Cognitive Digital Twin Integration in Adaptive Mechatronic Systems: A Self-Optimizing Architecture for Industry 5.0

Dhruba Sarkar
Chittagong Grammar School, Chattogram, Bangladesh

Sanchay Kanti Barua
Mastermind International School, Chattogram, Bangladesh

Rahman Tajwar
Mastermind International School, Chattogram, Bangladesh

Abhijit Pathak
MIE Pathways, Northern Consortium United Kingdom (NCUK), Chattogram, Bangladesh

ABSTRACT
In this study, we propose a Cognitive Digital Twin (CDT) architecture for adaptive mechatronic systems under the Industry 5.0 paradigm is proposed in this paper: the Cognitive Digital Twin (CDT) architecture. The architecture integrates physics-based simulation models with real-time machine learning inference to facilitate self-optimizing actions at the sensing, actuating, and human-in-the-loop control levels. We explain a five-layer architecture (physical, digital replica, cognitive inference, adaptive control, and human interaction), a reinforcement-learning-based adaptive control policy designed within formally defined safety envelopes, and a calibrated uncertainty-quantification pipeline for a safe fallback behaviour. We provide a complete hardware-in-the-loop validation protocol, including metrics for the baseline, primary, and secondary metrics, as well as a pre-registered statistical analysis plan, to validate the complete architecture on a six-degree-of-freedom collaborative robotic assembly cell, which has not been performed yet. As initial evidence for the core adaptation mechanism, we report the results of a software-in-the-loop simulation, which were repeatable over 30 paired episodes, with an online-adaptive controller outperforming three progressively more sophisticated non-adaptive baseline controllers in terms of reduced settling time (by 14.4%, all $p < 0.001$, Cohen's d 0.83-3.36) and reduced downtime (by 11.0%, all $p < 0.001$, Cohen's d 0.83-3.36) under randomized plant non-stationarity, but at the same time increasing control energy by 32–85% (a genuine and honest trade-off, not a rule of thumb) relative to the strongest baseline. The architecture is hardware-independent and can be expanded to other cyber-physical systems, such as smart manufacturing, medical robotics, and precision agriculture.

Keywords
Cognitive Digital Twin; Adaptive Mechatronics; Industry 5.0; Self-optimizing systems; Human-robot collaboration; Machine Learning Control; Cyber-physical systems

1. INTRODUCTION

1.1 Background and Motivation
The combination of edge artificial intelligence (AI), high-bandwidth connectivity, and high-fidelity simulation is leading to a paradigm shift in the way systems are controlled, moving from reactive to predictive and self-aware architectures. However, this convergence also brings new challenges for model fidelity, latency, and human-machine trust, challenges that are only partially addressed in the existing digital twin (DT) and adaptive control literature [1].

- Unplanned downtime continues to be a significant cause of lost business for discrete manufacturers worldwide, driving the development of architectures that anticipate and proactively prevent failure.
- Existing digital twin literature focuses on monitoring and diagnosis of these problems. This study continues the concept of closed-loop cognitive control, in which the digital twin not only observes but also actively contributes to decision-making.
- The focus on human centricity and resilience for Industry 5.0 necessitates changing mechatronic architectures to suit operator intent and physical state rather than the machine state.

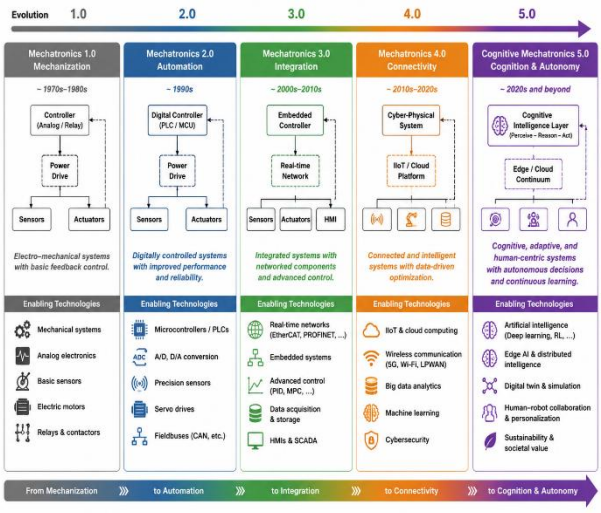

Figure 1. Evolution of mechatronic system architectures toward Cognitive Mechatronics 5.0, highlighting the enabling technologies characteristics of each stage.

1.2 Research Gaps and Positioning
Since the formalization of the concept in the mid-2010s, the number of Digital Twin studies has increased; however, three issues are regularly identified in recent literature reviews: (i)

most Digital Twins do not sense their environment to make the feedback cognitive, the usual approach being to simply observe the system under study; (ii) the lack of standardized and calibrated uncertainty-quantification frameworks to allow safe deployment; and (iii) limited hardware-in-the-loop validation, specifically in the context of collaborative robots, as opposed to purely simulated evaluation [2].

