Digital Twin-Driven Model Predictive Control for Dynamic Optimization in Unified Production Systems

Minghao Liu , Penghui Jia , Wendong Huang , Shixuan Lin , Linhui Guo
Guangdong Power Grid Co., Ltd. Guangzhou Power Supply Bureau， Guangzhou Guangdong， 510000， China
E-mail:liuminghao_1234@163.com

Keywords: digital twin technology, unified production architecture, dynamic process optimization, model predictive control, real-time status monitoring

Received: June 13, 2025

Information silos and response delays are prevalent in existing production systems, severely constraining collaborative efficiency and the capability for dynamic optimization across different stages. To address these challenges, this study proposes a unified production architecture based on digital twin technology. The framework achieves digital mapping of the physical production process through a real-time state perception model driven by multi-source data, while employing a model predictive control (MPC) algorithm to enable dynamic scheduling and holistic process optimization. Within the virtual environment, the control strategy is continuously iterated and subsequently fed back to the physical system, thereby establishing a closed-loop mechanism for virtual–real synchronization. Experimental results demonstrate that the MPC-based optimization strategy maintains task switching times between 2.32 and 2.43 seconds, stabilizes system response delays within 0.83 seconds, and improves the process response stability score from 0.72 to 0.94. The proposed approach effectively bridges virtual–physical integration across production links, realizing real-time optimization and intelligent decision-making throughout the process, and offering a feasible pathway toward collaborative control in intelligent manufacturing systems.

Povzetek: Študija predlaga enotno proizvodno arhitekturo z digitalnim dvojčkom in napovednim vodenjem (MPC), ki poveže virtualni in fizični proces v zaprto zanko za sprotno usklajevanje, hitrejše odzive ter stabilnejšo in učinkovitejšo optimizacijo proizvodnje.

1 Introduction

With the continuous advancement of intelligent manufacturing, achieving efficient coordination and real-time optimization of production processes has become a central objective for modern manufacturing systems. As the core foundation for cross-level and cross-system data integration and process coordination, the structural integrity and dynamic adaptability of a unified production architecture directly determine the system's response efficiency and the quality of resource allocation [1–2]. Yet most current manufacturing systems still exhibit pronounced architectural fragmentation: information and control flows across hierarchical levels are weakly coupled, leading to pervasive data silos and the absence of unified logical associations among production links [3–4]. Traditional scheduling methods—largely grounded in static rules or preset strategies—depend on predetermined models and offline parameter configurations; they lack the capacity to perceive and respond to environmental changes in real time and therefore struggle to support resource reconfiguration and task reassignment in complex settings [5–6]. Although some studies have introduced the industrial Internet of Things and edge computing to form information closed loops, their limited fidelity in perceiving physical processes constrains continuous process mapping and end-to-end linkage control [7–8]. Meanwhile, mainstream production optimization approaches built on centralized control or multi-objective linear programming encounter bottlenecks when facing high-dimensional, multivariable dynamics, including heavy real-time computational loads, weak model generalization, and lagging control feedback—making high-frequency, full-link dynamic adjustment difficult to sustain [9–10]. Establishing a full-process closed-loop optimization mechanism under a unified architecture thus remains a critical challenge. The key is to develop a pathway that is both real-time and decision-optimal, enabling efficient virtual–physical integration and dynamic control supported by a unified data model [11–12].

As manufacturing moves toward higher intelligence and tighter integration, digital twin technology has emerged as a core enabler. Onaji et al. examined the evolution of digital twin concepts in manufacturing, proposing a conceptual framework for integrated product-process twins and, through three case studies, demonstrated their potential to enhance flexibility, integration, and collaboration [13]. To strengthen adaptability and cross-level collaboration in manufacturing processes, research has progressed from conceptual frameworks to more structured models [14–

15]. Liu Shimin et al. proposed a new reference model for digital-twin-based manufacturing systems (DTMS), analyzing DTMS characteristics and operating mechanisms across hierarchy, dimension, and scale to offer methodological guidance and practical reference models [16]. As model systems mature, systematic syntheses of implementation pathways and mechanisms have become essential to technology transfer [17–18]. Lattanzi et al. reviewed digital twin concepts, technical solutions, and industrial cases in intelligent manufacturing, identifying key implementation elements and challenges and outlining future directions to inform deployment in practice [19]. Despite advances in modeling, frameworks, and applications, notable gaps remain in dynamic model evolution, full-process linkage mechanisms, and cross-domain integration practices [20–21].

