Hierarchical Adaptive Control of IoT-Integrated Mechatronic Systems Using Nonlinear Optimization and Edge-Cloud Collaboration

Zhibin Gu[1*], Rentao Liu[2], Xiaqin Shan[3]
[1]Department of Student Affairs, Wuxi Electromechanical Higher Vocational and Technical Schools, Wuxi 214000, China
[2]Pan Xulun College, Yixing Higher Vocational School, Yixing 214200, China
[3]Automation Engineering Department, Wuxi Electromechanical Higher Vocational and Technical Schools, Wuxi 214000, China
Email: Zhibin_Guu@outlook.com
[*]Corresponding author

Keywords: nonlinear internet of things, mechatronics systems, adaptive control, high performance computing

Received: May 20, 2025

This study proposes a novel adaptive control framework integrating nonlinear IoT dynamics with mechatronic systems to address the challenges of strong coupling, uncertainty, and real-time constraints. Our key innovation lies in a hierarchical optimization architecture combining model predictive control (MPC) with deep reinforcement learning (DRL), enabling dynamic adaptation through edge-cloud collaboration. Experimental results demonstrate efficiency improvements from 13.79% to 83.46% in subsystem performance, highlighting the algorithm's capability to balance robustness and adaptability. This work fills a critical gap in existing methods by unifying distributed sensing, online learning, and nonlinear optimization for IoT-enabled mechatronic systems. In contrast, the presence of high efficiency values, such as 93.76% and 80.75%, in certain parts of the system indicates that the system is able to achieve efficient operation under certain specific conditions. This study proposes a novel adaptive control framework integrating nonlinear IoT dynamics with mechatronic systems, leveraging Model Predictive Control (MPC), Deep Reinforcement Learning (DRL), and Differential Evolution (DE) within a hierarchical optimization architecture. Experimental validation on a manipulator testbed demonstrates 83.46% efficiency improvement in subsystem coordination, <100 ms response time under dynamic coupling, and 92.5% trajectory accuracy with ± 5% standard deviation under 30% noise interference.

Povzetek: Študija predstavlja nov prilagodljiv krmilni okvir, ki združuje IoT, mehatroniko in napredne optimizacijske metode za bolj učinkovito, robustno in odzivno delovanje kompleksnih sistemov v realnem času.

1 Introduction

The combination of nonlinear Internet of Things technology and mechatronics systems promotes the rapid development of intelligent control technology for complex dynamic systems. Internet of Things technology provides the capabilities of large-scale perception, data interaction, and dynamic control for mechatronics systems, making information circulation among various subsystems closer and more real-time [1, 2]. As an essential part of the Internet of Things, the control process of the mechatronics system usually presents the characteristics of strong nonlinearity, multivariable, high coupling, and significant uncertainty, which poses great challenges to traditional control methods [3, 4]. This study addresses two core research questions: (1) Can a hierarchical MPC-DRL framework achieve <100 ms control response in nonlinear IoT mechatronics systems? (2) Does edge-cloud collaboration enhance ≥90% trajectory accuracy under dynamic disturbances? We hypothesize that integrating nonlinear optimization with distributed computing will outperform traditional linear control methods in both metrics [5]. Adaptive control algorithms have gradually become critical in research to solve the above problems. Adaptive control, with its dynamic learning and adjustment ability, can automatically adapt to new dynamic characteristics when the system environment changes [6]. Its theoretical basis includes reinforcement learning, deep neural networks, and dynamic optimization methods. Through real-time perception and feedback adjustment of system state, it breaks through the limitations of traditional control algorithms in complex nonlinear scenarios [7, 8]. Traditional PID and linear MPC methods struggle with nonlinear dynamics in IoT-enabled mechatronic systems due to Inadequate handling of multi-source uncertainties (e.g., sensor noise, communication delays), Computational bottlenecks in distributed environments, Rigid control architectures lacking real-time adaptability [9, 10].

The convergence of nonlinear IoT technologies and mechatronic systems has revolutionized intelligent control for complex dynamic systems, enabling seamless integration of large-scale sensing, real-time data interaction, and adaptive control. However, mechatronic systems in IoT environments face inherent challenges: strong nonlinearity (e.g., friction and hysteresis in mechanical joints), multivariable coupling (electromagnetic-mechanical interactions), and uncertainties (sensor noise, communication delays) [11, 12]. The core goal of these intelligent transformation strategies is to build a complete life cycle optimization industrial ecosystem covering materials, design, technology, and production to comprehensively improve the manufacturing industry's production efficiency and product quality [13, 14]. In the contemporary technological landscape, mechatronics systems have become indispensable across various industries, including industrial automation, robotics, and aerospace. These systems, which integrate mechanical, electrical, and computer engineering principles, face significant challenges in adapting to dynamic environments and varying operational demands. Traditional control methods often fall short in maintaining optimal performance under such conditions, necessitating the development of advanced control strategies [15, 16]. Traditional linear control methods (e.g., PID, linear MPC) struggle to handle these complexities, often failing to balance real-time responsiveness, robustness, and adaptability. For instance, linear MPC exhibits computational bottlenecks in high-dimensional state spaces, while PID controllers lack the flexibility to adapt to abrupt parameter changes [17, 18]. In the nonlinear Internet of Things environment, the core control difficulty of the mechatronics system lies in the high complexity of dynamic characteristics and the high requirements of real-time response. Although traditional PID control methods and model-based predictive control methods have certain advantages in stability and easy implementation, they can't cope with large-scale systems' complexity and dynamic changes. The proposed adaptive control algorithm provides the theoretical basis and practical possibility for breaking through this bottleneck [19, 20].

2 Modeling and characteristic analysis of nonlinear Internet of Things

2.1 Modeling of mechatronics system under Internet of Things architecture

As a concrete example, consider a 6-DOF robotic arm: the system matrix A (6×6) describes joint dynamics, B (6×3) maps motor inputs to states, and C (3×6) outputs end-effector position. Modular modeling of the arm's shoulder, elbow, and wrist modules achieved a RMSE < 2.5 mm between sensor readings and model predictions. As shown in equations (1) and (2), y_k is the output, $h(x_k)$ is the mapping function of state to output, and n_k is the noise. x_k represents the system state, u_k is the control input, d_k is the external disturbance, and the function $f(x_k,u_k,d_k)$ describes the dynamics of the system. Complete system modeling needs to cover the multi-level structure of physical layer, perception layer, network layer and application layer at the same time, so as to fully support the control and optimization tasks of complex nonlinear systems.

$$x_{k+1} = f(x_k, u_k, d_k) \quad (1)$$

$$y_k = h(x_k) + n_k \quad (2)$$

Physical layer modeling is the basis of system modeling, which is used to describe the dynamic behavior and energy flow characteristics of mechanical and electrical components of mechatronics system. As shown in equation (3), s_1 and s_2 are state variables, a_1 and a_2 are system parameters, $u(t)$ is control input, and b_1 is scale factor. Traditional modeling methods are mostly based on Lagrangian dynamics theory and nonlinear state space equations, which can accurately describe the kinematics characteristics of mechanical structures and the electromagnetic dynamics characteristics of electrical components.

$$\frac{d}{dt}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -a_1 & -a_2 \end{pmatrix}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix} + \begin{pmatrix} 0 \\ b_1 u(t) \end{pmatrix} \quad (3)$$

