i is the search direction and H_i^k is a positive-definite approximation of the Hessian of the Lagrangian, updated by a BFGS-type formula.

The equality and inequality constraints are linearized at V_i^k, leading to an approximating quadratic programming (QP) subproblem of the form.

$$\min_{d_i} \nabla\mathcal{L}_i\left(V_i^k\right)^T d_i + \frac{1}{2} d_i^T H_i^k d_i, s.t\ c_{eq}\left(V_i^k\right) + \nabla c_{eq}\left(V_i^k\right) d_i = 0, c_{ineq}\left(V_i^k\right) + \nabla c_{ineq}(V_i^k) d_i \leq 0 \quad (17)$$

In the implementation used in this work, the dynamics (12) are embedded in the prediction model that constructs $J_i(V_i)$, and the velocity bounds (13) are handled as simple box constraints. In that case, (17) reduces to a QP with bound constraints on $V_i^k + d_i$; its objective still has the standard form (16) with H_i^k interpreted as an approximate Hessian of the Lagrangian.

The gradient of the cost with respect to the control vector is computed numerically by a finite-difference approximation:

$$[\nabla J_i(V_i)]_j \approx \frac{J_i(V_i + \varepsilon e_j) - J_i(V_i)}{\varepsilon}, \varepsilon \ll 1, \quad (18)$$

where, e_j is the unit vector with 1 at index *j* and 0 elsewhere. After solving the QP, the control sequence is updated as:

$$V_i^{k+1} = V_i^k + \alpha^k d_i^k, \quad (19)$$

where, $\alpha^k \in (0,1]$ is the step size obtained from a line-search procedure to ensure sufficient decrease of a suitable merit function built from J_i, c_{eq}, and c_{ineq}. At each sampling time, only the first control input in V_i^{k+1} is applied to the robot, and the remaining part of the sequence is shifted and used to warm-start the next SQP iteration.

4. SIMULATION SETUP AND RESULTS

All robots in the formation (both leader and followers) share a common control structure based on a DMPC cost function optimized using SQP. The cost function is defined by (5)-(11). This function ensures key operational objectives: reference trajectory tracking, collision avoidance, formation shape maintenance, and stable convergence. These are fundamental goals and common challenges that any robot formation control system must address.

4.1. Simulation setup

4.1.1. Formation topology and reference trajectory

The trajectory of the leader robot is obtained as the result of a path-planning problem from the start point to the goal point on a 2D environmental map with static obstacles, generated by the $A*$ algorithm combined with covariant Hamiltonian optimization for motion planning (CHOMP) optimization [23]. This trajectory then serves as a virtual reference path. The leader solves an optimization problem to track the generated trajectory while simultaneously transmitting its predicted state information to its neighboring follower robots. Each neighboring follower also solves its own optimization problem to track its virtual

leader and subsequently shares its predicted state information with the next followers. The resulting system topology forms a tree-like communication structure as illustrated in Figure 1.

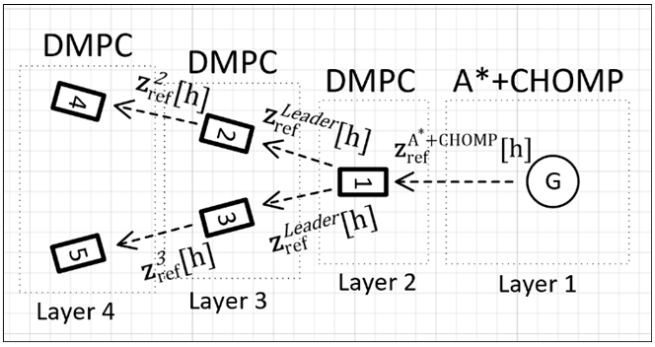

Figure 1. Tree system topology

The system adopts a hierarchical structure, where robot 1 serves as a virtual leader. No global broadcasting or control signal exchange exists among robots, ensuring a fully distributed architecture suitable for DMPC. The physical leader follows a predefined trajectory and shares its predicted state sequence $[0: N_p]$ with directly connected followers. First-layer followers receive this trajectory and solve local DMPC problems incorporating formation-keeping, obstacle avoidance via barrier functions, and a Lyapunov-based terminal cost. Second-layer followers, disconnected from the leader, optimize based on neighbor consensus, obstacle avoidance, and terminal stability.

