DOI: 10.35598/mcfpga.2026.016

Hardware-Aware Mathematical Model of Communication-Robust Control for UAV Swarms on Microcontroller and FPGA Platforms

Yehor Miezientsev
ORCID 0009-0004-2228-0303
Department of Network and Internet Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
egor.mezencev@gmail.com

Yurii Kravchenko
ORCID 0000-0002-0281-4396
Department of Network and Internet Technologies
Faculty of Information Technology
Taras Shevchenko National University of Kyiv
Kyiv, Ukraine
yurii.kravchenko@knu.ua

***Abstract*—The abstracts present a hardware-aware mathematical model for communication-robust decentralized control of unmanned aerial vehicle swarms on embedded platforms. The model connects a dynamic communication graph, packet delivery probability, age of information, and embedded resource constraints with the local decision policy of each agent. The proposed representation keeps centralized training outside the flight platform while preparing a compact decentralized policy for real-time execution on microcontrollers, microprocessors, or FPGA-assisted nodes. The architecture includes a communication reliability estimator, a disturbance filter, a graph aggregation module, and a compact policy core. This makes it possible to evaluate a swarm-control method not only by mission quality, but also by latency, memory, radio bandwidth, and energy limitations that are critical for cyber-physical embedded systems.**

Keywords*—*UAV swarm, embedded control, microcontroller, FPGA, multi-agent reinforcement learning, communication reliability, mathematical modeling.

I. INTRODUCTION

Unmanned aerial vehicle groups are increasingly considered as distributed cyber-physical systems in which sensing, communication, decision making, and actuation are performed by many autonomous agents rather than by a single central controller. This architecture is attractive for monitoring, search and rescue, infrastructure inspection, and other missions where the loss of one agent must not stop the whole group. However, the same decentralization creates a difficult engineering problem: each UAV has access only to local observations and delayed messages from neighboring agents, while the quality of wireless links changes due to mobility, obstacles, interference, and deliberate electronic countermeasures [1], [2].

The considered problem belongs to the areas of mathematical modeling of information signals and systems and to the architecture of specialized embedded computing systems. In this work the control policy is treated as an information-processing system in which sensor observations, radio-channel quality indicators, and neighboring messages are converted into control actions under latency, memory, energy, and bandwidth constraints. Therefore, the algorithm cannot be evaluated only by mission reward or trajectory accuracy. It also has to satisfy the constraints on inference latency, memory footprint, communication bandwidth, and power consumption [5], [6].

This paper focuses on a hardware-aware representation of communication-robust multi-agent control, namely on transforming a mathematical model of decentralized UAV coordination into an embedded execution architecture. The model is intentionally compact as it preserves the communication properties required for robust cooperation while remaining suitable for later implementation on a microcontroller-only node, an MCU-plus-FPGA node, or a system-on-chip FPGA prototype [3]-[8].

II. EMBEDDED CONTROL PROBLEM STATEMENT

Let the swarm contain N agents. Each UAV is represented by a local state vector, a control vector, a local observation, and a set of messages received from neighboring agents. The topology of interaction is not constant because distances between UAVs, antenna orientation, interference level, and channel congestion change with time. It is therefore convenient to describe the communication subsystem by a time-varying graph [4]

$$G_t = (V, E_t), E_t \subseteq V \times V. \tag{1}$$

Here V is the set of UAVs, and the edge set contains communication links that satisfy the effective communication radius and minimum channel-quality conditions. In real missions, the presence of an edge does not guarantee useful information exchange. A packet may be delayed, damaged, or not delivered. Therefore, each message is additionally associated with a delivery probability and with an age-of-information value, which characterizes how old the last trusted data from a neighbor is at the decision moment.

The compact state used during centralized training is defined as

$$s_i(t) = \{x_i, z_i, b_i, G_t, p_i, \Delta_i\}. \tag{2}$$

In (2), the vector of embedded resources includes available memory, processor load, energy reserve, and queue occupancy of the communication interface. During deployment only local observations, message history, and reliability estimates are used by the flight node, while the full state is available only in simulation or during offline training. This corresponds to the centralized-training and decentralized-execution paradigm and is suitable for embedded platforms because expensive learning is performed outside the aircraft, whereas the flight node performs only inference.

III. Communication reliability and objective function

The probability of receiving a usable message from a neighboring UAV is modeled as a smooth function of distance, disturbance level, and age of information [1], [2]:

$$p_{ij}(t)=\sigma\left(\beta_0-\beta_1 d_{ij}(t)-\beta_2\rho(t)-\beta_3\Delta_{ij}(t)\right). \quad (3)$$

The disturbance term can include measured packet loss, received signal degradation, detected jamming patterns, or a normalized interference level. Such a formulation does not require the policy network to know the exact physical source of degradation. Instead, it receives a compact reliability estimate that can be obtained from low-level radio statistics and timestamped messages.

For embedded UAV control, the objective should reflect two groups of requirements. The first group is mission-related: keeping formation, reaching target points, avoiding collisions, and maintaining group coherence. The second group is hardware-related: satisfying the control-loop period, limiting memory use, reducing radio load, and preserving energy. These requirements can be combined in the following optimization criterion:

$$J=E\left[\sum_{t=0}^{T}\gamma^t\left(C_t+\lambda_E E_t+\lambda_\tau\tau_t+\lambda_M M_t+\lambda_B B_t\right)\right]. \quad (4)$$

In (4), the mission cost is complemented by penalties for energy consumption, end-to-end inference and actuation latency, memory usage, and radio bandwidth. The coefficients define the importance of hardware limitations. The policy is feasible only if the embedded constraints are satisfied: the control-loop latency must not exceed the selected real-time period, the model size must fit the available memory, and the exchanged data must not overload the radio channel.

