ISSN: 2832-790X
Online Journal of
Robotics & Automation Technology
DOI: 10.33552/OJRAT.2025.04.000584
Iris Publishers

Mini-Review Article

Distributed AI for Robotics and Automation: From Multi-Agent Coordination to Federated Edge Learning

Iacovos Ioannou[12]* and Vasos Vassiliou[12]

[1]*CYENS Centre of Excellence, Nicosia, Cyprus*

[2]*Department of Computer Science, University of Cyprus, Nicosia, Cyprus*

***Corresponding author:** Iacovos Ioannou, CYENS Centre of Excellence, Nicosia, Cyprus Department of Computer Science, University of Cyprus, Nicosia, Cyprus

Received Date: September 09, 2025

Published Date: September 17, 2025

Abstract

Distributed AI (DAI)-encompassing multi-agent coordination, distributed control, and federated/edge learning-is now central to automating robotic work cells and mobile fleets where tight latency budgets, data sovereignty, and operational resilience dominate system design. Rather than a monolithic "brain in the cloud," a DAI approach decomposes autonomy into local perception/decision/control loops with only the necessary coordination across peers. This mini-review distils a practical blueprint:

(i) Multi-robot task allocation and consensus to decide who does what, when;

(ii)Distributed model predictive control (DMPC) to decide how robots move together safely and efficiently;

(iii)Federated learning (FL) to adapt perception and predictive models on-device; and

(iv)Runtime safety envelopes via Control Barrier Functions (CBFs). We emphasize edge-first layering (cell → line/fleet → plant/cloud), standards-based interoperability (ROS 2/DDS and OPC UA), and twin-first validation. The core message is actionable: combining auctions/consensus, DMPC, FL, and CBFs over deterministic ROS 2/DDS transports provide a tractable path from pilots to brownfield scale-up without sacrificing safety, uptime, or auditability.

Keywords: Distributed AI; Multi-robot systems; Federated learning; Dis- tributed MPC; ROS 2; DDS; OPC UA; Digital twin; Multi-agent RL; Runtime safety

Abbreviations: DAI: Distributed Artificial Intelligence; MARL: Multi-Agent Reinforcement Learning; DMPC: Distributed Model Predictive Control; FL: Federated Learning; CBF: Control Barrier Function; AMR: Autonomous Mobile Robot.

Introduction

Modern robotic lines and AMR fleets are cyber-physical systems (CPS) with geographically and functionally distributed intelligence. Centralized orchestration often collapses under real- time jitter, privacy constraints, and single points of failure. The DAI framework in [1] addresses these limitations by structuring autonomy around cooperating agents that keep fast loops local while exposing lean, standards-based interfaces for coordination. This framing bridges long-standing strands of research-task allocation and consensus for multi-robot cooperation [2-4], dis- tributed optimal control for joint motion planning [5], and edge learning for perception & maintenance [6, 7]-into a deployable, modular stack. Interoperability and determinism are achieved by combining ROS 2's publisher/subscriber graph with DDS Quality- of-Service (QoS) policies for reliability, deadlines, and liveliness [8, 9]. OPC UA (IEC 62541) bridges IT/OT boundaries via secure information models [10]. Digital twins de-risk change management by stress-testing policies and control updates before rollout [11, 12]. Together, these ingredients transform autonomy from a lab artifact into a production capability with traceable behavior and service-level objectives (SLOs).

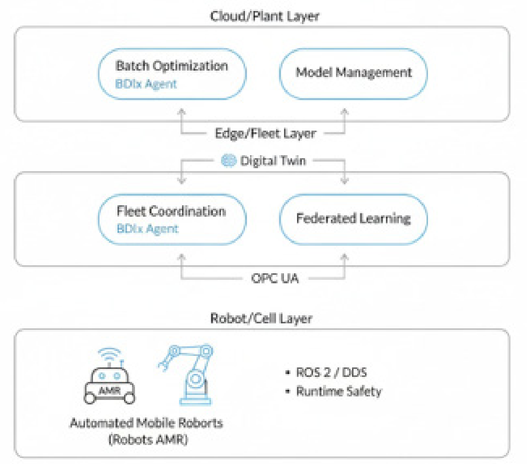

Figure 1: Layered DAI architecture with BDIx agents. BDIx agents run at the Edge/Fleet layer (coordination) and Plant/Cloud layer (batch optimization/model management). The Robot/Cell layer hosts ROS 2/DDS loops and CBF supervisors; OPC UA bridges IT/OT; a Digital Twin validates updates before staged roll- out.