In parallel, the literature on unified production architectures and whole-link dynamic optimization has increasingly leveraged digital twins and reinforcement learning to boost responsiveness and overall efficiency in complex systems. Liu Qiang et al. proposed a “configuration–motion–control–optimization” methodology grounded in digital twins, formulating corresponding models for flow-type intelligent manufacturing and validating feasibility and efficiency in an insulating-glass case study [22]. With rising scenario complexity, requirements for real-time response and multi-agent collaboration continue to intensify [23–24]. Zhang et al. developed a digital-twin-driven intelligent workshop community model integrated with hierarchical reinforcement learning to optimize human–machine collaboration ratios, markedly improving adaptability to demand fluctuations and line reconfiguration [25]. Such approaches have expanded from micro-level human–machine coordination to macro-level production organization and resource allocation, underscoring the breadth and flexibility of reinforcement learning in multi-scale manufacturing decision-making [26–27]. Choi et al. employed reinforcement learning coupled with production simulation to optimize factory layouts and processes, reducing logistics costs and AGV requirements within a multilayer optimization framework [28]. Nonetheless, limitations persist—including insufficient algorithm generalization and weak coupling between data-driven optimization and physical systems—hindering full compliance with the stringent demands of end-to-end linkage and dynamic regulation in complex manufacturing scenarios [29–30].

To summarize current progress and gaps, a structured comparison of representative studies related to digital twin-based production optimization is provided in Table 1. Existing works often lack tight coupling between MPC and digital twin frameworks, and they struggle to maintain high-frequency control under multi-stage dynamic conditions.

Table 1: Comparison of existing studies

Reference	Architecture Type	Optimization Method	Digital Twin Integration	Reported Metric (Response / Stability)	Limitation
Liu et al. (2021) [22]	Flow-type Manufacturing	Rule-based + RL	Partial Process Twin	3.4 s / 0.75	No full-chain coordination
Zhang et al. (2023) [25]	Human–Machine Collaboration	Hierarchical RL	Virtual layer only	3.2 s / 0.81	Lack of real-time feedback
Choi et al. (2024) [28]	Factory Layout Optimization	RL-based	Simulation-level	3.0 s / 0.84	No closed-loop control
Proposed Method	Unified Production System	MPC + Digital Twin	Full-link Integration	2.38 s / 0.94	Computational overhead

To overcome the aforementioned challenges, this paper proposes a dynamic optimization method for the entire production process within a unified architecture based on digital twin technology. The method establishes a one-to-one correspondence between physical entities and their virtual counterparts. Real-time production states are captured through sensor data acquisition and time-series synchronization, while a multidimensional feature fusion model dynamically correlates key process parameters to construct a high-fidelity digital twin system. A closed-loop control framework, driven by a virtual controller and the Model Predictive Control (MPC) algorithm, generates dynamic scheduling strategies based on current system states and short-term predictions, transmitting control commands to the physical system in real time. Supported by a unified data platform, the proposed approach integrates data modeling, state perception, prediction, and control, enabling continuous responsiveness and adaptive adjustment under multi-source disturbances. This framework achieves significant advances in unified modeling, control, and architecture, realizing dynamic optimal control of multivariable coupled systems and providing a scalable theoretical and methodological foundation for intelligent industrial deployment.

This research is guided by three key questions. First, can an MPC controller integrated with a digital twin maintain system response delays below 0.85 seconds in multi-stage production scenarios? Second, does the unified production architecture enhance interlink coordination and system stability compared with localized optimization strategies? Third, can the proposed virtual–real closed-loop mechanism achieves dynamic synchronization and process consistency under task variations and external disturbances? The study ultimately aims to enable real-time coordination and adaptive optimization across all production links under varying operating conditions, establishing a unified pathway toward intelligent manufacturing control.