After obtaining the motion trajectory, to verify the proposed model's capability of avoiding static obstacles, four additional obstacles of different sizes and shapes were introduced in the simulation. These obstacles were placed adjacent to or overlapping the previously generated trajectory (Figure 2). In this way, the experiment evaluates both the ability of the proposed model to avoid static obstacles encountered during motion and its capability to maintain the formation of the robot team.

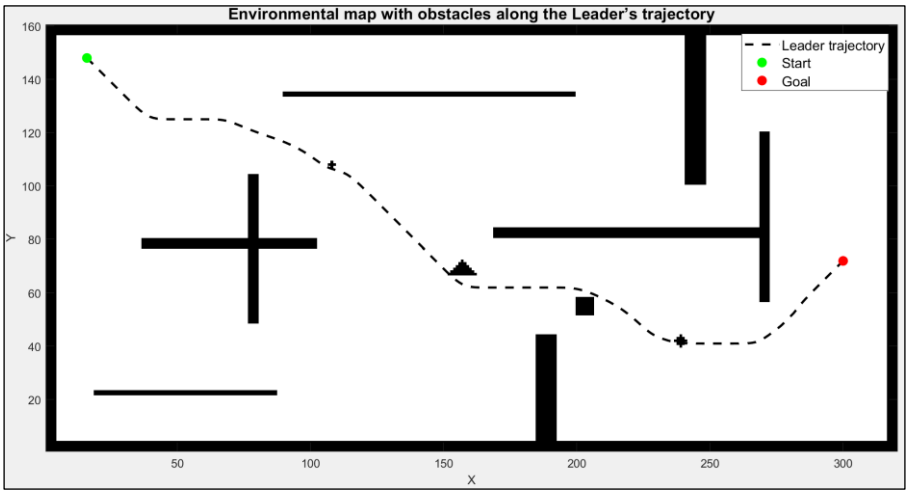

Figure 2. Environmental map containing the leader robot's trajectory and obstacles

4.1.2. DMPC and SQP parameters

Table 1 summarizes the main parameters of the proposed DMPC–SQP controller, consistent with the notation introduced in section 3, and additionally specifies the Gaussian noise injected into the simulation to evaluate robustness. In this study, zero-mean Gaussian noise is added to both the position and heading measurements of every robot at each control step [24], [25]. The noise levels are selected to reflect small-amplitude sensing uncertainty while preserving numerical stability of the SQP solver.

Table1. Parameters used in the DMPC-SQP controller

Parameter	Description	Value
T_s	Sampling time	0.1s
H	Prediction horizon	5 steps
N	Number of robots	5 (1 leader, 4 followers)
Leader index	Physical leader	Robot 3
$v_{x,i}, v_{y,i}$ bounds	Translational velocity limits	[-1,1] m/s
w_i bounds	Angular rate limit	[-π/4, π/4] rad/s
λ_d	Formation/position tracking weight	50
λ_θ	Heading tracking weight	30
λ_v	Control effort weight	0.1
Q_f	Terminal weight	diag(20,20,2)
r_{safe}	Safety margin	r_{obs}+1.5
r_{obs}	Obstacle radius	2.5 (map units)
r_{filter}	Obstacle search radius	6
K_{near}	Max. nearest obstacles considered	60
μ_{barr}	Obstacle barrier weight	8
SQP solver	MATLAB *fmincon*	Algorithm: SQP
Max SQP iter.	Solver iterations per step	5
Tolerances	Optimality, step tolerance	5×10^{-3}, 10^{-4}
sigma_pos	Position noise standard deviation	0.01m
sigma_theta	Heading noise standard deviation	0.1°