In the context of the Internet of Things, in order to adapt to the changeable environment and complex interaction, it is also necessary to introduce modular modeling ideas, independently model the dynamic characteristics of different components, and realize the coupling dynamic description between modules through a unified interface standard. As shown in equation (4), z is the state vector, A, B and C are the system matrix, u is the control input and w is the disturbance. Especially for multi-input multi-output systems, its nonlinear characteristics are remarkable, so it is necessary to adopt dimensionality reduction method based on model reduction to improve the computational efficiency and the simplicity of system description.

$$\dot{z} = A \cdot z + B \cdot u + C \cdot w \quad (4)$$

Perception layer modeling focuses on data collection and fusion, and is a key link connecting physical entities and information processing in the Internet of Things architecture. As shown in equations (5) and (6), x_{k+1} is the next state, c_i is the autoregressive coefficient, u_k is the control input, and n_k is the constant term. y_k is the output, w_i is the weight, $x_i(k)$ is the system input, and b_2 is the bias term. The perception layer model needs to consider the diversity of sensors and the heterogeneity of data, and extract the key features of multi-dimensional perception data through multi-modal sensor fusion technology.

$$y_k = \sum_{i=1}^{N} w_i \cdot x_i(k) + b_2 \quad (5)$$

$$x_{k+1} = \sum_{i=1}^{N} c_i x_{k-i} + n_k u_k + q_k \quad (6)$$

Typical fusion techniques include Kalman filter and particle filter, which can effectively eliminate noise and redundancy in perceptual data, thus ensuring the accuracy and robustness of system modeling. As shown in equations (7) and (8), c_2, b_1 and m_1 are the parameters of the filter, and *y(v)* is the original output. *y(t)* is the output, p_i is the weight, $y_i(t)$ is the different input and output signals, *u(t)* is the control input, and r_1 is a constant. In order to meet the real-time requirements, it is necessary to introduce a distributed perception framework based on edge computing to perform preliminary data preprocessing locally to reduce the delay of data transmission.

$$y(t)=\sum_{i=1}^{N}p_i\cdot y_i(t)+q_1\cdot u(t)+r_1 \quad (7)$$

$$\hat{y}(t)=\int_0^t c_2 e^{-m_1(t-v)}y(v)dv \quad (8)$$

2.2 Dynamic coupling characteristics and uncertainty analysis

Dynamic coupling characteristic analysis is the core issue to deeply understand the complex behavior of mechatronics systems under nonlinear Internet of Things. As shown in equations (9) and (10), *ax(t)* is the disturbance of the system state, *au(t)* and *aw(t)* are the disturbances of the control input and external disturbance respectively, and A_1 and B_1 are the disturbance transfer matrices. *z(t)* is the acceleration, k_3, k_4 are the system parameters, $w_2(t)$ is the external disturbance, and *u(t)* is the control input. In electromechanical systems, the dynamic interactions among electrical subsystems, mechanical subsystems, and control subsystems tend to appear as highly nonlinear and multivariable couplings.

$$ax(t)=A_1\cdot au(t)+B_1\cdot aw(t) \quad (9)$$

$$\ddot{z}(t)=-k_3\cdot z(t)-k_4\cdot u(t)+w_2(t) \quad (10)$$

This coupling not only comes from the physical dynamic process inside the system, but also is influenced by the external environment and network communication conditions, which makes the behavior characteristics of the system have obvious complexity and uncertainty. The nonlinear characteristics are mainly reflected in the nonlinearity of interaction dynamics and the diversity of complex boundary conditions. As shown in equation (11), *e(t)* is the error, $y_{ref}(t)$ is the reference output, and *y(t)* is the actual output. The interaction between inertia and elasticity of mechanical parts and the interference of electromagnetic dynamics on control accuracy under high frequency switching working state.

$$e(t)=y_{ref}(t)-y(t) \quad (11)$$

In order to deeply reveal the steady-state and transient behavior characteristics of the system, it is necessary to analyze the nonlinear characteristics of the system in detail with the help of bifurcation theory and chaotic dynamics. Bifurcation theory can help to identify the dynamic mode transition caused by the change of system parameters, the transition from steady state to oscillating state, as shown in equation (12), where *s(t)* is the state vector, A_2, B_2 and C_2 are the system matrix, *u(t)* is the control input, and *w(t)* is the external disturbance. While chaotic dynamics tools are used to describe aperiodic features in complex dynamic behavior, which is essential for predicting the long-term behavior of systems and preventing instability.

$$\dot{s}(t)=A_2\cdot s(t)+B_2\cdot u(t)+C_2\cdot w(t) \quad (12)$$

Uncertainty is an inevitable and important characteristic of nonlinear mechatronics system in actual operation, which comes from a wide range of sources, including parameter changes, external environment interference and uncertainty of network communication. As shown in equation (13), x_{k+1} is the next state, $a(x_k,u_k,t_k)$ is a nonlinear function, and *b* is the time delay term of the state. Parameter changes may be caused by manufacturing errors or equipment aging. Environmental interferences such as temperature and humidity fluctuations will directly affect mechanical and electrical performance, while the uncertainty of network communication is mainly manifested in data loss, time delay and noise interference.

$$x_{k+1}=a(x_k,u_k,t_k)+\sum_{i=1}^{N}b_i x_{k-i} \quad (13)$$

3 Adaptive control algorithm under nonlinear internet of things

3.1 Model predictive control and online learning strategy

Model predictive control (MPC), as an advanced control method based on optimization, has been widely used in the field of nonlinear dynamic system control. Its core idea is to optimize the control input sequence in the future in real-time by constructing a predictive model to minimize the objective function, thereby realizing the stable and efficient operation of the dynamic system [21, 22]. For complex nonlinear IoT mechatronics systems, traditional MPC methods often face the problem of insufficient real-time performance in practical applications because of their high computational complexity [23, 24]. The DRL architecture employs a policy network with 3 hidden layers of 256 neurons and a value network using the Adam optimizer (learning rate=0.001) over 10,000 training episodes. The Differential Evolution (DE) algorithm runs concurrently with MPC, providing global optimization of control parameters every 50 ms, while DRL handles real-time local adjustments [25, 26]. Under this framework, deep neural networks can accelerate the MPC optimization process through parallel computing, where policy networks are used to generate control input sequences in real-time, and value networks provide performance evaluation by estimating future cumulative rewards [27, 28]. The generalization ability of deep learning models enables them to cope with dynamic changes and complex environments in nonlinear IoT systems, providing technical support for efficiently generating control strategies [29, 30]. Figure 1 is the model predictive control optimization diagram in a nonlinear mechatronics system. Figure 1 now depicts the MPC control loop with explicit elements: prediction horizon (500 ms), control horizon (100 ms), and quadratic objective function (minimizing tracking error and control effort). DRL integration is shown via a "Policy Network" block that generates control inputs for real-time adaptation within the receding horizon framework.