IV. Proposed embedded execution architecture

The proposed approach separates the learning model from the execution model. During training, graph attention mechanisms can use a richer representation of inter-agent relations [3], [4]. Before deployment, the policy is compressed by selecting a limited neighborhood size, quantizing network weights, reducing hidden dimensions, and replacing expensive operations with fixed-point approximations [5]. A microcontroller can execute the reliability estimator and a compact recurrent policy, while an FPGA or system-on-chip FPGA can accelerate message gating, vector aggregation, and matrix operations [7], [8].

A compact form of the local policy used for deployment can be written as

$$a_i(t)=\pi_\theta\left(o_i,h_i,c_i\right),c_i=Agg\left(p_{ij}m_j\right). \quad (5)$$

Here the action is generated from the local observation, recurrent memory state, and aggregated communication context. The aggregation block uses reliability weights, so outdated or uncertain messages influence the decision less than fresh and trusted messages. This makes the communication channel a part of the control model rather than an external disturbance.

Fig. 1 presents the proposed execution architecture. The input layer receives onboard sensor data and neighboring messages. The reliability estimator calculates message probability, age of information, and a confidence coefficient for every available neighbor. The disturbance filter suppresses unreliable, duplicated, or outdated messages and performs short-term prediction when a link is temporarily lost. The graph or attention aggregation block converts a variable number of stabilized neighbor vectors into a fixed-length context vector. The local policy core then generates an action for the autopilot or actuator interface.

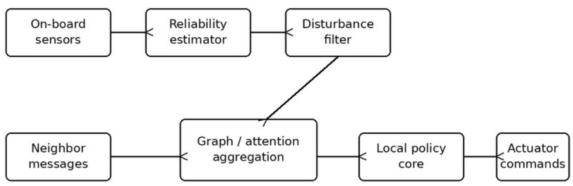

Fig. 1. Embedded architecture of communication-robust decentralized control.

The architecture supports three typical embedded variants: a microcontroller-only implementation with fixed-point software inference; an MCU-plus-FPGA implementation where aggregation and the policy core are accelerated in programmable logic; and a system-on-chip FPGA implementation for scalable prototypes and hardware-in-the-loop verification [5]-[8].

V. Discussion of modeling results

The model makes it possible to identify several operating regimes of the swarm without changing the general controller structure. In the transparency regime, packet losses are rare and the graph remains dense, so the controller mainly removes noise from current neighbor messages. In the compensation regime, losses become noticeable but are still short enough to be reconstructed from recent history; the filter and reliability estimator become essential for keeping the input distribution stable. In the fragmentation regime, the communication graph is sparse and the probability of valid messages becomes low; the controller should switch to safe local behavior and reduce dependence on group coordination.

For a hardware designer, these regimes are useful because they define what must be computed in real time. In the transparency regime, the main load is the policy core. In the compensation regime, a significant part of computation is transferred to timestamp analysis, confidence calculation, and aggregation of incomplete messages. In the fragmentation regime, the radio channel should not be overloaded by retransmission attempts, while the embedded controller should prioritize local safety and energy conservation [7], [8].

The most important advantage of the proposed model is that it avoids a gap between high-level multi-agent reinforcement learning design and embedded implementation.

Classical simulations often assume that agents exchange complete state vectors with negligible delay. In contrast, the proposed formulation treats the communication channel as part of the controlled system. This makes the policy more realistic for UAV swarms, where the information space is formed by radio measurements, packet age, and topology changes [5]-[8].

VI. Conclusion

The abstracts propose a hardware-aware mathematical model of communication-robust decentralized control for UAV swarms on embedded platforms. The model extends the multi-agent control formulation by including a dynamic communication graph, packet delivery probability, age of information, and an embedded resource vector. The objective function combines mission quality with penalties for energy consumption, latency, memory use, and bandwidth load. This makes it possible to evaluate the proposed control policy with respect to both mission performance and real-time embedded implementation constraints on microcontroller and FPGA platforms.

The proposed execution architecture contains a communication reliability estimator, disturbance filter, graph aggregation module, and compact policy core. It can be implemented in a microcontroller-only configuration for small swarms or in an MCU-plus-FPGA configuration for deterministic low-latency inference. Further research will focus on quantization-aware training, hardware-in-the-loop verification, and experimental comparison of software and FPGA-accelerated implementations under controlled radio-channel degradation.

References

[1] R. Lowe, Y. Wu, A. Tamar, J. Harb, P. Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," in Proc. Advances in Neural Information Processing Systems, vol. 30, 2017.

[2] C. Yu, A. Velu, E. Vinitsky, J. Gao, Y. Wang, A. Bayen, and Y. Wu, "The surprising effectiveness of PPO in cooperative multi-agent games," in Proc. Advances in Neural Information Processing Systems, vol. 35, pp. 24611-24624, 2022.

[3] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in Proc. Advances in Neural Information Processing Systems, vol. 30, pp. 5998-6008, 2017.

[4] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in Proc. International Conference on Learning Representations, 2017.

[5] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in Proc. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 2704-2713.

[6] J. Chen and X. Ran, "Deep learning with edge computing: A review," Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, 2019.M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.

[7] M. Procaccini, A. Sahebi, and R. Giorgi, "A survey of graph convolutional networks (GCNs) in FPGA-based accelerators," Journal of Big Data, vol. 11, art. no. 163, 2024, doi: 10.1186/s40537-024-01022-4.

[8] Y. Yang et al., "Synetgy: Algorithm-hardware co-design for ConvNet accelerators on embedded FPGAs," in Proc. ACM/SIGDA International Symposium on Field-Programmable Gate Arrays, 2019, pp. 23-32.