4.1.3. Obstacle representation and log-barrier function

Static obstacles are extracted from the binary grid map, and only those within a local radius $r_{filter} = 6$ around the current robot position are considered in the cost to reduce computation. For each obstacle $\mathcal{O}$, with distance $d_o[h] = \|p_i[h] - p_o\|$, a smooth logarithmic barrier is applied whenever the robot enters the safety region:

$$J_{obs}^{i}[h] = \begin{cases} \mu_{barr}(-log\rho_o[h]), & r_{obs} < d_o[h] < r_{safe}, \\ 10^6, & d_o[h] < r_{obs}, \\ 0, & otherwise, \end{cases}$$

$$\rho_o[h] = \frac{d_o[h]-r_{obs}}{r_{safe}-r_{obs}} \in (0,1)$$

The ratio $\rho_o[h]$ is clipped to $[10^{-9}, 1-10^{-9}]$ for numerical stability. This construction yields a differentiable penalty on (r_{obs}, r_{safe}), compatible with the SQP solver, while the large constant cost 10^6 enforces an effective hard constraint inside the obstacle radius.

4.1.4. SQP solver configuration and environment

For each robot i, the optimization variable is the stacked control vector

$$V_i = [v_i[0], v_i[1], \dots, v_i[H-1]]^T \in \mathbb{R}^{3H}$$

where the horizon states are generated by forward simulation of the discrete kinematic model in (12).

The SQP iterations are carried out by MATLAB's *fmincon* with algorithm: SQP (quasi-Newton with BFGS Hessian approximation); finite-difference gradients of the cost; box constraints enforcing the velocity bounds in (13); maximum 5 iterations per MPC step; warm start by shifting the previous optimal sequence one step ahead. All simulations were executed in MATLAB on a PC platform (Intel Core i7-8650U CPU@1.90GHz, 16 GB RAM), which is sufficient to run the controller faster than the sampling period $T_s = 0.1$s.

4.2. Simulation results

4.2.1. Formation tracking and obstacle avoidance

As shown in Figure 3, the proposed hierarchical virtual-leader DMPC framework enables the leader to follow the smoothed global trajectory computed by $A*$ + CHOMP, while all followers maintain the desired V-formation. The black cells represent static obstacles placed along or near the planned path. The formation successfully avoids collisions and preserves its shape even in curved segments and narrow passages. The zoomed-in views at points B1 and B2 confirm that the inter-robot distances and relative geometry remain close to their desired values throughout the motion.