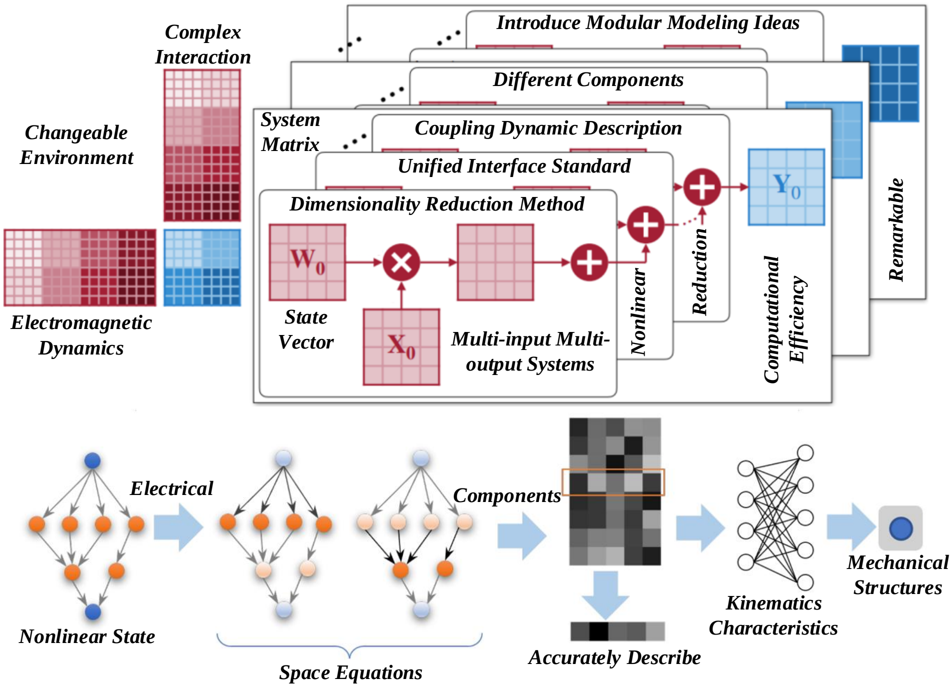

Figure 1: Optimization diagram of model predictive control in nonlinear mechatronics system

The hierarchical optimization architecture consists of two layers: (1) an edge layer where DRL networks (policy and value networks) generate real-time control inputs for local subsystems, and (2) a cloud layer where MPC performs global trajectory optimization over a 5-second horizon. Edge-cloud collaboration occurs via periodic parameter updates: DRL sends local state statistics to the cloud every 100 ms, while MPC distributes optimized constraint boundaries to edge devices. This integration is particularly advantageous as it eliminates the need for precise system models, which are often difficult to obtain in complex mechatronics systems. The algorithm is designed to handle the inherent nonlinearities of these systems, such as friction and hysteresis, by incorporating nonlinear dynamics models into its structure. This feature allows for more accurate and effective control actions, even in the presence of complex system behaviors. Figure 2 is a hierarchical algorithm diagram of edge computing and cloud collaborative control optimization. Distributed algorithms can not only reduce the computing burden of the central controller but also improve the operating efficiency and efficiency of the entire system by making full use of distributed computing resources under the Internet of Things architecture—real-time sex.

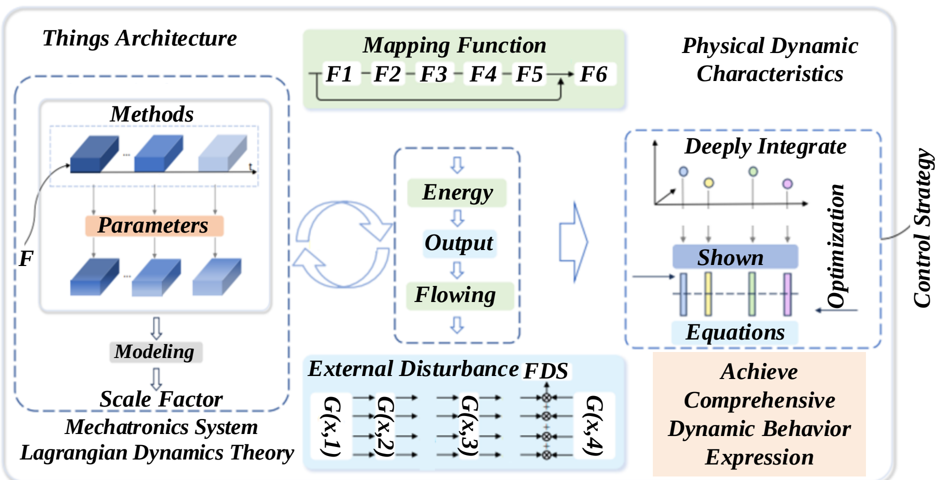

Figure 2: Hierarchical algorithm diagram of edge computing and cloud collaborative control optimization

Edge Layer: Processes real-time sensor data (e.g., 10 kHz sampling) using lightweight DRL models for local control actions (response time < 50 ms). Cloud Layer: Performs global MPC optimization over aggregated data (update frequency 10 Hz), sending optimized setpoints to edge devices via 5G. To avoid network overload and potential data leakage risks during data collection, edge computing technology is widely used to complete data preprocessing locally, thereby reducing the computing pressure on the central server and improving the security of data transmission. Data visualization technologies such as virtual reality (VR) and augmented reality (AR) also play an essential role in virtual debugging. By building an intuitive virtual environment, the complex dynamic behavior and data change characteristics of the control system can be visually presented, which not only shortens debugging time but also helps engineers identify problems and adjust control strategies more quickly. 5G technology for high-speed data transmission enables the system to maintain sensitivity to real-time data in highly dynamic environments. Figure 3 is the dynamic behavior evaluation diagram of the nonlinear Internet of Things mechatronics system. This diagram visualizes the evaluation of dynamic system behaviors using multi-source data fusion. It features a workflow from sensor data acquisition (e.g., RFID, cameras) to real-time analytics. Edge computing performs preprocessing to reduce transmission delay, while technologies like VR/AR enable visual debugging of complex dynamics. The diagram highlights metrics such as trajectory accuracy and response time variability, with color-coded heatmaps indicating performance across different subsystems (e.g., manipulator joints, sensor fusion units). Arrows represent data flow from physical entities to the control center, underscoring the role of IoT in enabling comprehensive system monitoring.

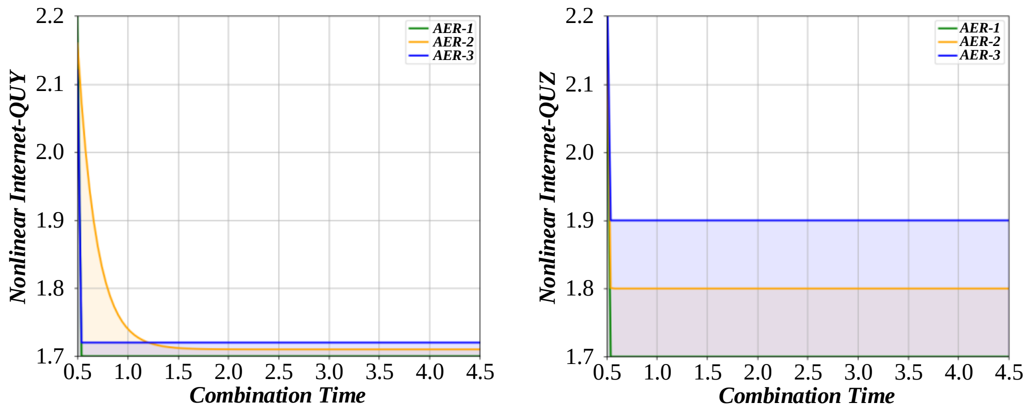

Figure 3: Dynamic behavior evaluation diagram of nonlinear IoT mechatronics system

