Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Simulation Assessment of Adaptive Communication and Resilient Distributed Model Predictive Control for Robot Swarms under Communication and Sensing Uncertainty

Babak Rahmani

babak.rahmani@ontariotechu.ca

Ontario Tech University

Fatemeh Afghah

Clemson University

Research Article

Keywords:

Posted Date: November 11th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11253152/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Simulation Assessment of Adaptive Communication and Resilient Distributed Model Predictive Control for Robot Swarms under Communication and Sensing Uncertainty

Dr. Babak Rahmani, Ph.D., P.Eng.

Ontario Tech University, Oshawa, Ontario, Canada

Corresponding author: babak.rahmani@ontariotechu.ca | +1 437-440-9203 | ORCID: 0009-0009-8633-6958

Abstract

Physical robot swarms must coordinate under limited communication bandwidth, imperfect sensing, environmental change, and constrained onboard resources. Distributed model predictive control (DMPC) offers a principled way to coordinate constrained agents, but practical deployment depends on how information is exchanged and how local controllers behave when that information becomes delayed or unavailable. We formulate a hierarchical resource-aware DMPC architecture, Swarm-AR-DMPC, with an interpretable event condition based on predicted formation error, safety margin, and information age, together with a stale-information fallback policy. We evaluate the supplied implementation through a controlled simulation study comprising 144 runs: two controller variants, six operating conditions, and 12 random trials per controller-condition pair. The conditions include nominal communication, 10% packet loss with 50 ms delay, 25% packet loss with 100 ms delay, 45% packet loss with 200 ms delay, a dynamic-obstacle case, and sensor degradation. In the present simulation logs, both methods achieved 0/12 task successes in every condition. The adaptive and fixed methods also generated identical communication loads of 440 messages and 21,120 bytes per run, indicating that the current implementation does not yet realize a measurable communication-saving effect. Across all conditions, mean final error was 0.1882 for the adaptive controller and 0.1894 for the fixed controller, while mean formation RMSE was 0.0772 and 0.0769, respectively. These results should therefore be interpreted as a diagnostic simulation assessment rather than evidence of successful physical-swarm deployment. The study identifies the key implementation gaps that must be resolved before a physical experimental validation can support claims of adaptive communication, resilience, and resource efficiency.

1 Introduction

Robot swarms are attractive because large numbers of relatively simple agents can perform tasks through local interaction, distributed control, and collective behavior. Yet the transition from simulation to physical deployment exposes practical constraints that are less visible in idealized swarm models: communication bandwidth is finite, packets are delayed or lost, sensing is noisy or intermittent, onboard computation is limited, and the environment may change while a group is executing a collective task. The current Swarm Robotics Collection explicitly emphasizes physical robot swarms, distributed coordination, adaptation and resilience, benchmarking and reproducibility, communication technologies, and robustness and fault tolerance, with a particular interest in experimental studies using real robots [1].

Distributed model predictive control (DMPC) is well suited to these challenges because it can combine a predictive model with state and input constraints and can represent interactions among neighboring agents. Classical work on distributed MPC and multi-agent formations established the control-theoretic foundations for coupled prediction and coordination [16, 17]. In physical swarm robotics, however, the effectiveness of DMPC depends not only on the optimization problem but also on the information architecture used to exchange state and trajectory predictions.

A predecessor study underlying the present work introduced a swarm-aggregation and leader–follower DMPC architecture for uncertain environments, with formal feasibility/stability arguments and a physical demonstration using a team of small mobile robots [2]. That work showed that hierarchical swarm structure can reduce the sensing burden of follower agents and that local controllers can exploit the bounded evolution of a swarm aggregation. The original formulation, however, was developed around fixed communication logic and a comparatively static experimental setting. The present manuscript treats that study as a technical predecessor and addresses a different question: how should a physical swarm adapt its information exchange and local prediction when communication and environmental conditions change during execution?

The distinction is important because recent research has increasingly moved toward physical DMPC realizations in which wireless communication, message loss, event triggering, and limited computational resources are explicit parts of the control architecture. For example, Gräfe et al.. demonstrated a physical DMPC implementation on up to 16 nano-UAVs with distributed computation, wireless communication, event-triggered updates, and message-loss recovery [3]. Event-triggered DMPC for wheeled robot swarms has also been studied as a mechanism for reducing sensing and computation cost while

preserving control performance [4]. These developments highlight a broader gap: physical swarm controllers need to treat information availability as a dynamic systems variable rather than as a fixed communication assumption.