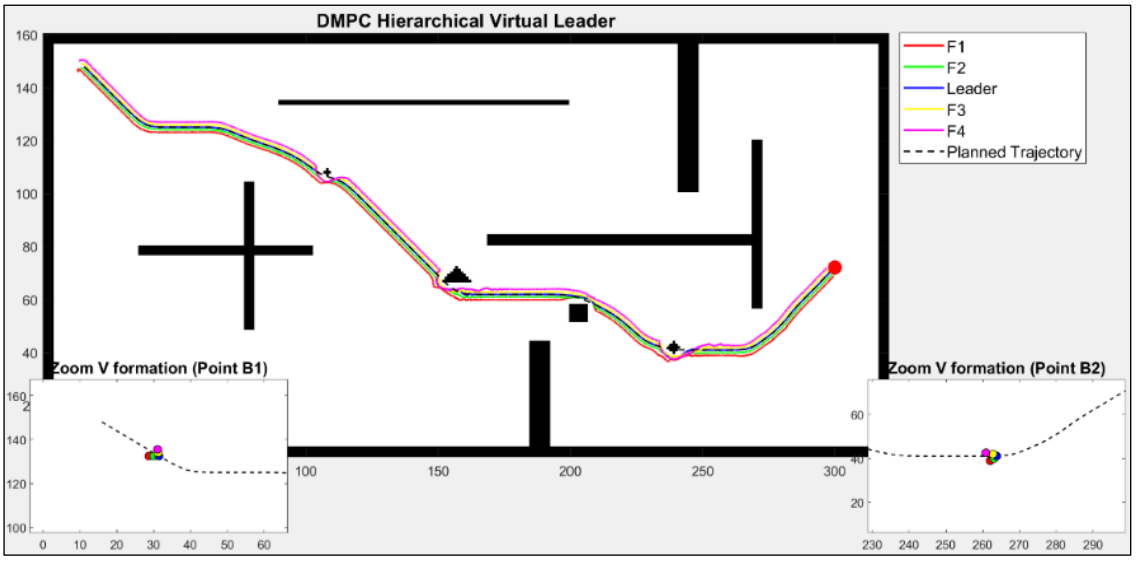

Figure 3. Simulation of the hierarchical virtual-leader DMPC formation tracking and obstacle avoidance

4.2.2. Tracking errors and steady-state performance

Figure 4 shows the position tracking errors of all robots, and Figure 5 presents the corresponding heading errors. Table 2 reports the minimum, maximum, and average values of these errors for each robot. Although the injected Gaussian noise has relatively small amplitude (0.01 m in position and 0.1° in heading), its presence allows evaluating the disturbance-rejection capability of the hierarchical DMPC-SQP formulation. The tracking-error curves in Figures 4 and 5 show that noise does not introduce drift or oscillation in either position or orientation. Instead, the errors remain bound and smooth, indicating that the predictive optimization compensates for stochastic perturbation effectively.

Overall, the results demonstrate that-even without explicit stochastic modeling or filtering-the proposed DMPC-SQP architecture inherently suppresses the effect of low-level Gaussian noise. This suggests promising robust characteristics and indicates that the model can be extended toward more challenging uncertainty conditions (e.g., higher noise, biased sensors, or partially missing measurements) in future work.