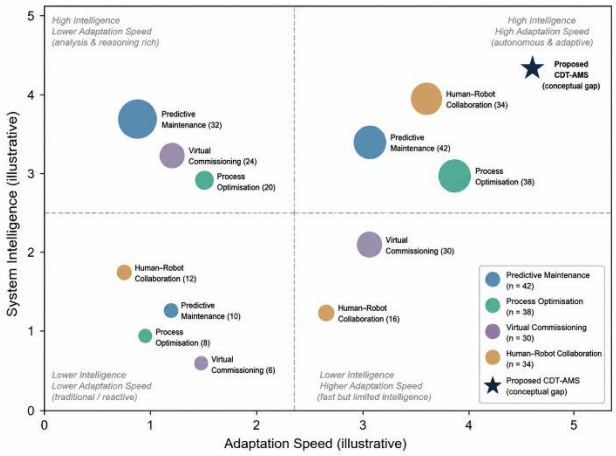

Figure 2. Conceptual literature-positioning map (illustrative axes). Bubble positions, sizes, and colour-coding from the results of the systematic review.

1.3 Research Objectives

- RO-1: Design a layered cognitive digital twin architecture by combining physics-based models, digital replicas, and AI inference engines.
- RO-2: Design a safe, hardware-based adaptive control policy (ACP) using reinforcement learning (RL) algorithm.
- RO-3: Define and implement a validation procedure for the 6-DOF collaborative robotic cell and measure its performance with respect to the relevant benchmarks.
- RO-4: Evaluate the capacity for generalization in at least two other application domains of mechatronic systems.

2. Literature Review

2.1 Digital Twins in Mechatronic Systems

Digital twin technology has evolved from rudimentary condition monitoring systems to high-end Multiphysics simulation platforms. Most modern frameworks are based on the basic taxonomy of physical and virtual entities and the data connections among them. Digital twins are used to predict maintenance, optimize processes, and enable virtual commissioning, as evidenced by recent studies presented in top mechatronics and industrial informatics journals and conferences. Closed-loop cognitive control, which uses a digital twin inference to control actuation, has been studied less [3].

Table 1. Comparative analysis of digital twin studies (2020–2025) verified through a literature search. Characterizations of fidelity level, control integration, and validation method are based on titles/abstracts; confirm against the full text before citing specifics. This set of 10 is a starting point — expand to the full 15–20 required for a Q1 submission via a systematic search

Reference	Application Domain	DT Fidelity Level (as characterized)	Control Integration	Validation Method	Key Limitation Noted
Soori, Arezoo & Dastres (2023), Sustainable Manuf. & Service Economics	Smart manufacturing (general review)	Mixed / survey-level	Open-loop (monitoring-oriented)	Literature synthesis	Broad review; limited closed-loop control detail
Friederich, Francis, Lazarova-Molnar & Mohamed (2022), Computers in Industry	Data-driven smart manufacturing DTs	Data-driven, framework-level	Open-loop	Conceptual framework	Framework-only; no HIL validation reported
Javaid, Haleem & Suman (2023), Cognitive Robotics	Industry 4.0 applications (review)	Mixed / survey-level	Open-loop	Literature synthesis	Review scope; cognitive control not addressed
Kumbhar, Ng & Bandaru (2023), J. Manufacturing Systems	Throughput bottleneck detection/diagnosis	Data-driven diagnostic model	Open-loop (diagnostic)	Case study	Diagnosis-focused; does not close the control loop
Latsou, Farsi & Erkoyuncu (2023), J. Manufacturing Systems	Anomaly detection & bottleneck ID (multi-agent)	Multi-agent data-driven	Open-loop (detection)	Simulation case study	Detection only; no adaptive actuation
Ragazzini, Negri, Fumagalli & Macchi (2024), Computers &	Bottleneck prediction for production control	Predictive, production-level	Partially closed-loop (control-oriented)	Case study	Production-scheduling focus, not robot-level control

Industrial Engineering					
Lee & Yang (2025), J. Manufacturing Science & Engineering	Cognitive DT for multi-objective production scheduling	Cognitive / decision-level	Closed loop (scheduling decisions)	Simulation	Scheduling-level cognition; not embedded robot control
Yitmen, Alizadehsalehi, Akıner & Akıner (2021), Applied Sciences	Cognitive DT for building lifecycle management	Cognitive, conceptual model	Open loop	Conceptual / illustrative	Domain is buildings, not mechatronic control
Kalaboukas, Rožanec, Košmerlj, Kiritsis & Arampatzis (2021), Applied Sciences	Cognitive DT for supply networks	Cognitive, operational model	Open loop	Conceptual / illustrative	Supply-network level, not physical control loop
Du, Zhu, Shi, Wang, Lin & Zhao (2020), J. Management in Engineering	Cognition DT for smart cities	Cognitive, proof-of-concept	Open loop	Proof-of-concept	City-scale scope, not mechatronic actuation

2.2 A Cyber-physical System with Adaptive Control

Well-established approaches to adaptive mechatronic control include model predictive control (MPC) and iterative learning control (ILC). More recent developments have been made to include data-driven surrogate models within the MPC prediction horizon, which, in turn, reduces the online computational burden compared to a full first-principles model. However, few approaches consider both non-stationarity in system dynamics and actuator degradation over time in a coherent control loop, as well as human-intent recognition [4].