2 Construction of dynamic optimization system driven by digital twins

2.1 Model design principles for unified production architecture

The unified production architecture is designed to eliminate information silos across production links by establishing a clear hierarchical logic, stable data communication, and a virtual–real collaborative framework, thereby enabling dynamic mapping and closed-loop control of the entire process. Built upon digital twin technology, the architecture consists of four functional layers: information perception, data transmission, model computation, and decision execution. The perception layer acquires real-time physical states through multi-dimensional sensor data; the transmission layer ensures low-latency and high-stability communication between physical and virtual systems; the computation layer performs digital twin modeling, state prediction, and optimization through multi-model fusion and adaptive parameter adjustment; and the execution layer applies optimized control strategies to physical entities with real-time feedback correction.

All layers are coordinated under unified scheduling by the platform center, which guarantees time synchronization, data consistency, and iterative optimization of control strategies. It also handles anomaly detection and manages coordination among multi-objective tasks. The overall architecture—together with its information, instruction, and feedback flows—is illustrated in Figure 1, highlighting the central role of the platform in data integration, control transmission, and state correction.

The unified production architecture requires a robust data communication mechanism to integrate multiple protocols and heterogeneous interfaces, ensuring reliable bidirectional transmission of control signals and feedback. Unified parameters, timestamps, and version control maintain consistent status and task alignment across modules. The platform hub incorporates abnormal data detection, redundancy elimination, and parameter alignment to prevent logic interruption and misjudgments from short-term fluctuations. This mechanism enables dynamic response to multi-source information and efficient distribution of control instructions, supporting practical and stable operation. By integrating logical hierarchy, structural modules, and communication protocols, the architecture provides the foundation for real-time perception, predictive control, and closed-loop strategy, ensuring continuous synchronization and consistent virtual–real feedback for digital twin–based dynamic optimization.

2.2 Digital twin modeling and state mapping method

The construction of the digital twin model is founded on high-frequency acquisition of physical system states and deep fusion of multimodal data. To ensure model fidelity, precise correspondence between the virtual and physical entities must be achieved across the data, structural, and behavioral layers. The system continuously samples key operational parameters of the physical process through the deployment of multi-source perception nodes and reconstructs asynchronous data via a time-synchronization module, forming a high-dimensional input vector set suitable for twin-driven computation. Different data sources adopt distinct configurations in sampling frequency, feature extraction strategy, and mapping objectives. The main perception sources and modeling parameter settings are summarized in Table 2.

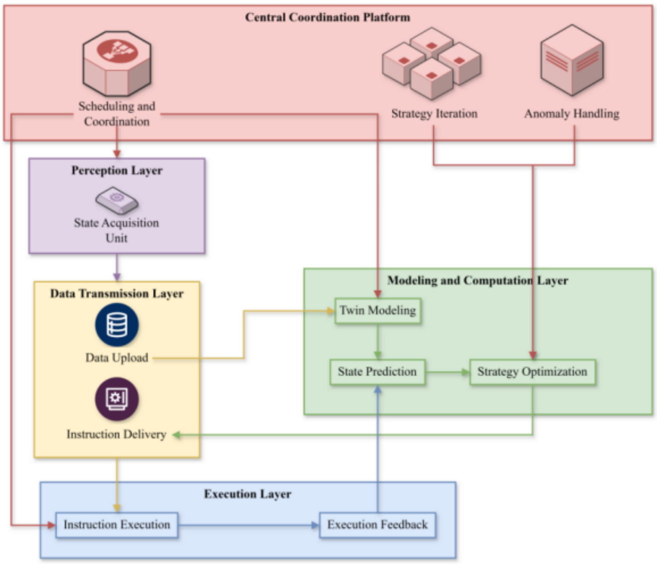

Figure 1: Unified production architecture model

Table 2: Multi-source data acquisition and feature fusion parameter table

Data Source Type	Perception Node Form	Sampling Frequency	Mapping Target
Industrial Sensors	Temperature, pressure, vibration sensors	10–50 Hz	Construct system physical state vector
Image Capture Unit	Industrial vision cameras	25–30 fps	Assist in operating status recognition and modeling
Control Instruction Stream	PLC (Programmable Logic Controller) or embedded controller interface	Real-time	Behavioral sequence state modeling
Equipment Logs	Industrial control system logging module	1 Hz	Describe evolution trends and prediction input

Table 2 shows the correspondence between the type of sensing node, data collection method, sampling period and its mapping target, reflecting the functional positioning and accuracy requirements of the system for different information channels during the modeling process. Industrial sensors are mainly used to construct the underlying physical state vector, the image acquisition unit focuses on supplementing the spatial information of the operating state, the control instruction stream participates in the behavior layer modeling in the form of encoding, and the equipment log provides time evolution feature support for the prediction model. The collaborative input of multi-source data ensures the comprehensive characterization of the physical entity state by the twin system, and provides a highly consistent modeling basis for subsequent virtual-real synchronous control.