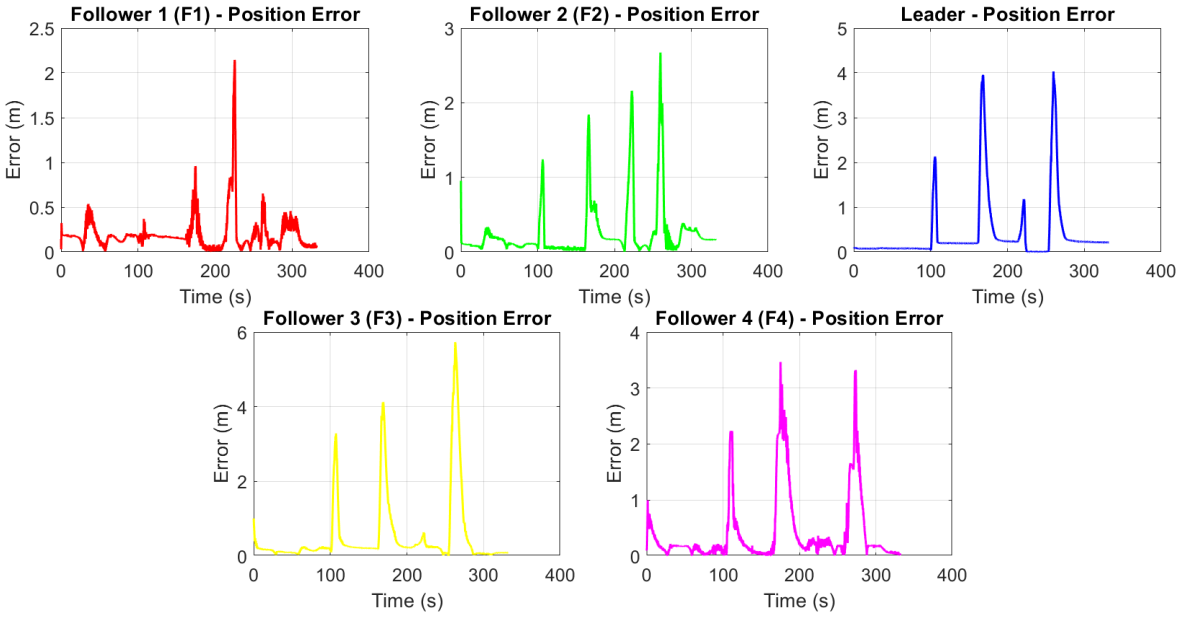

Figure 4. Position error of each robot

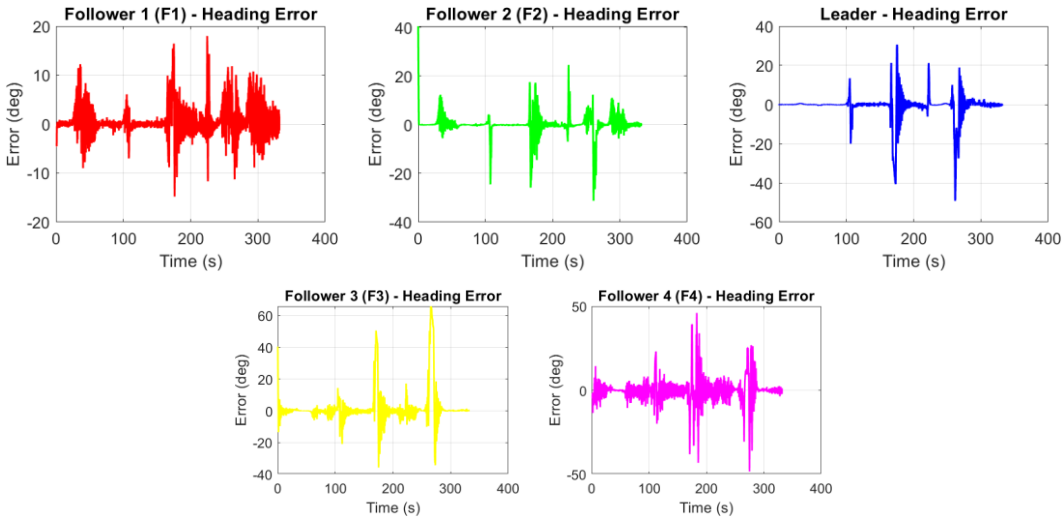

Figure 5. Heading error of each robot

Table 2. Table of the minimum, maximum, and average position and heading errors of the robots

Robot	Min pos error	Max pos error	Avg pos error	Min heading error	Max heading error	Avg heading error
Leader	0.0227	4.5327	0.4284	-32.0948	68.2998	1.1102
F1	0.0009	2.1725	0.2064	-14.0287	27.1414	0.5154
F2	0.0066	3.4554	0.2941	-30.0047	48.8564	0.3915
F3	0.0305	5.5310	0.5606	-33.8092	73.4396	1.2931
F4	0.0058	2.5944	0.3350	-37.3614	31.1505	-0.6190

4.2.3. Computational performance

The computational efficiency of the proposed hierarchical DMPC–SQP scheme is evaluated through the recorded optimization times (Table 3). The average SQP solve time per robot is 0.009835 s (≈101.7 Hz), showing that each local optimization problem is solved very quickly. The maximum solve time, 0.048408 s, occurs during demanding avoidance maneuvers but remains well below the sampling period $T_s = 0.1$ s, ensuring real-time feasibility.

At the system level, the peak mean solve time across all robots is 0.035190 s (step 1031), indicating a moment of increased formation-wide computational load; however, this value also satisfies real-time constraints. The mean total DMPC cycle time, including all Gauss-Seidel updates, is 0.051994 s, corresponding to a 19.23 Hz control update rate-significantly faster than the required 10 Hz.

These results demonstrate that the proposed hierarchical DMPC–SQP framework is computationally efficient and suitable for real-time implementation in obstacle-rich multi-robot environments. Additional simulation visuals are available at: https://sites.google.com/view/tuan-dp/trang-ch%E1%BB%A7.