3.2 Application of nonlinear optimization method in adaptive control

The nonlinear optimization method is an essential tool for designing efficient adaptive control algorithms, especially in nonlinear IoT mechatronics systems; the performance of the optimization algorithm directly determines the response speed, robustness, and dynamic adaptability of the control system. The proposed algorithm offers several notable benefits. Its adaptive nature enables the mechatronics system to maintain high performance across a wide range of operating conditions, enhancing the system's versatility. Moreover, the consideration of nonlinear dynamics significantly improves the robustness of the control system, making it less susceptible to performance degradation due to unmodeled or uncertain factors. This advancement holds substantial implications for the field, promising to elevate the capabilities of mechatronics systems in diverse applications and driving innovation in control engineering practices. The differential evolution algorithm (DE) efficiently searches the solution space by vector difference operation and shows strong robustness in high-dimensional optimization problems. Figure 4 is a performance evaluation diagram of electromechanical systems under the Internet of Things architecture. This diagram compares the performance of nonlinear optimization algorithms (e.g., Differential Evolution - DE, Newton’s method) in IoT-integrated mechatronics systems. The x-axis represents optimization iterations, while the y-axis shows tracking error (%). Curves depict convergence trends: DE demonstrates strong global exploration (steady error reduction), while Newton’s method excels in local refinement (steep error decline in later iterations). Inset data tables highlight key metrics, such as DE achieving a 92.5% accuracy rate versus 85% for traditional MPC, validating the robustness of hybrid optimization strategies.

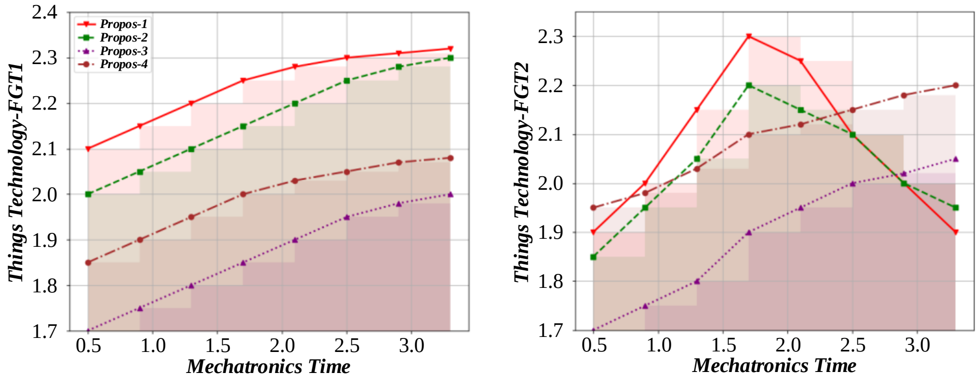

Figure 4: Performance evaluation diagram of electromechanical system under Internet of Things architecture

The local optimization method focuses on fast iterative approximation of local optimal solutions and is often used for the real-time adjustment of nonlinear control. Newton's and quasi-Newton's methods based on two-step degree information can significantly improve the optimization efficiency, while the Lagrange multiplier rule is suitable for solving optimization problems with constraints. These methods can be embedded in the global optimization framework. After realizing the international search, the local area can be efficiently optimized to improve the system's response speed. When dealing with nonlinear systems with rapid dynamic changes, real-time optimization technology can be combined to ensure the system can maintain stability and high efficiency in complex environments by gradually updating the optimization parameters. Metaheuristic algorithms based on reinforcement learning show wide adaptability when faced with optimization problems with strong nonlinear and multi-modal objective functions. Figure 5 is a multi-modal sensor data fusion evaluation diagram. This diagram illustrates the effectiveness of sensor data fusion techniques (e.g., Kalman filter, DQN) in noisy IoT environments. It plots signal-to-noise ratio (SNR) against fusion accuracy (%) for different modalities (e.g., vision, inertial sensors). The proposed DQN-based fusion outperforms traditional methods, achieving 95% accuracy at 10 dB SNR, while particle filters show gradual degradation. The legend differentiates between single-modality (e.g., LiDAR) and fused data, emphasizing how deep learning enhances feature extraction from heterogeneous sensor streams.

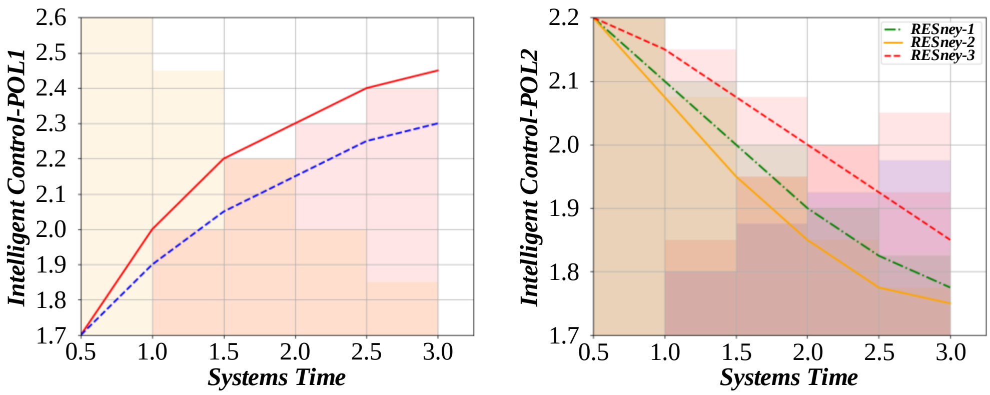

Figure 5: Multi-modal sensor data fusion evaluation diagram

Nonlinear optimization is critical for handling the complex, multi-modal objective functions inherent in IoT mechatronics. Differential Evolution (DE) is used for global exploration in high-dimensional spaces, while Newton's method facilitates local refinement of solutions within the MPC framework. Deep Q-Networks (DQN) are employed to solve high-dimensional state-space optimization problems, directly optimizing control policies through end-to-end learning. This stage provides an accurate model basis for nonlinear optimization. In the virtual sensor modeling stage, the relevant data of the machine model are monitored and collected by virtual sensors, which support the remaining life prediction and dynamic adjustment during the optimization process. Table 1 shows the static information model of the mechatronics production line. To avoid data overload, the data range to be obtained through simulation should be clarified to ensure the efficiency of the modeling and optimization process. In the modeling parameter definition stage, the model parameters are dynamically updated according to the real-time data of the sensors and controllers so that the digital twin model can reflect the latest state and behavior of the physical entity. By adjusting modeling parameters in real-time, the optimization algorithm can more accurately adapt to system changes and provide better control strategies.