DAI Framework (According to [1])

Layering and roles. Robot/Cell agents (ROS 2 nodes on robots/PLCs) close sub-10ms loops for actuation, perception pre- processing, and device health. The Edge Line/Fleet layer arbitrates shared resources (e.g., aisle right-of-way, tool/changeover windows) and schedules jobs under local constraints. The Plant/Cloud layer handles slower-time-scale optimization (e.g., shift-level scheduling), model lifecycle (train/validate/sign), and global KPIs. Clear role separation prevents control interference and simplifies certification. Communication. DDS QoS provides predictable pub/sub behavior (bounded latency, reliable delivery, liveliness monitoring). Topics encode state, intents, and bids; namespaces isolate cells while enabling selective cross-cell exchange. OPC UA exposes equipment hierarchies and alarms to MES/SCADA and supports secure handshakes for command authorization.

Learning at the edge. FL aggregates model updates rather than raw data, preserving privacy while leveraging fleet-wide experience [6, 7]. Non-IID data and connectivity heterogeneity are handled via client sampling, adaptive learning rates, and compression (e.g., sparsification/quantization). On-device drift detectors trig- ger light-weight personalization between federated rounds. Assurance and safety. CBF-based supervisors wrap learned or optimal controllers. At each cycle, a small QP minimally modifies the command to keep the system within a certified safe set [13]. This yields graceful degradation during anomalies (sensor dropout, packet loss) and makes learning compatible with safety standards.

Automation Patterns & How-To

Who does what, when

Use the Gerkey-Matarić taxonomy to match problem structure to allocation methods [2]. Market-based approaches (auctions, contract nets) scale gracefully under changing workloads by converting tasks into bids scored on travel time, energy, capability, and deadlines [3]. Distributed consensus stabilizes shared beliefs (e.g., queue states) and enables coordinated decisions without a central arbiter [4]. In non-stationary environments, MARL can learn dispatching heuristics online; policy updates should be rate-limited, with rollback hooks and CBF-wrapped exploration to bound risk.

How robots move together

DMPC couple's local trajectory optimizers through exchanged plans or compact intent messages [5]. Practical deployments rely on:

(i) short horizons with warm starts from the last feasible plan;

(ii) constraint tightening to absorb model mismatch;

(iii) asynchronous updates to tolerate network jitter; and

(iv) priority rules for deadlock-prone spaces (narrow aisles, intersections). Feasible- but-suboptimal plans are preferred over brittle global optima; CBF layers guarantee collision and speed-limit compliance if neighbours misbehave or packets drop.

Learning at the edge

For vision/QA and predictive maintenance, FL avoids central data pooling. Choose compact backbones that fit device memory and thermal envelopes; schedule federated rounds off-shift or during charging. Handle device churn with partial participation; use twin- generated augmentations for rare defect classes. Maintain a signed model registry; only models that pass twin-based regression and safety checks are promoted to production.

Trust and safety

CBFs define forward-invariant safe sets from task-relevant constraints (separation, speed near humans, keep-out zones). Each cycle solves a tiny QP to project the nominal action back into the safe set with minimum deviation [13]. This makes safety orthogonal to the controller choice (PID, DMPC, RL) and provides auditable guarantees that align with certification audits.