Table 3. SQP computational statistics

Mean solve time per robot	Max solve time per robot	Std deviation	Peak mean time (at step 1031)	Mean total DMPC cycle time
0.009835s	0.048408s	0.004150s	0.035190s	0.051994s

5. CONCLUSION

This paper introduced a hierarchical virtual-leader DMPC framework solved via SQP for V-formation control of omnidirectional mobile robots. The leader follows a trajectory generated by $A*$ and CHOMP, while followers rely only on local neighbor information. Simulations show that the approach achieves accurate formation keeping, smooth trajectory tracking, and safe obstacle avoidance. The controller runs in real time with low computational cost, and the formation remains stable even when small Gaussian noise is added to position and heading measurements, demonstrating inherent robustness.

Current limitations include the focus on static obstacles, low-amplitude noise, and the absence of comparisons with alternative DMPC architectures. Future work will extend the method to dynamic environments, analyze performance under communication delays or stronger uncertainty, and validate the approach on real robot platforms.

ACKNOWLEDGMENTS

The authors would like to thank Thai Nguyen University of Technology (TNUT), Viet Nam.

FUNDING INFORMATION

Authors state there is no external funding involved.

AUTHOR CONTRIBUTIONS STATEMENT

This journal uses the Contributor Roles Taxonomy (CRediT) to recognize individual author contributions, reduce authorship disputes, and facilitate collaboration.

Name of Author	C	M	So	Va	Fo	I	R	D	O	E	Vi	Su	P	Fu
Tuan Phu Duong	✓		✓	✓	✓	✓	✓		✓	✓				
Vinh Quang Nguyen		✓		✓		✓	✓		✓			✓		
Minh Tuan Nguyen		✓			✓		✓	✓		✓	✓	✓	✓	

C : **C**onceptualization
M : **M**ethodology
So : **So**ftware
Va : **Va**lidation
Fo : **Fo**rmal analysis
I : **I**nvestigation
R : **R**esources
D : **D**ata Curation
O : Writing - **O**riginal Draft
E : Writing - Review & **E**diting
Vi : **Vi**sualization
Su : **Su**pervision
P : **P**roject administration
Fu : **Fu**nding acquisition

CONFLICT OF INTEREST STATEMENT

Authors state no conflict of interest.

DATA AVAILABILITY

The data that support the findings of this study are available from the corresponding author upon request.