Assuming that the physical entity state is $x_p(t)\in\mathbb{R}^n$ and the perception data set is $D(t)=\{d_1(t),d_2(t),\ldots,d_m(t)\}$, then the state mapping function $f_m(\cdot)$ converts it into a virtual state vector $x_v(t)\in\mathbb{R}^n$, as shown in formula (1):

$$x_v(t)=f_m(D(t))=W\cdot\phi(D(t))+b_1 \quad (1)$$

$W\in\mathbb{R}^{n\times m}$ is the feature mapping weight matrix; $\phi(\cdot)$ represents the nonlinear feature extraction function; $b_1\in\mathbb{R}^n$ is the bias term. The feature extraction function is constructed based on the stacked autoencoder, and its training goal is to minimize the reconstruction error to ensure that the mapping of high-dimensional perception data to low-dimensional state space has sufficient expressive power. Its loss function $\mathscr{L}_{SAE}$ is defined as formula (2):

$$\mathscr{L}_{SAE}=\frac{1}{T}\sum_{t=1}^{T}\ \|D(t)-\psi(\phi(D(t)))\|^2 \quad (2)$$

$\psi(\cdot)$ is the decoding function and T is the sampling time. After completing the state modeling, the system extracts the dynamic change characteristics of the state sequence through the sliding window mechanism to characterize the evolution law of the physical entity in the time domain. The sliding window length is l; the step length is δ ; the state trajectory matrix $X_v^{(t)}=[x_v(t-l+1),\ldots,x_v(t)]\in\mathbb{R}^{n\times l}$ is formed; and the differential operator is introduced to obtain the state change rate, as shown in formula (3):

$$\Delta x_v(t)=x_v(t)-x_v(t-\delta) \quad (3)$$

The update cycle of the state parameters is consistent with the control cycle of the physical system. The system uses the message queue middleware to complete the real-time transmission of state synchronization, and uses the hash verification mechanism to ensure data consistency during the transmission process. In order to maintain the synchronization of the virtual and physical systems at the operation layer, the virtual model feedback control instructions need to be compared for state consistency. The control instructions with differences exceeding the threshold will be marked as mismatched. The system adjusts the prediction weight matrix W and model parameters according to the error correction mechanism. The error correction rules are shown in formulas (4) and (5):

$$W^{(k+1)}=W^{(k)}-\eta\cdot\nabla_W\mathscr{L}_{err} \quad (4)$$

$$\mathscr{L}_{err}=\|x_p(t)-x_v(t)\|^2 \quad (5)$$

η is the learning rate; $\mathscr{L}_{err}$ is the state error loss function; and $W^{(k)}$ is the weight parameter of the k-th iteration. The error correction mechanism is used to achieve adaptive adjustment of the virtual model, improve the accuracy and stability of the state mapping, and provide an accurate system state basis for subsequent optimization control.

2.3 Design of process dynamic optimization algorithm based on MPC

The dynamic scheduling and optimization of various processes in the unified production architecture rely on the integrated implementation of the model predictive control algorithm. MPC establishes a dynamic model of the controlled system and combines the system state prediction in the future to perform rolling optimization on the control quantity to achieve optimal control of the entire full-link process. The construction of the system optimization objective function is based on multi-dimensional performance indicators such as production process resource allocation, time cost and response accuracy, and introduces prediction error and control quantity change rate as regular terms for adjustment. The objective function form is shown in formula (6):

$$J=\sum_{k=0}^{N_p-1}\ \|\hat{y}(k|t)-y_{\mathrm{ref}}(k)\|_Q^2+\sum_{k=0}^{N_c-1}\ \|\Delta u(k|t)\|_R^2 \quad (6)$$