Brownfield checklist (quick start)

- Pilot a cell: Map robot/PLC I/O to ROS 2 nodes; tune DDS QoS (reliable, deadline, liveliness) for state/control topics; ex- pose PLC tags via OPC UA with role-based access.
- Edge coordinator: Start with auctions for tasking; add DMPC for shared-space traffic; encode simple priority rules to break ties.
- FL loop: Deploy a compact model per robot; run nightly/weekly federated rounds; enable on-device drift triggers for interim adaptation; track model lineage.
- Safety: Wrap controls with CBF supervisors; add health checks and watchdogs; PTP time-sync for consistent logs and event ordering.
- Twin-first: Validate policies and models in the digital twin; use staged rollouts (canary → cell → line); monitor KPIs and auto-rollback on regression.

Case Vignette: AMRs with Edge Vision & DMPC

Consider a kitting line with heterogeneous AMRs. Each robot runs a ROS 2 stack providing localization, battery/health telemetry, and an on-device defect detector for bins. A line-edge coordinator exe- cutes periodic auctions that assign pick missions using bid scores combining distance, residual battery, and congestion estimates. A DMPC layer regulates hallway right-of-way: robots share planned velocities over DDS; when packets arrive late, CBF supervisors bound speeds and enforce separation until the plan refreshes. Federated rounds aggregate detector updates during the night shift, while a twin replays the next day's job mix and human traffic pat- terns to validate the candidate model before promotion. This combination reduces dispatch oscillations, smooths hallway flows, and contains risk: when one AMR reboots mid-mission, others replan locally while CBFs ensure safe yields.

Conclusion

DAI turns autonomy into an engineered system property: agents negotiate tasks, coordinate motion, and adapt perception locally, while twins, DDS/OPC UA, and CBFs keep behavior predictable, explainable, and certifiable. The framework in [1] offers a pragmatic path from pilots to fleet-scale deployments-with measurable gains in throughput and availability-without surrendering safety or data control.

Acknowledgements

The author thanks colleagues at CYENS and the University of Cyprus.

Conflict of Interest

The author declares no conflicts of interest.

References

1. Ioannou I, Nagaradjane P, Vassiliou V, Pitsillides A, Christophorou C (2024) Distributed Artificial Intelligence for 5G/6G Communications: Frameworks with Machine Learning (1st ed.). CRC Press.
2. Gerkey BP, Matarić MJ (2004) A formal analysis and taxonomy of task allocation in multi-robot systems. Int J Robotics Res 23(9): 939-954.
3. Dias MB, Zlot R, Kalra N, Stentz A (2006) Market-based multirobot coordination: a survey and analysis. Proc IEEE 94(7): 1257-1270.
4. Olfati Saber R, Fax JA, Murray RM (2007) Consensus and cooperation in networked multi-agent systems. Proc IEEE 95(1): 215-233.
5. Camponogara E, Jia D, Krogh BH, Talukdar S (2002) Distributed model predictive control. IEEE Control Syst Mag 22(1): 44-52.
6. McMahan HB, Moore E, Ramage D, Hampson S, Aguëra y Arcas B (2017) Communication-efficient learning of deep networks from decentralized data. AISTATS. arXiv:1602.05629.
7. Bonawitz K, Eichner H, Grieskamp W (2019) Towards federated learning at scale: System design. arXiv:1902.01046.
8. Maruyama Y, Kato S, Azumi T (2016) Exploring the performance of ROS 2. Proc 13th ACM Int Conf.
9. Object Management Group (2015) Data Distribution Service (DDS) v1.4.
10. Mahnke W, Leitner S-H, Damm M (2009) OPC Unified Architecture. Springer.
11. Negri E, Fumagalli L, Macchi M (2017) A review of the roles of digital twin in CPS-based production systems. Procedia Manufacturing 11: 939-948.
12. Lu Y, Liu C, Wang K, Huang H, Xu X (2020) Digital twin-driven smart manufacturing: reference model and applications. Robotics Comput-Integr Manuf 61: 101837.
13. Ames AD, Xu X, Grizzle JW, Tabuada P (2017) Control barrier function based quadratic programs for safety critical systems. IEEE Trans Autom Control 62(8): 3861-3876.