REFERENCES

[1] M. I. El-Afifi *et al.*, "Coordinated distributed model predictive control for multi-carrier energy systems," *Scientific Reports*, vol. 14, 2024, doi: 10.1038/s41598-024-78314-5.
[2] S. Ahmad, Z. Feng, and G. Hu, "Multi-robot formation control using distributed null space behavioral approach," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, Hong Kong, China, 2014, pp. 3607-3612, doi: 10.1109/ICRA.2014.6907380.
[3] P. Chanfreut, J. M. Maestre, Q. Zhu, and W. P. M. H. M. Heemels, "Cooperative nonlinear distributed model predictive control with dissimilar control horizons," *2024 IEEE 63rd Conference on Decision and Control (CDC)*, Milan, Italy, 2024, pp. 6175-6180, doi: 10.1109/CDC56724.2024.10886159.
[4] G. Stomberg *et al.*, "Cooperative distributed model predictive control for embedded systems: Experiments with hovercraft formations," *Computer Science: Robotics*, 2024, doi: 10.48550/arXiv.2409.13334.
[5] A. Jordana, S. Kleff, A. Meduri, J. Carpentier, N. Mansard, and L. Righetti "Stagewise Implementations of Sequential Quadratic Programming for Model-Predictive Control," *HAL Open Science*, 2023.
[6] C. Stamouli, A. Tsiamis, M. Morari, and G. J. Pappas, "Adaptive Stochastic MPC under Unknown Noise Distribution," *Electrical Engineering and Systems Science: Systems and Control*, 2022, doi: 10.48550/arXiv.2204.01107.
[7] S. P. Talebi and S. Werner, "Distributed Kalman Filtering and Control Through Embedded Average Consensus Information Fusion," in *IEEE Transactions on Automatic Control*, vol. 64, no. 10, pp. 4396-4403, Oct. 2019, doi: 10.1109/TAC.2019.2897887.
[8] M. Ramírez-Neria, R. Madonski, E. G. Hernández-Martínez, N. Lozada-Castillo, G. Fernández-Anaya, and A. Luviano-Juárez, "Robust trajectory tracking for omnidirectional robots by means of anti-peaking linear active disturbance rejection," *Robotics and Autonomous Systems*, vol. 183, p. 104842, Jan. 2025, doi: 10.1016/j.robot.2024.104842.
[9] H. Zhang, S. Wang, Y. Xie, H. Wu, T. Xiong, and H. Li, "Nonlinear Model Predictive Control of an Omnidirectional Mobile Robot with Self-tuned Prediction Horizon," *2022 IEEE 17th Conference on Industrial Electronics and Applications (ICIEA)*, Chengdu, China, 2022, pp. 584-589, doi: 10.1109/ICIEA54703.2022.10006295.
[10] D. S. Lal and A. Vivek, "Dynamic modeling and control of omni-directional mobile robots," 2*017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*, pp. 1–7, Apr. 2017. doi: 10.1109/iccpct.2017.8074219.
[11] J. M. Maestre and R. R. Negenborn, "Distributed Model Predictive Control Made Easy," *Intelligent Systems, Control and Automation: Science and Engineering*, 2014, vol. 69, doi: 10.1007/978-94-007-7006-5.
[12] H. Do, H. Nguyen, C. Nguyen, M. Nguyen, and M. Nguyen, "Formation control of multiple unmanned vehicles based on graph theory: A Comprehensive Review," *EAI Endorsed Transactions on Mobile Communications and Applications*, vol. 7, no. 3, p. e3, Dec. 2022, doi: 10.4108/eetmca.v7i3.2416.
[13] W. Liu, J. Hu, H. Zhang, M. Y. Wang, and Z. Xiong, "A Novel Graph-Based Motion Planner of Multi-Mobile Robot Systems with Formation and Obstacle Constraints," in *IEEE Transactions on Robotics*, vol. 40, pp. 714-728, 2024, doi: 10.1109/TRO.2023.3339989.