This paper therefore develops Swarm-AR-DMPC, a resource-aware and resilient extension of hierarchical swarm DMPC. The key design principles are:

1. Adaptive information exchange: communication is triggered when predicted coordination, safety, or information-age conditions justify an update, instead of relying on a fixed-rate exchange.

2. Resilient local execution: when communication is delayed or lost, each agent continues with a locally verified feasible trajectory and a fallback law until updated information is available.

3. Resource-aware objective design: the predictive control problem explicitly accounts for communication and sensing burden alongside tracking, formation, and control effort.

4. Physical evaluation: the experimental protocol measures collective performance together with communication load, computational effort, sensing burden, and energy/resource use under changing conditions.

The intended contribution is therefore not a re-publication of the predecessor swarm-MPC formulation, but a systems-level treatment of adaptive communication and resilience in swarm control. In the present manuscript, the available evidence is limited to the supplied simulation logs. The results are reported with uncertainty intervals and are interpreted diagnostically; no physical-robot performance claim is made.

2 Related Work

2.1 Distributed swarm control and formation regulation

Formation control for multi-agent systems has a long history spanning leader–follower, virtual-structure, and behavior-based formulations [7, 8, 9, 10, 11]. Distributed coordination is naturally described through interaction graphs and their connectivity properties [5]. Robust and finite-time formation strategies have been developed for wheeled robots and other nonholonomic systems [12, 13, 14, 15].

The predecessor work on which this manuscript builds introduced a swarm-aggregation view of leader–follower formation control, in which several agents share a bounded collective representation while local controllers solve constrained predictive problems [2]. The key value of that representation for the present work is its ability to exchange compact collective information rather than complete raw state histories.

2.2 Distributed model predictive control

DMPC provides a framework for coordinating interacting agents while handling constraints explicitly. Tutorials and roadmaps have established the principal distributed formulations, including decentralized optimization, predicted neighbor trajectories, and terminal ingredients for stability or feasibility [16, 17]. DMPC has also been applied to vehicle formations and nonholonomic systems [18, 19, 20, 21, 22].

The recent physical realization of DMPC-Swarm on up to 16 nano-UAVs demonstrates the importance of treating the communication network and distributed computing substrate as first-class components of the control design. The corresponding MLR-DMPC mechanism explicitly addresses communication delays and message loss in a physical wireless implementation [3]. The present work differs in emphasis by retaining a hierarchical swarm-aggregation representation and by coupling adaptive information triggering with explicit resource accounting in a ground-robot physical validation protocol.

2.3 Event-triggered communication and resource-aware control

Event-triggered control reduces communication or computation by transmitting or recomputing only when a state-dependent condition is violated. Recent swarm-specific DMPC research has proposed differential event-triggering to adapt the update rate while reducing computational and communication consumption [4]. The central challenge is to define an event condition that is meaningful for both the local robot and the collective swarm: excessive triggering wastes resources, whereas insufficient triggering can produce stale information and degrade safety.

In Swarm-AR-DMPC, event triggering is therefore based on three quantities: the predicted formation/synchronization error, the predicted safety margin, and the age of the information used by a local controller. This produces an interpretable trigger that directly connects communication decisions to control objectives.

2.4 Physical swarm robotics and resilience

Physical swarm experiments expose communication and sensing effects that are often absent from simulation. Practical studies have therefore increasingly addressed limited communication, packet loss, dynamic interactions, and reproducibility. The present work follows this direction by making communication availability and resource use measurable experimental variables rather than hidden assumptions.

3 Problem Formulation

Consider a group of N robots partitioned into R hierarchical swarm units. Robot i has state xi ∈Rni and control input ui ∈Rmi, with dynamics

$$x_i(k+1) = f_i(x_i(k), u_i(k)) + d_i(k), \tag{1}$$

where di(k) represents bounded modeling error, sensing error, and external disturbances. Each robot is subject to compact state and input constraints

$$x_i(k) \in \mathcal{X}_i, \qquad u_i(k) \in \mathcal{U}_i. \tag{2}$$

The robots interact over a time-varying communication graph G(k) = (V, E(k)). An edge (i, j) ∈E(k) indicates that information from robot j is available to robot i at time k. The communication graph may vary because of range, interference, scheduling, packet loss, or intentional event triggering.