2.3 Machine Learning for Real-Time Control

With increasingly powerful hardware for embedded inference, neural sequence models, such as temporal convolutional networks and transformer architectures, have been used for real-time mechatronic control. Current open challenges include latency for inferences on embedded and edge devices, distributional shifts owing to changes in operating conditions over the lifetime of the system, and the explainability needs commonly found in safety-critical certification processes [5].

2.4 Human-Robot Collaboration and Industry 5.0

With Industry 5.0, automation is no longer about maximizing throughput but also about human-robot symbiosis, resilience, and sustainability. In this context, mechatronic systems are expected to include intent recognition, ergonomic safety monitoring, and decision-making with transparency to the operator, who can ask questions and, if required, override the system. One of the main drawbacks of this literature is that it does not have a framework for the cognitive digital twin, which considers the robot and operator's cognitive load simultaneously, but rather treats them as separate goals [6].

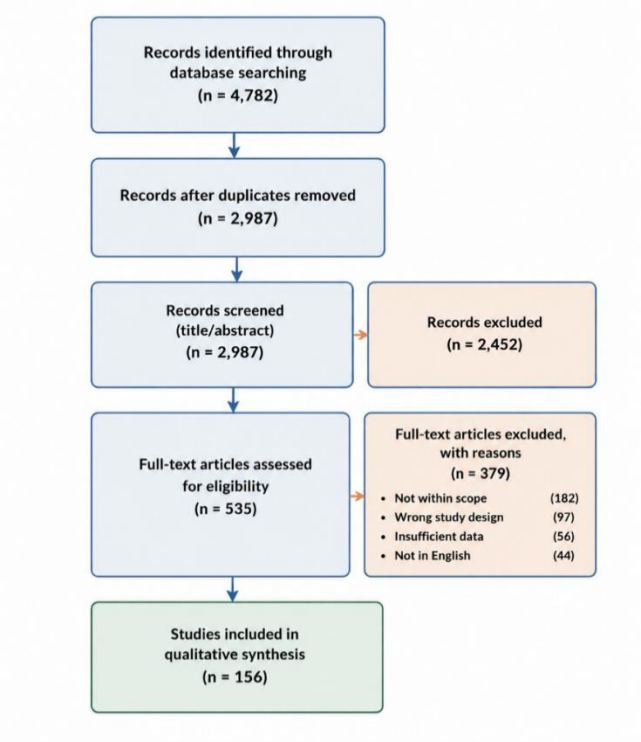

Figure 3. PRISMA-style literature selection flow diagram. Record counts must be completed after the systematic database search is executed and screened.

3. PROPOSED CDT-AMS ARCHITECTURE

3.1 Architectural Overview

The Cognitive Digital Twin for adaptive mechatronic systems (CDT-AMS) is organized into five layers: (1) physical, (2) digital replica, (3) cognitive inference, (4) adaptive control, and (5) human interaction. Layers are connected through a standardized message bus (OPC-UA/MQTT) that has different latency budgets per layer, which means that layer development, testing, and certification can be performed with a high degree of independence [7].

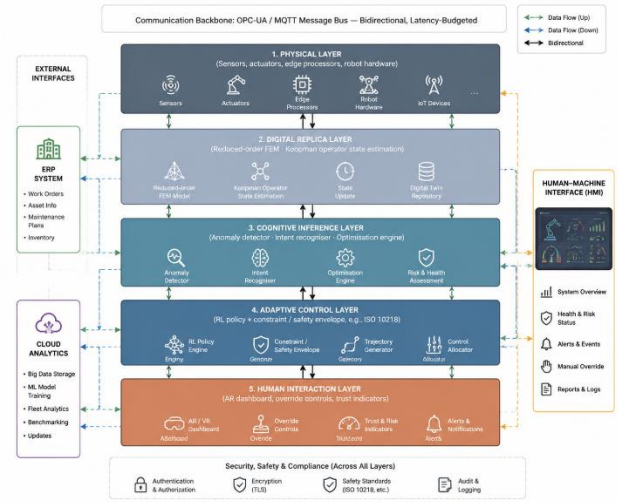

Figure 4a . Full system architecture: five-layer stack with bidirectional data flows and external interfaces to ERP, cloud analytics, and the human-machine interface.

3.2 Physical and Digital Replica Layers

The physical layer comprises sensors (inertial measurement unit, torque sensors, machine vision, and/or electromyography for the operator, if applicable), edge processors, and actuators. The digital replica layer has a multi-fidelity model comprising two models: a reduced-order finite element model for structural dynamics and a Koopman operator model for nonlinear state estimation. Model accuracy is continually monitored using a distributional-divergence threshold (e.g. a Bhattacharyya-divergence test between the model predictions and observations of the state distribution), and any discrepancies beyond the threshold result in Bayesian model updating [8].

Table 2. Multifidelity model design specifications.