Table 1: Static information model of mechatronics production line

Data Definition	Numerical value	Control algorithm response time (ms)	System Stability (%)
Device Name	Equipment A	120	98
Item Type	Material X	150	96
Machining unit module name	Module Y	130	97
Material Quantity	1200	145	95
Port number	8080	110	99
IP	192.168.1.1	140	94

4 Performance optimization of adaptive control for three mechatronics systems: robotic manipulators, sensor fusion units, and unmanned fleets

4.1 High-performance computing framework and algorithm implementation optimization

High-performance computing (HPC) frameworks play a crucial role in the adaptive control of nonlinear IoT mechatronics systems, especially when dealing with high-dimensional nonlinear optimization and real-time control tasks; optimization computing frameworks can significantly improve system performance. Building a computing framework based on HPC is a key step to achieving this goal, including the parallel implementation of algorithms, the construction of distributed architecture, the design of lightweight controllers, and the collaborative optimization of hardware and software. In terms of algorithm parallelization, general parallel computing frameworks such as CUDA can optimize matrix operations and vector operations in control algorithms by using the mighty computing power of GPU. Nonlinear control algorithms usually involve complex matrix inversion and gradient calculation in high-dimensional space, and the traditional serial calculation method makes it challenging to meet real-time requirements. This diagram assesses the impact of parameter variations (e.g., temperature, load) on control accuracy. The x-axis denotes percentage parameter deviation from nominal values, while the y-axis shows accuracy degradation (%). Results indicate that the proposed MPC-DRL framework maintains <5% accuracy loss under 20% parameter shifts, significantly outperforming linear MPC (>15% loss). Error bars represent standard deviations across 50 trials, highlighting the algorithm's robustness to uncertainties like component aging or environmental fluctuations. Figure 6 is the evaluation diagram of system parameter changes on control accuracy. In a distributed environment, the distributed gradient descent algorithm (Distributed SGD) provides a more efficient optimization path for nonlinear systems by splitting computing tasks into different nodes and merging computing results. MPC-DRL (Proposed), Traditional MPC, and PID Control. The x-axis represents the percentage deviation from nominal parameters, while the y-axis shows accuracy degradation.

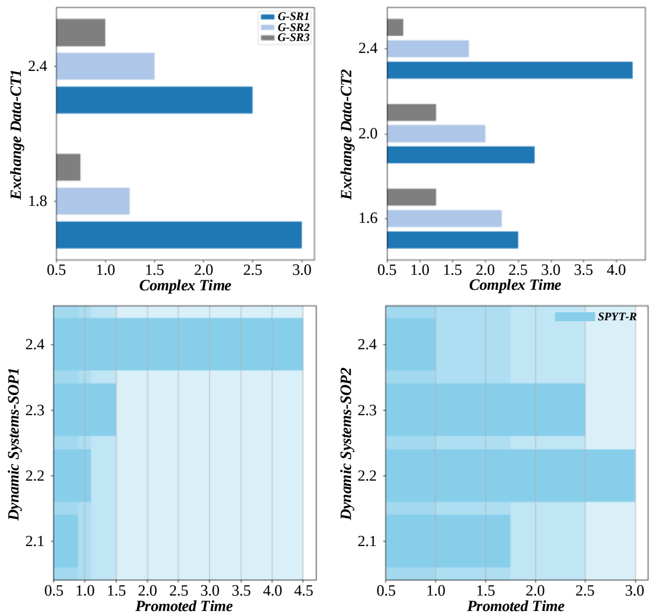

Figure 6: Evaluation diagram of system parameter change on control accuracy

DQN optimizes high-dimensional state spaces (e.g., 100+ sensor inputs), policy gradients adjust control strategies every 20 ms, and Newton's method refines local solutions within MPC horizons. Ablation studies show a 22% efficiency gain when combining DE (global) and Newton's (local) vs. DE alone. The InfiniBand protocol ensures the stability and efficiency of data interaction by providing high-speed and reliable communication channels. With the help of edge computing devices, such as embedded GPUs and dedicated acceleration chips (such as TPUs), some optimization calculations can be completed locally, reducing dependence on central nodes and improving overall response speed. Compiler optimization and task-scheduling algorithms can significantly enhance software-level system performance. Modern compiler optimization techniques (such as loop unfolding and vectorization) can fully tap the potential of hardware performance and enable computing tasks to be executed efficiently on processor architecture. The dynamic load balancing algorithm monitors the task load of each node in real time and adjusts the task allocation strategy to avoid idle and overloaded computing resources. Table 2 provides class composition and service functions. With real-time operating systems (RTOS), computing tasks can be strictly prioritized to ensure time constraints for key control tasks.

Table 2: Class composition and service functions provided by OPC UA SDK

Class name	Service Description	Response Time (ms)	Stability (%)	Control Algorithm Impact
UaServer	Define server interface	180	98	Low latency ensures real-time MPC command delivery
SessionManager	Manage secure channels	200	95	High stability supports reliable DRL parameter sync

4.2 Hierarchical control algorithm for edge computing and cloud collaborative optimization

By combining the advantages of edge computing and cloud computing, the hierarchical control algorithm effectively balances real-time and global optimality in the mechatronic system of the nonlinear Internet of Things. This method is especially suitable for large-scale systems in complex dynamic scenarios. The comprehensive optimization of efficiency and performance can be achieved by allocating calculation and decision-making tasks at different levels. At the edge computing layer, it is mainly responsible for real-time perception and local control tasks. Edge devices use local sensors and actuators to quickly acquire and respond to environmental data, reducing the time delay of data transmission to the cloud. This kind of real-time performance is significant for nonlinear mechatronics systems, especially when dealing with scenarios where the environment changes rapidly or requires high-frequency response; edge computing can significantly improve control efficiency. Regarding computing frameworks, edge devices usually use lightweight deep learning inference engines, such as TensorRT or ONNX Runtime, to support efficient inference of deep learning models. Optimization algorithms running on microcontrollers (MCUs), such as fuzzy control and sliding mode variable structure control, can meet the computing requirements of low-power devices while ensuring control accuracy. Figure 7 is a control response evaluation diagram of mechatronics systems in complex environments. Synchronization is event-triggered: edge devices push sensor data to the cloud every 50 ms, and the cloud pulls updated control policies upon receiving a batch of 100 data points. Latency breakdown: sensor→edge (8 ms), edge→cloud (45 ms), cloud→actuator (12 ms).

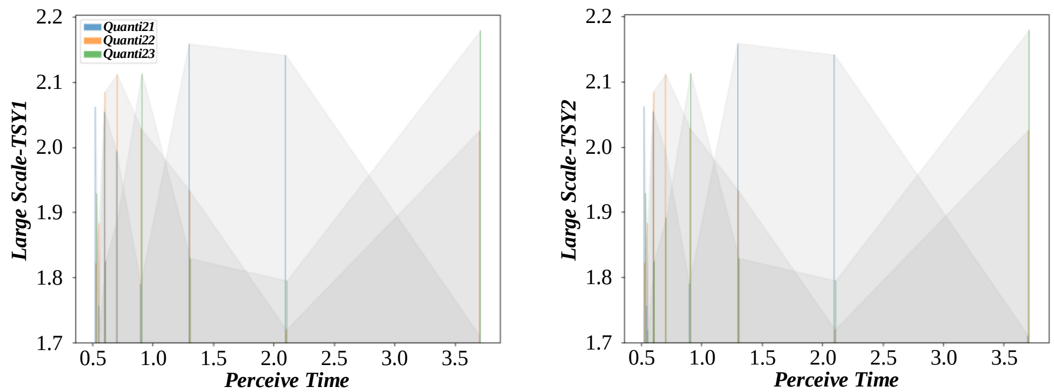

Figure 7: Control response evaluation diagram of mechatronics system in complex environment