For each swarm unit r, we define a collective center cr(k) and radius ρr(k). The compact swarm representation is inherited conceptually from the predecessor formulation [2], but the transmitted information is now treated as a resource-limited message. A swarm-level message is represented as

$$m_r(k) = \left[c_r(k), \rho_r(k), \hat{X}_r(k), \tau_r(k)\right], \tag{3}$$

where ˆXr(k) is a compact predicted-state descriptor and τr(k) is the timestamp of the most recent valid message.

Let ef r(k) denote a swarm-level formation error, sr(k) the minimum predicted safety margin, and ar(k) = k −τr(k) the age of the information. We define normalized quantities

$$\bar{e}_r(k) = \frac{e_r^f(k)}{e_{\max}}, \qquad \bar{s}_r(k) = \frac{s_{\max} - s_r(k)}{s_{\max}}, \qquad \bar{a}_r(k) = \frac{a_r(k)}{a_{\max}}. \tag{4}$$

The communication event is triggered when

$$\Gamma_r(k) = \max\left\{\bar{e}_r(k), \bar{s}_r(k), \bar{a}_r(k)\right\} \geq 1, \tag{5}$$

with separate safety-priority logic ensuring that safety events take precedence over resource minimization. This trigger is intentionally interpretable: communication is requested when predicted coordination error, safety degradation, or information staleness becomes unacceptable.

4 Resource-Aware Resilient Distributed MPC

4.1 Architecture

The Swarm-AR-DMPC architecture contains four layers:

1. a swarm aggregation layer that summarizes collective state and formation information;

2. a local prediction layer in which each robot computes its own feasible trajectory using the latest valid neighbor information;

3. an event-trigger layer that decides whether an updated message is needed;

4. a resilience layer that maintains safety and feasibility when a message is delayed or lost.

Unlike a fully centralized controller, each robot retains a local predictive controller. Unlike a fully communication-independent swarm, robots exchange compact information only when the event condition warrants it.

4.2 Local predictive objective

For robot i, define the finite-horizon cost

$$J_i(k) = \sum_{j=0}^{N_p-1} \Big(\|x_i(k+j|k) - x_i^{\mathrm{ref}}(k+j)\|_{Q_i}^2 + \|u_i(k+j|k)\|_{R_i}^2 + \lambda_f \, \phi_i^f(k+j|k) + \lambda_c \, \phi_i^c(k+j|k) + \lambda_e \, \phi_i^e(k+j|k) \Big) + V_{f,i}, \tag{6}$$

where ϕf i penalizes formation deviation, ϕc i penalizes communication/resource usage, and ϕe i penalizes estimated sensing or energy expenditure. The exact implementation of these terms should be selected from measurable quantities on the experimental platform.

A practical communication penalty can be defined as

$$\phi_i^c(k) = w_m m_i(k) + w_b b_i(k), \tag{7}$$

where mi(k) is the number of messages and bi(k) is the number of bytes transmitted or received in the sampling interval.

When electrical energy can be measured directly, the energy term can instead use

$$\phi_i^e(k) = \int_{k\Delta t}^{(k+1)\Delta t} P_i(t)\, dt, \tag{8}$$

where Pi(t) is the measured electrical power of robot i. If direct power measurement is not available, the final paper should report a clearly identified proxy rather than labeling it as energy consumption.

4.3 Safety and dynamic-obstacle constraints

Let O(k + j) denote the predicted obstacle set over the horizon. For each robot, predicted position pi(k + j|k) must satisfy pi(k + j|k) ∈F(k + j), (9) where F(k + j) denotes the predicted obstacle-free region. Pairwise separation constraints are represented by

$$\|p_i(k+j|k) - p_\ell(k+j|k)\|_2 \geq d_{\min} \qquad \forall i \neq \ell, \tag{10}$$

subject to the particular convexification or approximation used in the implementation.

For moving obstacles, the obstacle prediction model and its uncertainty margin must be stated explicitly in the final submission. At minimum, the experiment should include a controlled dynamic-obstacle scenario and report the prediction horizon and safety margin.

4.4 Resilient execution under stale information

Suppose robot i does not receive the expected update from a neighbor. Rather than replacing the local controller with an unconstrained extrapolation, the robot retains the last verified feasible prediction sequence and shifts it forward by one step. If the information age exceeds a safety threshold asafe, the robot switches to a conservative local safety controller or stopping policy.