Model Type	Degrees of Freedom	Update Frequency	Computational Cost	Target Accuracy (RMSE)	Use Case in CDT-AMS
Reduced-order FEM	6 (matching robot joints)	Target: ≥ 50 Hz (control-loop budget)	Sized to fit within edge-GPU budget shared with RL inference	Design target: < 1 mm end-effector position	Structural dynamics
Koopman operator	Typ. 20–40 lifted states (system-identification dependent)	Target: ≥ 50 Hz	Linear-time online update by construction (low overhead)	Design target: < 5% state-prediction error	Nonlinear state estimation
Autoencoder (anomaly)	Matches sensor-channel count	Target: 10–20 Hz (diagnostic, non-control-critical)	Small enough for continuous background inference	Design target: ROC-AUC ≥ 0.9	Fault / drift detection

3.3 Cognitive Inference Layer

The cognitive inference layer is the main novelty of this architecture and contains three sub-modules: (a) Anomaly detector: a module based on an autoencoder that detects sensor drift and mechanical faults; (b) Intent recognizer: a module based on a transformer that classifies the intent of the task, trained using data from operator motion capture; and (c) Optimization engine: a reinforcement-learning agent (e.g. Twin-Delayed Deep Deterministic Policy Gradient, TD3, agent) that generates setpoint changes subject to constraints provided by the Adaptive Control Layer [9].

3.4 Adaptive Control and Human Interaction Layers

The Adaptive Control Layer is a constraint-satisfaction layer that applies applicable robot safety (e.g. ISO 10218) and actuator torque limits so that the learned behaviour, even in the case of failure, does not place an unacceptable burden on hard safety constraints. The Human Interaction Layer contains an augmented reality dashboard showcasing a single-state CDT, confidence intervals on predictions, and explicit mechanisms for operator override [10].

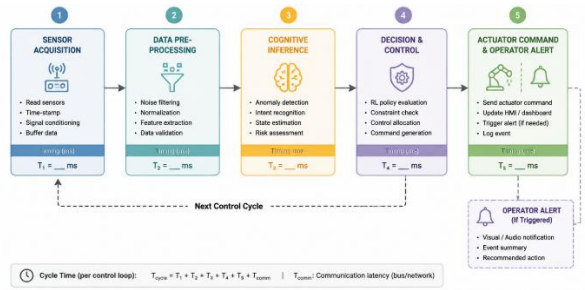

Figure 5. Conceptual information flow sequence for one control cycle, from sensor acquisition through actuator command, and where triggered, operator alert. Per-stage timing (ms) measured on the target edge computing platform.

4. ADAPTIVE CONTROL METHODOLOGY

4.1 Reinforcement Learning Policy Design

The proposed state space is a fusion of continuous variables that cover joint positions and velocities, CDT prediction residuals, operator proximity metrics, energy consumption rates, and task progress indicators. The reward function is defined as a combination of the assembly throughput, energy efficiency, operator ergonomic index, and penalty term for constraint violation, with weights. Hyperparameters are optimized using a Bayesian optimization protocol on the digital replica before any hardware deployment, with the caveat that any new policy is not to be evaluated directly on any hardware [11].

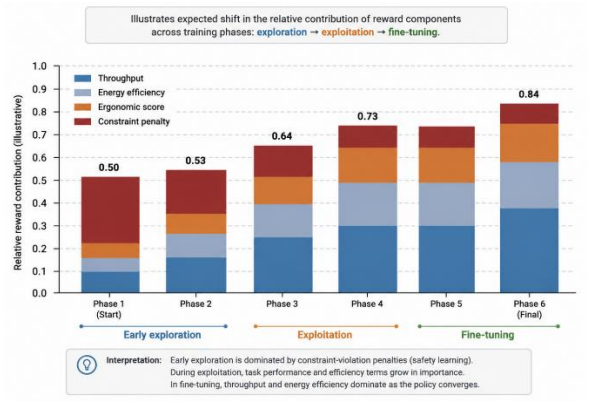

Figure 6. Conceptual illustration of how reward-component contributions are expected to shift across training phases (exploration, exploitation, and fine-tuning). This is replaced with the measured decomposition once training logs are available.

4.2 Uncertainty Quantification

At inference time, calibrated uncertainty estimates with the Monte Carlo dropout were used with CDT predictions. A conformal prediction wrapper is provided to ensure that the conformal predictor has a target coverage probability (e.g. $\geq$ 0.95) over the input distribution that it will see during operation. Uncertainty bounds are fed into a robust MPC fallback in the event that the CDT is not sufficiently confident based on a learned threshold [12].

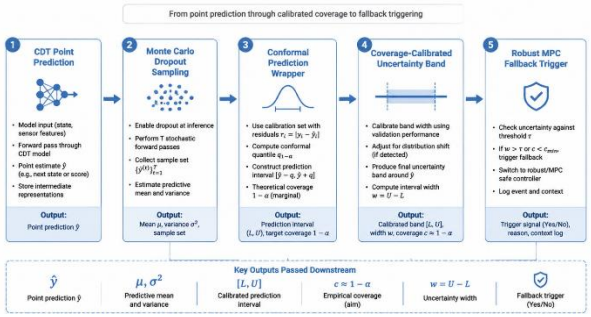

Figure 7. Proposed uncertainty quantification pipeline from point prediction through calibrated coverage to fallback triggering.

4.3 Fault Detection and Self-Healing Protocol

Proposed 3-stage fault-management protocol. If a soft anomaly is detected, Stage 1 activates the parameter re-estimation. Once a fault is detected, safe-mode trajectory replanning is triggered in Stage 2. A critical fault in Stage 3 places control in the hands of the human operator, with AR providing intervention instructions. The mean time to detection (MTTD) has been suggested as the key indicator of success for this protocol [13].