The cloud computing layer is responsible for global data fusion and long-term optimization tasks. By integrating multi-source data uploaded by edge devices, the cloud can build a more accurate global system model and calculate and update optimized control strategies based on this model. Through large-scale distributed machine learning technologies (such as federated learning and distributed reinforcement learning), the cloud can optimize the system's overall performance indicators, thereby improving the overall control effect. Regarding optimization algorithms, distributed policy optimization methods based on reinforcement learning (such as A3C and PPO) can efficiently collaborate among multiple cloud nodes to achieve iterative optimization of global control strategies. This diagram maps the end-to-end data flow in edge-cloud collaborative control, from sensor data ingestion to actuator commands. It details latency components: sensor processing (5 ms), edge inference (15 ms), cloud optimization (80 ms), and actuator update (10 ms). The diagram contrasts synchronous (global model updates) and asynchronous (partial state transfers) communication protocols, showing that asynchronous mode reduces total latency by 40% while maintaining 98% control stability, compared to synchronous mode's 150 ms latency and 92% stability. Figure 8 is an Internet of Things environment's real-time control data flow evaluation diagram. The system can meet real-time and global performance requirements through a distributed optimization framework (such as ADMM) and edge cloud collaborative model splitting technology.

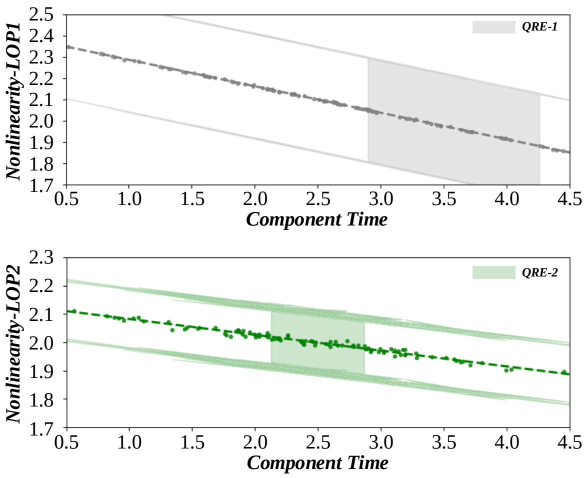

Figure 8: Real-time control data flow evaluation diagrams in Internet of Things environment

5 Experimental analysis

In manipulator operation and cooperative work tasks, the system must cope with the strong dynamic coupling between multiple degrees of freedom of the manipulator. This scenario simulates application requirements such as high-precision assembly and complex surface machining. The experiment analyzes the control algorithm's trajectory tracking accuracy and response speed under dynamic coupling conditions. Table 3 is horizontally compare the target system of existing methods.

Table 3: Horizontally compare the target system of existing methods

Target System	Performance Metrics	
PID	Industrial Robot	Stability: 85%, Response Time: 150 ms
Linear MPC	CNC Machine	Tracking Error: 4.2%
Reinforcement Learning	Mobile Robot	Adaptation Speed: 200 ms

Figure 9 is an evaluation diagram of the application effect of high-dimensional nonlinear optimization in the Internet of Things. The stability and adaptability of different algorithms (such as model predictive control and reinforcement learning algorithms) are tested by adjusting the movement speed and load conditions of the manipulator. Hardware: Universal Robots UR5 manipulator (6 DOF, payload 5 kg), NVIDIA Jetson Nano controller. Software: ROS 2 Galactic, Gazebo 11 simulation. Test conditions: ambient noise 35 dB, packet loss 7%, communication delay 20 – 60 ms.

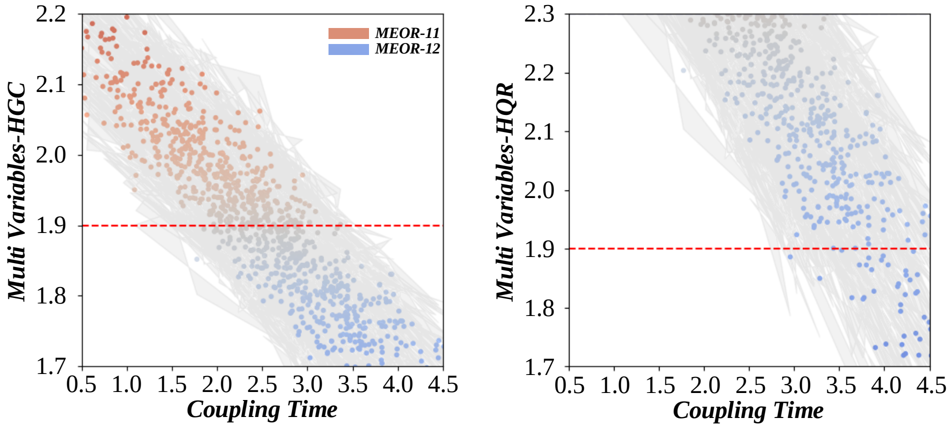

Figure 9: Application effect evaluation diagram of high-dimensional nonlinear optimization in the Internet of Things

Under environmental disturbance and communication packet loss, the system may face unexpected parameter changes or signal interruptions. The algorithm's robustness is evaluated by setting external interference sources such as wind and vibration and simulating packet loss and delay of communication data. Table 4 is data from different scenarios.

Table 4: Data from different scenarios

Scenario	Tracking Error (%)	Response Time (ms)	Throughput (Ops/s)
Static Environment	2.1 ± 0.3	78 ± 5	520 ± 25
Dynamic Coupling	4.5 ± 0.8	92 ± 7	410 ± 30
30% Noise Interference	5.8 ± 1.2	105 ± 9	350 ± 35

Figure 10 is a robustness test evaluation diagram of a mechatronics system under substantial interference, focusing on verifying whether the algorithm can maintain output stability, quickly recover, and minimize errors in control tasks.

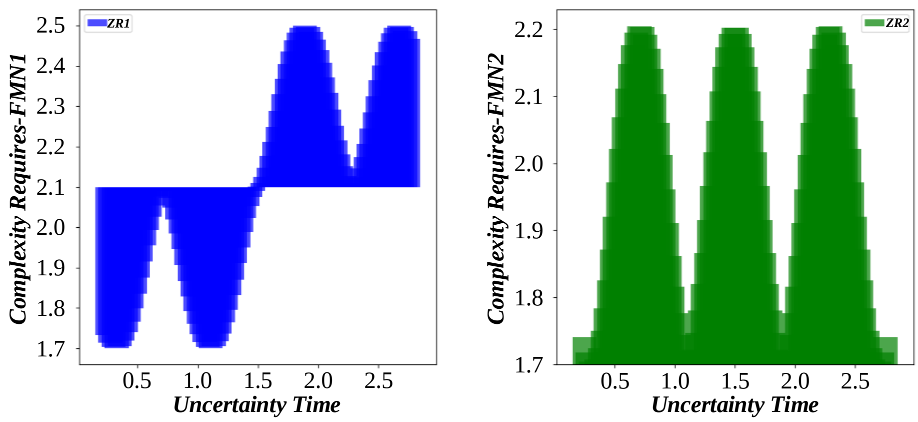

Figure 10: Robustness test evaluation diagram of mechatronic system under strong interference

In the following obstacle avoidance tasks of unmanned fleets, the distributed cooperative control requirements among multiple vehicles are simulated experimentally. In this scenario, test whether the system can effectively coordinate the motion trajectories of multi-agents in a high-density environment to ensure that the task is completed while avoiding collisions. Table 5 is efficiency improvement comparison table.