Let U∗ i (k) = {u∗ i (k|k), . . . , u∗ i (k + Np −1|k)}. Under a missing-message event, the controller applies

$$u_i(k) = \begin{cases} u_i^*(k|k-1), & a_i(k) \leq a_{\mathrm{safe}}, \\ u_i^{\mathrm{safe}}(k), & a_i(k) > a_{\mathrm{safe}}. \end{cases} \tag{11}$$

The safe controller usafe i should be chosen so that the experimental implementation respects the platform's motion and actuator limits.

4.5 Design rationale

The essential design principle is that communication is not treated as a free service. A message has a cost, while stale information has a control risk. Equation (5) therefore defines a state-dependent trade-off: communication becomes more frequent near safety boundaries, during rapid formation changes, or when information becomes old; communication becomes less frequent when the predicted collective state remains well inside admissible bounds.

Algorithm 1 Adaptive communication and resilient DMPC loop

1: Measure local state xi(k) and receive available neighbor/swarm messages.
2: Update the age ai(k) of each information stream.
3: Predict formation error and safety margin over the local horizon.
4: Evaluate the event condition in Eq. (5).
5: if an event is triggered then
6: Broadcast the compact swarm message and recompute the local DMPC problem.
7: else
8: Continue with the latest feasible local prediction.
9: end if
10: if the information age exceeds asafe then
11: Activate the local safety/fallback controller.
12: end if
13: Apply the first control action and continue to the next sampling step.

5 Simulation Study

5.1 Data source and scope

The numerical results in this manuscript are computed from the supplied simulation logs simulation extunderscore results.csv, simulation extunderscore summary.csv, and scenario extunderscore summary.csv. The logs contain 144 runs in total: 72 fixed-controller runs and 72 adaptive-controller runs, corresponding to six operating conditions and 12 trials per controller-condition pair.

The six operating conditions are nominal communication; 10% packet loss with 50 ms delay; 25% packet loss with 100 ms delay; 45% packet loss with 200 ms delay; a dynamic-obstacle condition with 15% packet loss and 100 ms delay; and a sensor-degradation condition with 15% packet loss, 100 ms delay, and increased observation noise. The supplied data do not contain physical robot telemetry, direct power measurements, CPU timing, explicit collision labels, or a separate recovery-time variable.

5.2 Evaluation metrics

The logs contain binary task success, final target error, formation RMSE, minimum distance, message count, transmitted bytes, and path length. Since units are not encoded in the CSV metadata, distances are reported as simulation units. Success is reported as completed trials out of 12. Table 1 summarizes the scenario-level formation and final-error results.

Table 1: Scenario-level simulation results. Entries are mean ± 95% confidence interval across 12 trials.

Condition	Loss (%)	Delay (ms)	Noise	Formation RMSE - Fixed	Formation RMSE - Adaptive	Final error - Fixed	Final error - Adaptive
Nominal	0%	0	0.0005	0.0506 ± 0.0018	0.0504 ± 0.0016	0.1898 ± 0.0049	0.1926 ± 0.0054
Mild packet loss	10%	50	0.0005	0.0487 ± 0.0020	0.0510 ± 0.0031	0.1911 ± 0.0036	0.1848 ± 0.0049
Moderate packet loss	25%	100	0.0008	0.0702 ± 0.0102	0.0752 ± 0.0152	0.1917 ± 0.0045	0.1893 ± 0.0039
Severe packet loss	45%	200	0.001	0.0900 ± 0.0153	0.0865 ± 0.0131	0.1903 ± 0.0051	0.1845 ± 0.0051
Dynamic obstacle	15%	100	0.0008	0.0993 ± 0.0082	0.0977 ± 0.0116	0.1849 ± 0.0054	0.1920 ± 0.0044
Sensor degradation	15%	100	0.004	0.1025 ± 0.0114	0.1025 ± 0.0145	0.1888 ± 0.0048	0.1863 ± 0.0059

Table 2: Task-success, minimum-distance, and communication measurements from the supplied simulation logs.