[14] T. Zhang and X. Zhang, "Distributed Model Predictive Control with Particle Swarm Optimizer for Collision-Free Trajectory Tracking of MWMR Formation," *Actuators*, vol. 12, no. 3, p. 127, Mar. 2023, doi: 10.3390/act12030127.
[15] P. T. Duong, T. M. Nguyễn, and Q. V. Nguyễn, "Application of the SQP algorithm in leader–follower formation control," *Journal of Military Science and Technology*, vol. 105, pp. 11–19, Aug. 2025, doi: 10.54939/1859-1043.j.mst.105.2025.11-19.
[16] A. Carron, D. Saccani, L. Fagiano, and M. N. Zeilinger, "Multi-agent Distributed Model Predictive Control with Connectivity Constraint," *IFAC-PapersOnLine*, vol. 56, no. 2, pp. 3806–3811, 2023, doi: 10.1016/j.ifacol.2023.10.1310.
[17] X. Zhou, Y. Zou, S. Li, X. Li, and H. Fang, "Distributed model predictive control for multi-robot systems with conflicting signal temporal logic tasks," *IET Control Theory & Applications*, vol. 16, no. 5, pp. 554–572, Feb. 2022, doi: 10.1049/cth2.12254.
[18] L. Ji, X. Qu, C. Tang, S. Yang, X. Guo, and H. Li, "Consensus Formation of Multi-agent Systems with Obstacle Avoidance based on Event-triggered Impulsive Control," *Journal of Intelligent & Robotic Systems*, vol. 109, no. 3, Nov. 2023, doi: 10.1007/s10846-023-01987-z.
[19] M. Cavorsi, L. Sabattini and S. Gil, "Multirobot Adversarial Resilience Using Control Barrier Functions," in *IEEE Transactions on Robotics*, vol. 40, pp. 797-815, 2024, doi: 10.1109/TRO.2023.3341570.
[20] A. Thirugnanam, J. Zeng, and K. Sreenath, "Control Barrier Functions for Collision Avoidance Between Strongly Convex Regions," *Computer Science: Robotics*, 2023, arXiv, doi: 10.48550/ARXIV.2306.13259.
[21] J. Zeng, Z. Li, and K. Sreenath, "Enhancing Feasibility and Safety of Nonlinear Model Predictive Control with Discrete-Time Control Barrier Functions," *Electrical Engineering and Systems Science: Systems and Control*, 2021, arXiv. doi: 10.48550/ARXIV.2105.10596.
[22] R. Grandia, A. J. Taylor, A. Singletary, M. Hutter, and A. D. Ames, "Nonlinear Model Predictive Control of Robotic Systems with Control Lyapunov Functions," *in Proceedings of Robotics: Science and Systems (RSS)*, Corvallis, Oregon, USA, Jul. 12–16, 2020.
[23] D. P. Tuan, N. T. Minh, N. H. Viet, and N. Q. Vinh. "A trajectory planning method for a group of leader-follower mobile robots in a noisy environment," *Hanoi University of Industry Journal of Science and Technology*, vol 61, No 9, pp 80-86, 2025, doi: 10.57001/huih5804.2025.324.
[24] N. Niknejad, G. S. Sankar, B. Kiumarsi, and H. Modares, "Robust Model Predictive Control Design for Autonomous Vehicles with Perception-based Observers," *Computer Science: Robotics*, 2025, doi: 10.48550/ARXIV.2509.05201.
[25] H. Li, B. Jin, and W. Yan, "Distributed model predictive control for linear systems under communication noise: Algorithm, theory and implementation," *Automatica*, vol. 125, p. 109422, Mar. 2021, doi: 10.1016/j.automatica.2020.109422.

BIOGRAPHIES OF AUTHORS

Tuan Phu Duong received the B.E. (2004), and M.E. (2015) degrees in Electrical Engineering from Hanoi University of Science and Technology, Vietnam. His research interests include measurement, automatic control, circuit design, reverse engineering, system design, embedded programming, and swarm robotics. He can be contacted at email: tuanphu8680@gmail.com.

Vinh Quang Nguyen works at Academy of Military Sciences and Technologies as the chief of the Department of Electronic Wireless Devices. He got Ph.D. degree in Russia, at Moscow State Technical University named after Bauman, in 2006. In July 2019, He became an Associate Professor. He has taken part in many scientific projects, for example: researching and applying of the control software, designing and installing of parallel systems for processing information in automatic systems accompanying flying objects, constructing of the project and technical theoretical factual foundations for fire-power antiaircraft complexes with small height to intercept Tomahawk rocket. Besides he has 75 publications in prestigious international journals in the fields of nonlinear control, modelling, the inertial navigation system, adaptive control, control systems in aeronautics. He can be contacted at email: vinhquang2808@gmail.com.

Minh Tuan Nguyen is currently the director of human resource training and development center at Thai Nguyen University, Vietnam, and also the director of advanced wireless communication networks (AWCN) lab. He has interest and expertise in a variety of research topics in the communications, networking, and signal processing areas, especially compressive sensing, and wireless/mobile sensor networks. He serves as technical reviewers for several prestigious journals and international conferences. He also serves as an editor for wireless communication and mobile computing journal and an editor in chief for ICSES transactions on computer networks and communications. He can be contacted at email: nguyentuanminh@tnut.edu.vn.