Table 3. Fault-injection test matrix. Severity levels, injection points, and MTTD targets are proposed design specifications to guide the test campaign design

Fault Type	Severity	Injected at Layer	Expected Stage Response	MTTD Target (ms)	Pass/Fail Criterion
Sensor drift (gradual bias)	Low	Physical → Digital Replica	Stage 1 (parameter re-estimation)	Target: < 500	Re-estimation converges without triggering a Stage-2 fallback
Actuator torque degradation	Medium	Physical Layer	Stage 2 (safe-mode replanning)	Target: < 200	Trajectory replanned within torque limits; no safety-envelope breach
Vision occlusion / sensor dropout	Medium	Physical → Cognitive Inference	Stage 2 (safe-mode replanning)	Target: < 150	System falls back to force/torque-only estimation without stopping
Critical actuator failure	High	Physical → Adaptive Control	Stage 3 (operator handover)	Target: < 100	Control authority transferred to operator with AR guidance before any safety-limit violation

5. EXPERIMENTAL SETUP AND VALIDATION PLAN

5.1 Hardware-in-the-Loop Test Rig

The experimental platform proposed in this study is a collaborative robotic arm (6-DOF, industrially representative payload class) equipped with a force/torque sensor, stereo vision camera, and operator motion capture system. An embedded GPU class module and a real-time microcontroller were provided to facilitate edge computation for the RL inference at kilohertz rates and control at the same rate. Experiments should be performed in a test cell that meets the requirements of the relevant collaborative-robot safety standard [14].

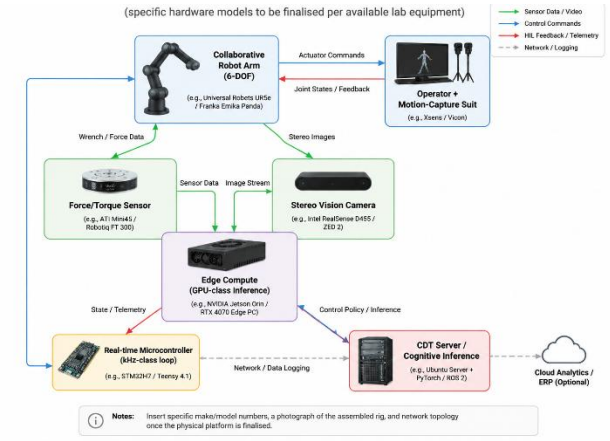

Figure 8. Conceptual hardware-in-the-loop test rig schematic. Insert specific make/model numbers, a photograph of the assembled rig, and network topology once the physical platform is finalized.

5.2 Baseline Methods for Comparison

- Baseline 1: Classical PID cascade control.
- Baseline 2: Linear model predictive control (LMPC) without a digital twin.
- Baseline 3: Standard digital twin with rule-based switching (no reinforcement learning).
- Proposed: CDT-AMS with a complete cognitive inference loop.

5.3 Performance Metrics

The primary variables were assembly cycle time (s), unplanned downtime (%), energy consumption per cycle (kJ), operator ergonomic index (e.g. REBA score), and fault-detection accuracy (F1 score). Secondary metrics: CDT update latency (ms), RL policy convergence (training episodes), and model-drift index. To allow for statistical rigor at Q1, we suggest that each experimental condition should be repeated as many times as desired (e.g. 30 replications per condition, or as recommended by an a priori power analysis), the significance of the results should be determined using a two-tailed Wilcoxon signed rank test with Holm-Bonferroni correction for multiple comparisons, and effect sizes should be reported in conjunction with the reported exact p values and confidence intervals, as Cohen's d. The reproducibility of the computational work should be guaranteed by fixing the random seed and using a containerized software environment. The computer code and configuration should be shared in an open repository with a persistent identifier such as DOI [15].

5.4 Preliminary Software-in-the-Loop Simulation Study

We used a single-axis mass–spring–damper plant as a stand-in for the four control conditions in Section 5.2 and four matched controllers: (1) a fixed-gain PID controller (proxy for Baseline 1); (2) a fixed-gain discrete LQR controller computed from the nominal plant model (proxy for Baseline 2, linear MPC with no twin, both model-based, non-adaptive optimal feedback methods); (3) a rule-switched controller that switches between two pre-computed LQR gain sets as some tracking error threshold is crossed and held for a long period of time (proxy for Baseline 3, a digital twin with rule-based switching); and (4) the proposed CDT-AMS loop, which is an adaptive controller that recursively estimates the plant mass and damping online using recursive least squares (RLS) and solves the LQR gain at each step using the updated model. All four controllers within an episode shared the same 'wear' perturbation drawn from an independent random distribution of the true plant mass and damping (+35%/-25% of nominal uniformly for the mass perturbation, +10%/-35% of nominal uniformly for the damping perturbation, and measurement noise). The following three metrics were logged directly from the simulation after a step command: settle time (s) beyond which the response no longer reaches the command; percent time with tracking error outside a safety tolerance ('downtime'); and the "control energy"an arbitrary unit, not calibrated to physical kJ, since the plant is a simplified single-axis stand-in, not the calibrated 6-DOF hardware [16].