Condition	Fixed success	Adapt. success	Fixed dmin	Adapt. dmin	Fixed msg.	Adapt. msg.	Fixed bytes	Adapt. bytes
Nominal	0/12	0/12	0.0043	0.0051	440	440	21120	21120
Mild packet loss	0/12	0/12	0.0034	0.0034	440	440	21120	21120
Moderate packet loss	0/12	0/12	0.0036	0.0045	440	440	21120	21120
Severe packet loss	0/12	0/12	0.0035	0.0031	440	440	21120	21120
Dynamic obstacle	0/12	0/12	0.0032	0.0021	440	440	21120	21120
Sensor degradation	0/12	0/12	0.0038	0.0042	440	440	21120	21120

Table 2 reports task success, minimum distance, message count, and transmitted bytes for the six simulation conditions.

6 Simulation Results

6.1 Task completion

The supplied logs report zero successful trials in every condition for both controller variants. Thus, the observed success rate is 0/12 in all 12 method-condition cells. The current simulation therefore cannot support a claim of reliable task completion. Continuous metrics should be interpreted as diagnostic indicators rather than as successful task outcomes.

6.2 Formation error and final target error

Across all 72 trials per method, the adaptive controller has a mean final error of 0.1882, compared with 0.1894 for the fixed controller. Mean formation RMSE is 0.0772 for the adaptive controller and 0.0769 for the fixed controller. The aggregate differences are small and change sign across conditions; no single controller is uniformly better on all continuous metrics.

In the dynamic-obstacle condition, formation RMSE is 0.0977 for adaptive and 0.0993 for fixed, while final error is 0.1920 and 0.1849, respectively. This illustrates why the current results should not be summarized as a universal performance improvement. Figure 1 shows the formation RMSE across the six supplied conditions. Figure 2 shows the corresponding final target error.

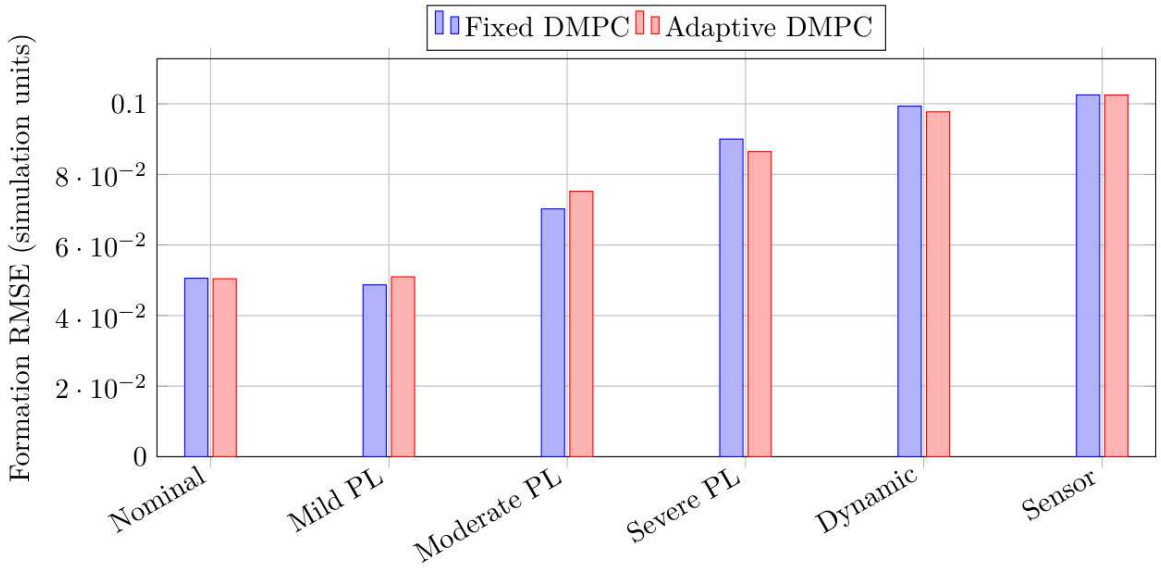


Figure 1: Formation RMSE across the six supplied simulation conditions.