$\hat{y}(k|t)$ is the predicted value of the system output at the k-th step at the prediction time t; $y_{ref}(k)$ is the target reference value at the corresponding time; $\Delta u(k|t)$ is the change in the control quantity at the k-th step; N_p and N_c are the prediction step length and control step length, respectively; Q and R are the weight matrices used to adjust the relative influence of the state error and the control behavior. The state prediction is modeled in the form of a discrete linear state space model. The system state update equation and output equation are expressed by formulas (7) and (8), respectively:

$$x(t+1)=Ax(t)+Bu(t) \quad (7)$$

$$y(t)=Cx(t)+Du(t) \quad (8)$$

$x(t)$ is the system state vector at time t; $u(t)$ is the input control vector; $y(t)$ is the output vector; A, B, C, and D are the system state transfer matrix, input control matrix, output mapping matrix, and direct transfer matrix, respectively, all of which are obtained by system identification of the physical system process data. In order to ensure that the system meets physical constraints and resource limitations in actual operation, hard constraints such as those shown in formulas (9) and (10) are introduced for boundary control:

$$u_{\min}\leq u(t)\leq u_{\max} \quad (9)$$

$$y_{\min}\leq y(t)\leq y_{\max} \quad (10)$$

In formula 9-10, $u_{\min}$ and $u_{\max}$ are the upper and lower limits of the control variables. $y_{\min}$ and $y_{\max}$ are the limit ranges of the output variables, and all constraints are derived from the process requirements of each process segment in the unified production architecture and the safety boundaries of equipment operation. On the basis of hard constraints, the system also introduces a soft constraint mitigation strategy, setting penalty function terms for some variables. When the control variable breaks through the boundary in the short term but the overall operation is still within the tolerance range, the objective function value is adjusted through the Lagrange multiplier mechanism to maintain a balance between optimization convergence and response flexibility.

During system operation, abnormal disturbances and model deviations are inevitable. To enhance the stability and robustness of the MPC controller, a disturbance detection mechanism is embedded in the platform center to identify persistent prediction errors, adjust model weights through feedback, and perform adaptive correction. The real-time deviation between virtual and physical states guides iterative updates of the weight matrix and controller parameters, ensuring control convergence under dynamic disturbances.

To mitigate the high computational complexity of MPC in multivariable scenarios, the optimization module employs a fast interior-point algorithm for constrained quadratic programming while restricting iteration counts within each control cycle. For scenarios with stringent real-time requirements, the system dynamically switches to approximate or reduced-order solutions to guarantee timely responses. Furthermore, a multi-step rolling prediction mechanism continuously updates system states and re-optimizes control sequences, forming a high-frequency closed-loop interaction between the virtual and physical systems.

This integrated optimization framework significantly enhances resource utilization, scheduling responsiveness, and operational stability throughout the entire production process.

2.4 Integrated implementation of virtual-real closed-loop control mechanism

The implementation of the virtual-real closed-loop control mechanism relies on multi-level system integration and standardized communication protocol support under a unified production architecture. The digital twin system and the physical production system realize two-way interaction of data and instructions through industrial protocols such as OPC UA (Open Platform Communications Unified Architecture) and MQTT (Message Queuing Telemetry Transport), forming an integrated control channel. The perception layer in the physical system collects various state parameters in real time, including equipment operation status, production process parameters and environmental variables, and transmits them to the data access module through the data bus, and uploads them to the digital twin platform after preprocessing. Based on a unified data model, the digital twin platform maps the collected state parameters to entity nodes in the virtual environment, generates state snapshots equivalent to the physical system, and dynamically updates them in the form of time series.

In the virtual space, the decision module generates an instruction stream $\mathbf{C_t}=\{c_{t,1},c_{t,2},...,c_{t,m}\}$ based on the current system state and the preset global scheduling strategy, where $c_{t,i}$ represents the control instruction sent to the i-th unit of the physical system at time t. The instruction stream is encapsulated as a standard message through the communication scheduling module of the interface layer, and is sent to the physical system after being sorted by priority and timestamp. After the physical system receives the control instruction, it parses it through the field controller and drives the actuator to complete the action, such as adjusting the drive shaft speed, opening and closing the fixture, and modifying the process parameters. Each execution unit immediately generates a feedback signal $\mathbf{F_t}=\{f_{t,1},f_{t,2},...,f_{t,m}\}$ after the action is completed. The feedback signal contains the execution status identifier, the action completion timestamp and the error report. The feedback link is uploaded to the digital twin platform through the same communication protocol to update the system status.