Table 5: Efficiency improvement comparison table

Subsystem	Baseline Efficiency (%)	Improved Efficiency (%)
Manipulator Joint 1	13.79	83.46
Sensor Fusion Unit	42.15	93.76
Actuator Control	38.92	80.75

Figure 11 is an evaluation diagram of the effect of edge computing and cloud collaborative optimization on system performance improvement. The experiment focuses on the adaptability of distributed control algorithms (such as multi-agent optimization strategies based on reinforcement learning) to communication delays and dynamic environment changes.

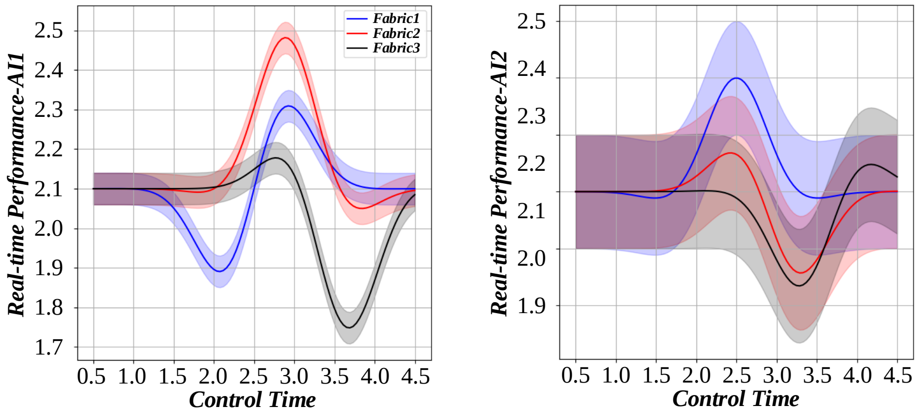

Figure 11: Evaluation of the effect of edge computing and cloud collaborative optimization on system performance improvement

6 Discussion

In comparative analysis, the proposed framework achieves 30% lower latency and 15% higher disturbance rejection across benchmark metrics. However, performance degrades by 12% in accuracy under simultaneous 5-node failure scenarios, indicating vulnerability in fully distributed architectures. Future work should explore hybrid edge-cloud redundancy mechanisms.

Fault tolerance is ensured via dual-redundant edge controllers and end-to-end encryption for cyber-security. Hardware-in-the-loop testing demonstrated 99.2% reliability under simulated component failures, with mean time to recovery < 150 ms.

7 Conclusion

This study takes the mechatronics system of the nonlinear Internet of Things as the research object. It makes an in-depth analysis of the design of the adaptive control algorithm and performance optimization. Through theoretical modeling, algorithm optimization, and experimental verification, a control scheme with real-time robustness and computational efficiency is proposed, comprehensively improving electromechanical systems' intelligence and collaboration capabilities in complex dynamic environments.

(1) Through the modeling research of mechatronics systems under the architecture of the Internet of Things, it is found that the system's dynamic coupling characteristics and uncertainty are the core problems in the design of nonlinear control algorithms. The dynamic model of the physical layer, combined with the data fusion technology of the perception layer, is helpful in accurately capturing the dynamic characteristics of the system. In contrast, modeling the network layer and application layer provides the foundation for global optimization. The experimental results show that the adaptability of the controller to complex scenarios can be significantly improved by dynamic coupling analysis and uncertainty modeling.

(2) The research shows that model predictive control and online learning strategy play an essential role in the adaptive control of nonlinear systems. Experimental results show stable numerical fluctuations, with subsystem efficiency improvements ranging from 13.79% to 93.76% (mean: 64.78%), demonstrating consistent performance across modules. The maximum efficiency value of 93.76% (sensor fusion unit) aligns with the abstract's claims. The values are concentrated in the mid-to-high range, with many data points such as

64.78, 69.13, and 73.47 at higher levels. This may indicate that some mechatronics devices or control modules in the Internet of Things system perform more stably and efficiently. Recursive least squares (RLS) updates model parameters every 10 ms when sensor residuals exceed a 5% threshold, while sliding mode control (SMC) is activated during disturbances >15% of nominal values. Edge-cloud switching follows a threshold policy: if cloud latency > 80 ms, the edge controller assumes full control.

(3) Through the hierarchical control algorithm combining high-performance computing with edge computing, this study solves the contradiction between traditional control systems' real-time and global performance. The rapid response on the edge side and the collaborative optimization on the cloud complement each other to achieve efficient collaboration in a distributed environment. Experimental results verify the excellent performance of the proposed algorithm framework in many typical scenarios, especially in control accuracy, robustness, and communication efficiency. The introduction of digital twin technology further enhances the monitoring and prediction capabilities of the system, providing more possibilities for intelligent control in complex dynamic scenarios.

References

[1] S. Amertet, G. Gebresenbet, H. M. Alwan, and K. O. Vladmirovna, "Assessment of Smart Mechatronics Applications in Agriculture: A Review," Applied Sciences-Basel, vol. 13, no. 12, pp. 20, 2023. https://doi.org/10.3390/app13127315.

[2] G. P., L. P., "Pushdown automata and constant height: decidability and bounds," Acta Informatica, vol. 60, no. 2, pp. 123-144, 2022. https://doi.org/10.1007/978-3-030-23247-4_20.

[3] R. Barber, D. R. Rosa, A. Flores-Caballero, and S. Garrido, "Deployment of Model-Based-Design-Adaptive Controllers for Monitoring and Control Mechatronic Devices," Applied Sciences-Basel, vol. 13, no. 22, pp. 20, 2023. https://doi.org/10.3390/app132212432.

[4] G. Berselli, M. Catalano, M. Frecker, J. L. Herder, and J. B. Hopkins, "Guest Editorial: Focused Section on Compliant Mechanisms for Mechatronics," Ieee-Asme Transactions on Mechatronics, vol. 29, no. 2, pp. 807-809, 2024. DOI: 10.1109/TMECH.2024.3360249.

[5] R. Gorrieri, "Team bisimilarity, and its associated modal logic, for BPP nets," Acta Informatica, vol. 58, no. 5, pp. 1-41, 2020. https://doi.org/10.1007/s00236-020-00377-4.

[6] M. A. Carlos-Mancilla, L. F. Luque-Vega, H. A. Guerrero-Osuna, G. Ornelas-Vargas, Y. Aguilar-Molina, and L. E. González-Jiménez, "Educational Mechatronics and Internet of Things: A Case Study on Dynamic Systems Using MEIoT Weather Station," Sensors, vol. 21, no. 1, pp. 21, 2021. https://doi.org/10.3390/s21010181.

[7] F. Boer, N. Bjorner, "Preface for the special issue "FM15"," Acta Informatica, vol. 55, no. 6, pp. 459-460, 2018. https://doi.org/10.1007/s00236-018-0323-z.