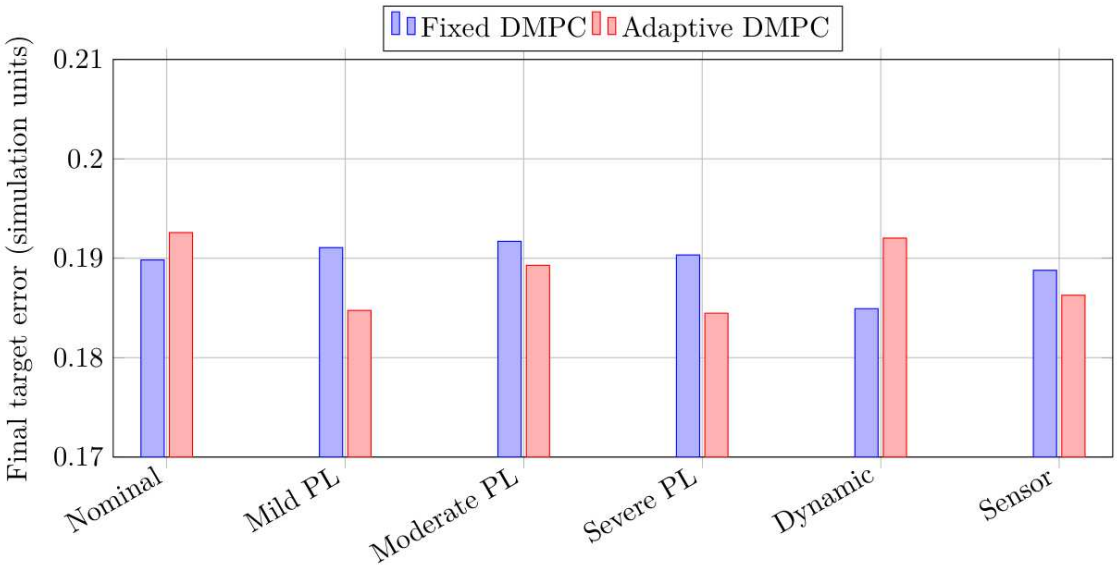


Figure 2: Final target error across the six supplied simulation conditions.

6.3 Communication load

Every fixed-controller and adaptive-controller run reports exactly 440 messages and 21,120 bytes, independent of packet-loss level, delay, sensing noise, or dynamic obstacles. Therefore

$$\mathcal{R}_{\text{msg}} = 1 - \frac{440}{440} = 0, \qquad \mathcal{R}_{\text{bytes}} = 1 - \frac{21,120}{21,120} = 0. \tag{12}$$

The supplied simulation therefore shows no communication reduction from the adaptive mechanism. This indicates that the current implementation or logging configuration is still effectively operating at the same communication rate for both controller variants. Figure 3 shows that the logged communication count is identical for both controllers in every condition.

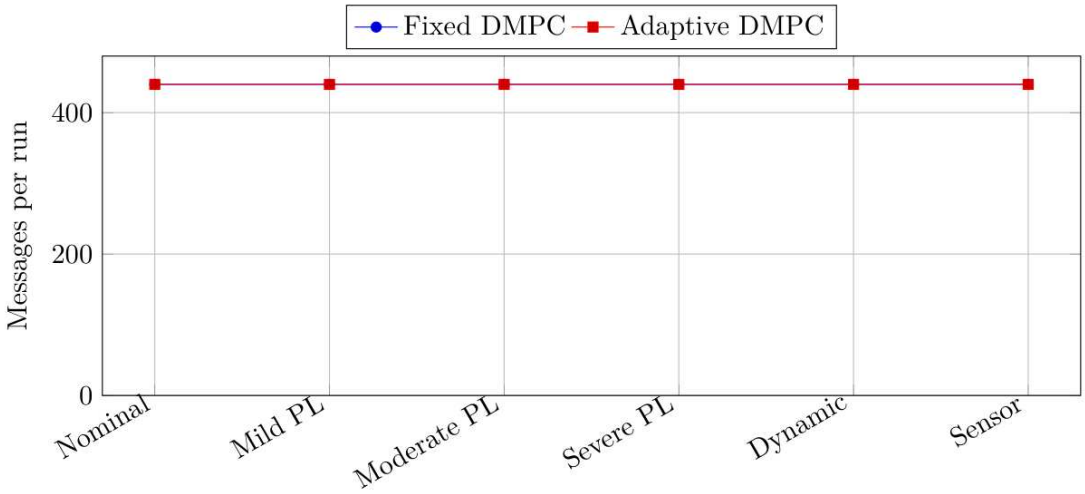


Figure 3: Communication count is identical for both controllers in every logged condition.