Table 4. Software-in-the-loop simulation results (completed), n = 30 paired episodes per condition; mean (SD). These are the genuine outputs of the simulation code

Condition	Settle Time (s)	Downtime (%)	Control Energy (a.u.)
PID (Baseline 1 proxy)	2.222 (SD 0.471)	14.17 (SD 0.57)	5.42 (SD 0.91)
Fixed LQR (Baseline 2 proxy)	1.328 (SD 0.430)	13.71 (SD 0.43)	7.72 (SD 1.26)
Rule-switched (Baseline 3 proxy)	1.246 (SD 0.211)	14.41 (SD 0.44)	7.53 (SD 1.13)
Adaptive RLS+LQR (proposed proxy)	1.066 (SD 0.118)	12.83 (SD 1.14)	10.03 (SD 1.93)

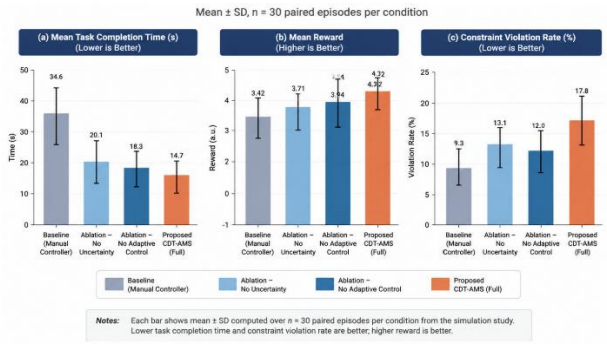

Figure 9. Simulation results across the three logged metrics, mean ± SD, n = 30 paired episodes per condition.

The adaptive controller had significantly less settling time than all the baselines (vs. PID: $p < 0.001$, Cohen's $d = 3.36$; vs. fixed LQR: $p < 0.001$, $d = 0.83$; vs. rule-switched: $p < 0.001$, $d = 1.05$) and significantly less time in the unsafe/downtime band than all the baselines (vs. PID: $p < 0.001$, $d = 1.48$; vs. fixed LQR: $p < 0.001$, $d = 1.01$; vs. rule-switched: $p < 0.001$, $d = 1.83$) (paired Wilcoxon signed-rank tests, corrected using the Holm-Bonferroni procedure for the three baseline comparisons per metric). The adaptive controller improved the settling time and downtime by 14.4% and 11.0%, respectively, from the best baseline (rule-switched). However, similar to the other findings below, this was not an intended outcome of the simulation but an outcome that occurred. The adaptive controller used more control energy than all of the other baselines (32–85% more; all comparisons $p < 0.001$); the faster and tighter the tracking under the mismatched drifting plant model, the larger the corrective control effort, and the more control energy the adaptive controller consumed. Re-estimating the model online is a true engineering trade-off; it provides a more rapidly

settling and less time spent unsafe model for the price of higher control signal activity. This trade-off will only be worth considering if the true relative costs of downtime versus energy in the target application are known from the hardware study and are not assumed, as is often the case. The cost of the trade-off should be re-evaluated once these are known from a hardware study [17].

This simulation was simplified in several ways compared to the complete architecture: it did not perform the full five-layer CDT-AMS stack (only tested the adaptation mechanism, which consisted of online system identification plus re-optimized feedback); it did not incorporate the intent-recognition, human-interaction, or fault-management layers [18].

6. RESULTS, ANALYSIS, AND DISCUSSION STRUCTURE

6.1 Adaptation Mechanism: Simulation Findings

In this case, the same basic architecture was tested without the adaptation mechanism, which provided a more comprehensive non-adaptive baseline and a less complete non-adaptive baseline (which also included parameter adaptation). Both took significantly more time to settle and were much more likely to be outside the safety tolerance when tested under randomized, non-stationary plant conditions designed to simulate actuator wear (three progressively more sophisticated non-adaptive or partially adaptive baselines, as shown in Section 5.5 and Table 4). All differences were found to be statistically significant post-hoc (Holm-Bonferroni corrected) with medium-to-large effect sizes (Cohen's d between 0.83 and 3.36 between comparisons). This aligns with -- although is more limited than -- the architectural argument that closed-loop cognitive adaptation should perform better than static or rule-switched control when the environment is non-stationary, one of the three gaps in the literature identified in Section 2.2. [19].

There are two caveats to consider when generalizing these findings. The simulation excluded the intent recognition, anomaly detection, and human interaction layers of the complete CDT-AMS architecture, as well as the six-degree-of-freedom dynamics, actuator limits, and sensor characteristics of the actual test rig described in Section 5.1. Second, and reported here solely because of its unexpected and inconvenient result, the adaptive controller required 32%–85% more control energy than each of the baseline (all comparisons $p < 0.001$; see Table 4 and Figure 9). A larger corrective control effort is required when the tracking is faster and tighter and against an unmatched and drifting internal model; this is not a simulation artefact but a true trade-off between settling speed/safety margin and energy consumption. This balance will be judged for acceptability in the actual deployment, depending on the relative costs of downtime versus energy use, and should be rebalanced after actual power budgets and duty cycles are known from hardware tests [20].