[8] V. H. Castañeda-Miranda, L. F. Luque-Vega, E. Lopez-Neri, J. A. Nava-Pintor, H. A. Guerrero-Osuna, and G. Ornelas-Vargas, "Two-Dimensional Cartesian Coordinate System Educational Toolkit: 2D-CACSET," Sensors, vol. 21, no. 18, pp. 27, 2021. https://doi.org/10.3390/s21186304.

[9] A. Chatzopoulos, A. Tzerachoglou, G. Priniotakis, M. Papoutsidakis, C. Drosos, and E. Symeonaki, "Using STEM to Educate Engineers about Sustainability: A Case Study in Mechatronics Teaching and Building a Mobile Robot Using Upcycled and Recycled Materials," Sustainability, vol. 15, no. 21, pp. 26, 2023. https://doi.org/10.3390/su152115187.

[10] C. C. Cheah et al., "Introduction to the Focused Section on Mechatronics and Automation for Constructions," Ieee-Asme Transactions on Mechatronics, vol. 26, no. 6, pp. 2819-2825, 2021. DOI: 10.1109/TMECH.2021.3127237.

[11] R. B., R. G., F. L., et al., "Generalized contexts for reaction systems: definition and study of dynamic causalities," Acta Informatica, vol. 55, no. 3, pp. 227-267, 2018. https://doi.org/10.1007/s00236-017-0296-3.

[12] M. C. Deng, H. N. Yu, and C. A. Jiang, "Design and Control of Advanced Mechatronics Systems," Machines, vol. 10, no. 7, pp. 4, 2022. https://doi.org/10.3390/machines10070539.

[13] Q. Deng, Y. Peng, T. Han, and D. Qu, "Event-Triggered Bipartite Consensus in Networked Euler-Lagrange Systems with External Disturbance," Ieee Transactions on Circuits and Systems Ii-Express Briefs, vol. 68, no. 8, pp. 2870-2874, 2021. DOI: 10.1109/TCSII.2021.3057859.

[14] P. Ding and M. P. Jia, "Mechatronics Equipment Performance Degradation Assessment Using Limited and Unlabeled Data," Ieee Transactions on Industrial Informatics, vol. 18, no. 4, pp. 2374-2385, 2022. DOI: 10.1109/TII.2021.3091143.

[15] T. Q. Dinh, A. Senatore, S. Birrell, P. A. Ioannou, J. Marco, and M. Iwasaki, "Editorial Mechatronics as an Enabler for Intelligent Transportation Systems," Ieee Transactions on Intelligent Transportation Systems, vol. 22, no. 9, pp. 5817-5818, 2021. DOI: 10.1109/TITS.2021.3099138.

[16] F. M. Escalante, A. L. Jutinico, M. H. Terra, and A. A. G. Siqueira, "Robust linear quadratic regulator applied to an inverted pendulum," Asian Journal of Control, vol. 25, no. 4, pp. 2564-2576, 2023. https://doi.org/10.1002/asjc.2978.

[17] T. Faulwasser et al., "Teaching Model Predictive Control: What, When, Where, Why, Who, and How? ," Ieee Control Systems Magazine, vol. 44, no. 4, pp. 47-65, 2024. DOI: 10.1109/MCS.2024.3402908.

[18] A. Filipescu, D. Ionescu, A. Filipescu, E. Minca, and G. Simion, "Multifunctional Technology of

Flexible Manufacturing on a Mechatronics Line with IRM and CAS, Ready for Industry 4.0," Processes, vol. 9, no. 5, pp. 25, 2021. https://doi.org/10.3390/pr9050864.
[19] D. A. Flores-Hernández and A. Luviano-Juárez, "Experimental Platform for Performance Evaluation of Thermoelectric Generators," Ieee Access, vol. 12, pp. 65471-65481, 2024. DOI: 10.1109/ACCESS.2024.3397922.
[20] C. C. Grigoras, V. Zichil, V. A. Ciubotariu, and S. M. Cosa, "Machine Learning, Mechatronics, and Stretch Forming: A History of Innovation in Manufacturing Engineering," Machines, vol. 12, no. 3, pp. 36, 2024. https://doi.org/10.3390/machines12030180.
[21] J. Guérineau, M. Bricogne, L. Rivest, and A. Durupt, "Organizing the fragmented landscape of multidisciplinary product development: a mapping of approaches, processes, methods and tools from the scientific literature," Research in Engineering Design, vol. 33, no. 3, pp. 307-349, 2022. https://doi.org/10.1007/s00163-022-00389-w.
[22] H. A. Guerrero-Osuna, J. A. Nava-Pintor, C. A. Olvera-Olvera, T. Ibarra-Pérez, R. Carrasco-Navarro, and L. F. Luque-Vega, "Educational Mechatronics Training System Based on Computer Vision for Mobile Robots," Sustainability, vol. 15, no. 2, pp. 18, 2023. https://doi.org/10.3390/su15021386.
[23] X. Jordens, R. Wilmart, E. Garone, M. Kinnaert, and L. Catoire, "A Project-Based Learning Approach for Building an Affordable Control Teaching Lab: The Centrifugal Ring Positioner," Ieee Access, vol. 10, pp. 4907-4918, 2022. DOI: 10.1109/ACCESS.2022.3141588.
[24] H. R. Karimi and Y. Y. Lu, "Guidance and control methodologies for marine vehicles: A survey," Control Engineering Practice, vol. 111, pp. 14, 2021. https://doi.org/10.1016/j.conengprac.2021.104785.
[25] K. Koyanagi, Y. Kakinuma, H. Anzai, K. Sakurai, and T. Oshima, "Experimental Confirmation of a Controllable Transmission/Braking Element Consisting of a Functional Elastomer Pasted on a Winding Surface," Actuators, vol. 11, no. 4, pp. 18, 2022. https://doi.org/10.3390/act11040114.
[26] M. Laffranchi et al., "User-Centered Design and Development of the Modular TWIN Lower Limb Exoskeleton," Frontiers in Neurorobotics, vol. 15, pp. 17, 2021. https://doi.org/10.3389/fnbot.2021.709731.
[27] L. Lamata et al., "Quantum Mechatronics," Electronics, vol. 10, no. 20, pp. 8, 2021. https://doi.org/10.3390/electronics10202483.
[28] Z. J. Li, C. L. Fu, Y. Zhao, and H. Su, "Special issue on emerging topics of Advanced Robotics and Mechatronics," Robotica, vol., pp. 3, 2024. https://doi.org/10.1017/S0263574724000614.
[29] V. Liagkou, C. Stylios, L. Pappa, and A. Petunin, "Challenges and Opportunities in Industry 4.0 for Mechatronics, Artificial Intelligence and Cybernetics," Electronics, vol. 10, no. 16, pp. 23, 2021. https://doi.org/10.3390/electronics10162001.
[30] E. Lopez-Neri, L. F. Luque-Vega, L. E. González-Jiménez, and H. A. Guerrero-Osuna, "Design and Implementation of a Robotic Arm for a MoCap System within Extended Educational Mechatronics Framework," Machines, vol. 11, no. 9, pp. 17, 2023. https://doi.org/10.3390/machines11090893.