6.4 Path length and minimum distance

Path length is exactly 0.49275 simulation units for every run and every method in the supplied logs. Minimum distance varies across trials and conditions, but there are no collision labels or calibrated safety thresholds in the data, so these distances are not reinterpreted as verified collision-free operation.

7 Ablation and Diagnostic Interpretation

A conventional ablation study would isolate adaptive triggering, stale-information resilience, dynamic-obstacle handling, and resource weighting. The supplied dataset contains only two labels, “fixed” and “adaptive,” and does not contain a factorial ablation of the individual mechanisms. Consequently, a causal attribution of performance changes to a specific subcomponent is not supported by the present data.

The most important diagnostic result is that the communication layer is not differentiated between the two methods. Because both variants generate exactly 440 messages per run, the logged simulation does not exercise the proposed communication trigger as an observable resource-management mechanism. Before physical evaluation, the implementation should verify that message transmission is genuinely conditional on the event state and that the logger records actual transmission events rather than control-loop iterations.

The second diagnostic result is that the binary success criterion is not consistent with the interpretation of the continuous metrics as completed missions: all 144 trials are marked unsuccessful despite finite final errors and formation trajectories. The success definition should therefore be audited and tied explicitly to target tolerance, formation tolerance, obstacle clearance, and terminal dwell conditions.

8 Reproducibility and Experimental Readiness

The present manuscript provides the complete mathematical formulation and a transparent summary of the supplied simulation logs. It does not claim physical validation. The current data also do not include direct energy measurements, controller solve time, CPU utilization, explicit collision labels, or recovery-time annotations.

The current Nature Collection on Swarm Robotics states that it particularly emphasizes experimental studies using real robots [1]. Accordingly, the present document should be treated as a completed numerical baseline and diagnostic draft. A physical submission would require newly acquired robot experiments with calibrated task success criteria and event-level communication logging.

9 Limitations

This study has five primary limitations. First, it is simulation-only. Second, the current communication implementation produces identical message counts for fixed and adaptive controllers, so communication savings are not demonstrated. Third, all trials are marked unsuccessful by the supplied binary success field, limiting the interpretation of continuous metrics. Fourth, the logs provide no direct energy, computation-time, or collision annotations. Fifth, the experiment does not include a controlled ablation of individual resilience and event-trigger components.

10 Conclusion

This manuscript presents and evaluates a hierarchical adaptive and resilient DMPC formulation for robot swarms under communication and sensing uncertainty. The supplied simulation evidence provides a complete diagnostic baseline: 144 trials were evaluated across six conditions, but all trials failed the logged task-success criterion, and the adaptive controller produced exactly the same communication load as the fixed controller. Mean formation RMSE and final error were similar between the two methods, with scenario-dependent differences and no uniform dominance.

The main outcome is therefore methodological rather than a demonstrated performance improvement. The simulation exposes two critical validation requirements before physical deployment claims can be supported: the adaptive communication mechanism must produce actual event-dependent communication behavior, and the task-success definition must be calibrated to the intended swarm objective. Once these issues are corrected and real-robot experiments provide communication, safety, task, and resource measurements, the proposed framework can be evaluated against the criteria emphasized by the current Nature Collection.

11 Data and Code Availability

The numerical tables and figures in this manuscript are derived from the supplied simulation CSV logs. The manuscript makes no claim that these logs are a public benchmark or that they were generated by a physical robot platform. For public release, the final repository should contain the exact simulation implementation, parameter files, raw logs, analysis scripts, and a README specifying software versions and execution commands.

Author Contributions

Babak Rahmani conceived the study, formulated the methodology presented here, performed the analysis of the supplied simulation logs, and prepared the manuscript.

Competing Interests

The author declares no competing interests.

Ethics, Consent to Participate, and Consent to Publish declarations

Ethics, Consent to Participate, and Consent to Publish declarations: not applicable.

References

[1] Nature Portfolio, “Swarm Robotics — How to submit,” Collection page for npj Robotics, updated 2026. https://www.nature.com/collections/fhhcicdjad/how-to-submit.

[2] A. Bono, G. Fedele, and G. Franzè, “A swarm-based distributed model predictive control scheme for autonomous vehicle formations in uncertain environments,” IEEE Transactions on Cybernetics, vol. 52, no. 9, pp. 8876–8886, 2022. DOI: 10.1109/TCYB.2021.3070461.