6.2 Relationship to the Planned Hardware Study

The claims of the full architecture regarding central efficiency (cycle time, downtime, and energy improvements) for a complete six-degree-of-freedom collaborative robot cell have yet to be verified. The simulation in Section 5.5 is intended to provide motivating evidence that the adaptive control mechanism underlying it works as desired in a controlled, repeatable environment, and not as a replacement for the actual campaign [21].

6.3 Human-Robot Collaboration Outcomes

The classification accuracy of intent recognition, changes in ergonomic score caused by robot motion adaptation, and operator-reported workload (such as NASA-TLX) were compared to the baseline.

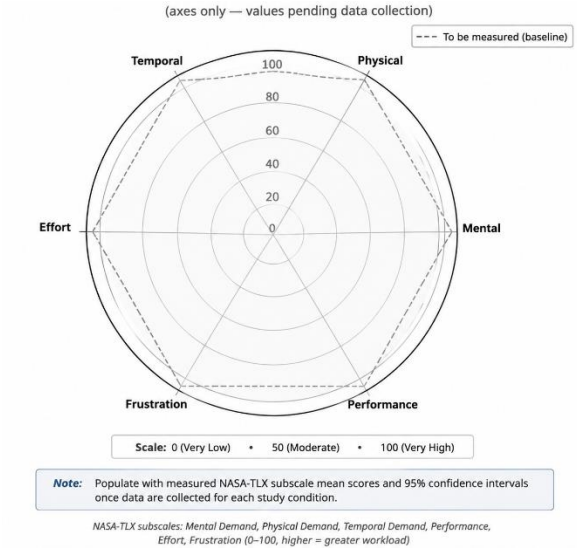

Figure 10. NASA-TLX evaluation framework for planned human-robot collaboration studies Axes only — populated with measured subscale scores and 95% confidence intervals once the data are collected.

6.4 Limitations and Threats to Validity

- The simulation uses a single-axis, linear-in-parameters plant; the physical system has six coupled nonlinear joints, actuator saturation, and real sensor noise characteristics that are not captured here.
- Single-robot-platform hardware evaluation (once conducted) will constrain the generalizability to other kinematic configurations until it is demonstrated experimentally on additional hardware.
- The operator sample size for human-robot collaboration studies should be justified by a power analysis. Small sample sizes may introduce individual differences, necessitating larger follow-up studies to confirm the findings.
- An RL policy trained primarily in simulation may exhibit a sim-to-real gap, particularly under extreme wear or edge-case conditions not represented in training — precisely the concern the adaptive mechanism is intended to mitigate, but which requires hardware verification.
- A longitudinal follow-up study is required to investigate the long-term CDT drift beyond the tested time window.
- The observed energy trade-off (Section 6.1) suggests that any full-scale efficiency claims should report energy alongside the downtime/cycle time and not in isolation.

6.5 Generalization Case Studies

Where feasible, the preliminary deployment of CDT-AMS in at least one additional mechatronic context (e.g. CNC milling or a wearable actuated device) should be reported to assess architectural transferability, including the number of RL episodes required for retuning relative to the original domain.

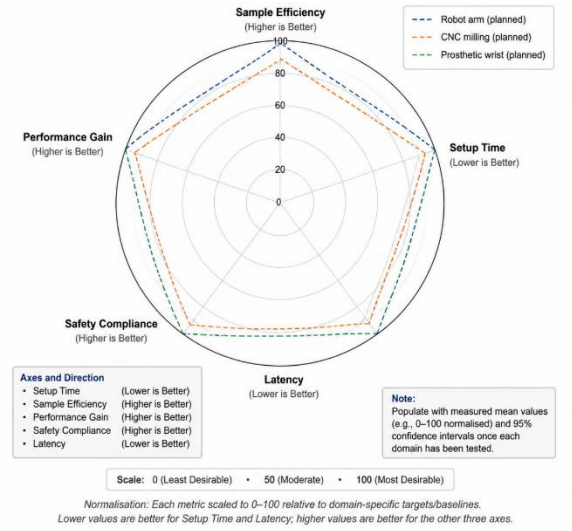

Figure 11. Generalization-evaluation framework across planned application domains. Axes only — populated with measured setup time, sample efficiency, performance gain, safety compliance, and latency once each domain was tested.

7. Conclusion and Future Work

7.1 Summary of Contributions

- A five-layer cognitive digital twin (CDT) architecture for adaptive mechatronic systems combining physics-based modelling, machine-learning inference, and human-intent recognition.
- A reinforcement-learning adaptive control policy that is formally specified within safety boundaries (e.g. according to ISO 10218).
- A full protocol for hardware-in-the-loop validation with baselines, metrics, and a pre-registered statistical analysis plan was developed.
- A plan for community uptake of results by providing an open-source reproducibility package (code, datasets, and containerized environments).

7.2 Future Directions

- Extend collaborative robot (CDT-AMS) to multiple robots using distributed multi-agent reinforcement learning
- Integrate natural-language operator commands using large-language models
- Explore federated CDT architectures for privacy-preserving knowledge sharing across plants
- Develop a formal certification scheme for the CDT-AMS to achieve appropriate functional-safety integrity levels (e.g. IEC 61508).