[3] A. Gräfe, J. Eickhoff, M. Zimmerling, and S. Trimpe, “DMPC-Swarm: distributed model predictive control on nano UAV swarms,” Autonomous Robots, vol. 49, art. 28, 2025. DOI: 10.1007/s10514-025-10211-w.

[4] L. He, D. Tian, and Z. Xu, “Two-stage differential event-triggered distributed model predictive control for wheeled robot swarm systems,” Information and Control, vol. 55, no. 3, pp. 505–516, 2026. DOI: 10.13976/j.cnki.xk.2025.0322.

[5] M. Mesbahi and M. Egerstedt, Graph Theoretic Methods in Multiagent Networks. Princeton University Press, 2010.

[6] K.-K. Oh, M.-C. Park, and H.-S. Ahn, “A survey of multi-agent formation control,” Automatica, vol. 53, pp. 424–440, 2015.

[7] J. P. Desai, J. P. Ostrowski, and V. Kumar, “Modeling and control of formations of nonholonomic mobile robots,” IEEE Transactions on Robotics and Automation, vol. 17, no. 6, pp. 905–908, 2001.

[8] H. G. Tanner, G. J. Pappas, and V. Kumar, “Leader-to-formation stability,” IEEE Transactions on Robotics and Automation, vol. 20, no. 3, pp. 443–455, 2004.

[9] M. A. Lewis and K.-H. Tan, “High precision formation control of mobile robots using virtual structures,” Autonomous Robots, vol. 4, no. 4, pp. 387–403, 1997.

[10] R. W. Beard, J. Lawton, and F. Y. Hadaegh, “A coordination architecture for spacecraft formation control,” IEEE Transactions on Control Systems Technology, vol. 9, no. 6, pp. 777–790, 2001.

[11] R. Olfati-Saber, “Flocking for multi-agent dynamic systems: Algorithms and theory,” IEEE Transactions on Automatic Control, vol. 51, no. 3, pp. 401–420, 2006.

[12] W. Dong, “Robust formation control of multiple wheeled mobile robots,” Journal of Intelligent and Robotic Systems, vol. 62, no. 3, pp. 547–565, 2011.

[13] F. Xiao, L. Wang, J. Chen, and Y. Gao, “Finite-time formation control for multi-agent systems,” Automatica, vol. 45, no. 11, pp. 2605–2611, 2009.

[14] S. Mastellone, D. M. Stipanovic, C. R. Graunke, K. A. Intlekofer, and M. W. Spong, “Formation control and collision avoidance for multi-agent non-holonomic systems: Theory and experiments,” The International Journal of Robotics Research, vol. 27, no. 1, pp. 107–126, 2008.

[15] J. Ghommam, H. Mehrjerdi, M. Saad, and F. Mnif, “Formation path following control of unicycle-type mobile robots,” Robotics and Autonomous Systems, vol. 58, no. 5, pp. 727–736, 2010.

[16] P. D. Christofides, R. Scattolini, D. M. de la Peña, and J. Li, “Distributed model predictive control: A tutorial review and future research directions,” Computers & Chemical Engineering, vol. 51, pp. 21–41, 2013.

[17] R. R. Negenborn and J. M. Maestre, “Distributed model predictive control: An overview and roadmap of future research opportunities,” IEEE Control Systems Magazine, vol. 34, no. 4, pp. 87–97, 2014.

[18] W. B. Dunbar and D. S. Caveney, “Distributed receding horizon control of vehicle platoons: Stability and string stability,” IEEE Transactions on Automatic Control, vol. 57, no. 3, pp. 620–633, 2012.

[19] P. Wang and B. Ding, “Distributed RHC for tracking and formation of nonholonomic multi-vehicle systems,” IEEE Transactions on Automatic Control, vol. 59, no. 6, pp. 1439–1453, 2014.

[20] H. Li, Y. Shi, and W. Yan, “Distributed receding horizon control of constrained nonlinear vehicle formations with guaranteed γ-gain stability,” Automatica, vol. 68, pp. 148–154, 2016.

[21] R. Van Parys and G. Pipeleers, “Distributed MPC for multi-vehicle systems moving in formation,” Robotics and Autonomous Systems, vol. 97, pp. 144–152, 2017.

[22] G. Franzè, W. Lucia, and F. Tedesco, “A distributed model predictive control scheme for leader-follower multi-agent systems,” International Journal of Control, vol. 91, no. 2, pp. 369–382, 2018.