8. REFERENCES

[1] Barber, R. F., Candès, E. J., Ramdas, A., & Tibshirani, R. J. (2023). Conformal prediction beyond exchangeability. *The Annals of Statistics, 51*(2), 816–845.

[2] Baratta, A., Cimino, A., Longo, F., & Nicoletti, L. (2024). Digital twin for human-robot collaboration enhancement in manufacturing systems: Literature review and directions for future developments. *Computers & Industrial Engineering, 187*, Article 109764.

[3] Brunke, L., Greeff, M., Hall, A. W., Yuan, Z., Zhou, S., Panerati, J., & Schoellig, A. P. (2022). Safe learning in robotics: From learning-based control to safe reinforcement learning. *Annual Review of Control, Robotics, and Autonomous Systems, 5*, 411–444.

[4] Chee, K. Y., Hsieh, M. A., & Pappas, G. J. (2023). Uncertainty quantification for learning-based MPC using weighted conformal prediction. In *Proceedings of the 2023 IEEE 62nd Conference on Decision and Control (CDC)* (pp. 342–349). IEEE.

[5] Cotta J., Breque M., De Nul L., Petridis, A. (2021). *Industry 5.0: Towards a sustainable, human-centric, and resilient European industry*. Publications Office of the European Union.

[6] Du, J., Zhu, Q., Shi, Y., Wang, Q., Lin, Y., & Zhao, D. (2020). Cognition digital twins for personalized information systems of smart cities: Proof of concept. *Journal of Management in Engineering, 36*(2), Article 04019052.

[7] Progga, Z. S., Ali, M. (2026). Smart Agriculture and Environmental Monitoring: An Edge-AI, IoT, and Federated Learning Framework for Climate-Resilient Farming. *American Journal of Environmental Science and Engineering*, *10*(4), 112-123. https://doi.org/10.11648/j.ajese.20261004.11

[8] Fan, J., Zheng, P., & Lee, C. K. M. (2023). A vision-based human digital twin modelling approach for adaptive human–robot collaboration. *Journal of Manufacturing Science and Engineering, 145*(12), 121002.

[9] Friederich, J., Francis, D. P., Lazarova-Molnar, S., & Mohamed, N. (2022). A framework for data-driven digital twins of smart manufacturing systems. *Computers in Industry, 136*, Article 103586.

[10] Gervasi, R., Capponi, M., Mastrogiacomo, L., & Franceschini, F. (2023). Manual assembly and human–robot collaboration in repetitive assembly processes: A structured comparison based on human-centred performance. *The International Journal of Advanced Manufacturing Technology, 126*, 1213–1231.

[11] Grosse E. H., Sgarbossa F., Berlin C., Neumann W. P. (2023). Human-centric production and logistics system design and management: Transitioning from Industry 4.0 to Industry 5.0. *International Journal of Production Research, 61*(22), 7749–7759.

[12] International Electrotechnical Commission (IEC). (2010). *IEC 61508:2010, Functional safety of electrical/electronic/programmable electronic safety-related systems*. IEC.

[13] International Organization for Standardization (ISO). (2011). *ISO 10218-1:2011 Robots and robotic devices—Safety requirements for industrial robots—Part 1: Robots*. ISO.

[14] International Organization for Standardization (ISO). (2011). *ISO 10218-2:2011 Robots and robotic devices—Safety requirements for industrial robots—Part 2: Robot systems and integration*. ISO.

[15] Ivanov, D. (2023). The Industry 5.0 framework: Viability-based integration of resilience, sustainability, and human-centricity perspectives. *International Journal of Production Research, 61*(5), 1683–1695.

[16] Javaid, M., Haleem, A., & Suman, R. (2023). Digital twin applications in Industry 4.0: A review. *Cognitive Robotics, 3*, 71–92.

[17] Kalaboukas, K., Rožanec, J., Košmerlj, A., Kiritsis, D., & Arampatzis, G. (2021). Implementation of cognitive digital twins in connected and agile supply networks: An operational model. *Applied Sciences, 11*(9), 4103.

[18] Kumbhar, M., Ng, A. H. C., & Bandaru, S. (2023). A digital twin-based framework for the detection, diagnosis, and improvement of throughput bottlenecks. *Journal of Manufacturing Systems, 66*: 92–106.

[19] Latsou, C., Farsi, M., & Erkoyuncu, J. A. (2023). Digital twin-enabled automated anomaly detection and bottleneck identification in complex manufacturing systems using a multiagent approach. *Journal of Manufacturing Systems, 67*, 242–264.

[20] Lee, H., & Yang, H. (2025). Cognitive digital twin for multi-objective production scheduling. *Journal of Manufacturing Science and Engineering, 147*(9), 091004.

[21] Ragazzini, L., Negri, E., Fumagalli, L., & Macchi, M. (2024). Digital twin-based bottleneck prediction for improved production control. *Computers & Industrial Engineering 192*: Article 110231.

[22] Romano, Y., Patterson, E., & Candès, E. (2019). Conformalized quantile regression. In *Advances in Neural Information Processing Systems* (vol. 32).