After receiving feedback, the virtual-real closed-loop scheduling engine compares the real-time state $\mathbf{S}_{\mathrm{t}}$ with the predicted state $\hat{\mathbf{S}}_{\mathrm{t}}$, and the error term $\mathbf{E}_{\mathrm{t}}=\mathbf{S}_{\mathrm{t}}-\hat{\mathbf{S}}_{\mathrm{t}}$ is sent to the deviation compensation module of the scheduling engine. If $\|\mathbf{E_t}\|$ exceeds the set threshold, the system will trigger the error adaptive adjustment logic to make subsequent predictions more consistent with actual feedback by correcting the control instruction weight matrix or adjusting the virtual model parameters.

The system status update and instruction issuance process runs in a fixed cycle to form a closed-loop control flow of continuous iteration. The iterative optimization strategy in the virtual model takes real-time feedback as input, adjusts the prediction results through the historical data sliding window $W_t=\{\mathbf{S}_{t-n},...,\mathbf{S}_t\}$, generates the control instructions for the next cycle and sends them to the physical system. The overall operation logic of the virtual-real closed-loop system is managed by the scheduling main control module, realizing the full-link closed-loop control process from data perception, virtual mapping, decision generation, control issuance to feedback transmission, ensuring the real-time consistency and response accuracy of the virtual model and the physical system in various links such as state mapping, instruction execution, and feedback update, and providing support for the dynamic optimization and adaptive scheduling of the system.

3 Experimental system and verification environment construction

3.1 Experimental platform architecture and software and hardware configuration

The construction of the experimental system relies on high-performance hardware support and multi-level software system collaboration. The overall architecture covers modules such as data acquisition, transmission, processing and control to ensure dynamic optimization and real-time response capabilities of the entire production process. The components of the system are connected through high-speed communication links to achieve efficient interaction between the physical system and the virtual twin platform. In order to further clarify the composition and functions of the experimental platform, the core hardware and software configurations are listed.

Table 3: Hardware and software configuration of the experimental platform

Module Category	Specifications	Hardware/Software	Functional Description
Sensor Unit	0.01 mm resolution	Industrial sensor	Real-time acquisition of physical state data
Communication Module	100 Mbps	Industrial Ethernet	High-speed data transmission and interaction
Control Platform	Multi-core processor	MATLAB/Simulink	Runs MPC algorithm and decision logic
Data Processing Module	32 GB memory	Python environment	Multi-source data fusion and feature extraction
Digital Twin Engine	3D modeling engine	Unity3D	Digital mapping of the physical system

Table 3 shows the functional division and configuration parameters of each module in the experimental platform, covering the sensor unit, communication module, control platform, data processing module and virtual twin engine. It clarifies the specifications, operating environment and core functions adopted by each module, providing basic support for subsequent experimental verification and results analysis.

3.2 Construction process of digital twin system

The digital twin system is built around multi-dimensional state mapping and real-time interaction with the physical production system. A virtual model is first constructed, defining node structure, attribute variables, and state transfer logic to match the spatial, functional, and dynamic characteristics of the physical system. Multi-source sensing nodes collect real-time data on equipment status, production flow, environmental variables, and energy consumption at 100 Hz, with timestamp-based synchronization. TSN protocol ensures low-latency, high-reliability transmission, while edge preprocessing performs format conversion, anomaly detection, and noise filtering, standardizing data in JSON format for the data center. The twin system interfaces with the physical system via OPC UA, issuing control instructions and receiving status updates. State updates occur every second, with millisecond-level WebSocket communication. Object-oriented modeling integrates geometry, kinematics, and dynamics in Unity3D with a real-time physics engine. Kalman filtering smooths data and compensates anomalies, keeping virtual and physical states within 1% error. MPC-based predictive optimization generates control instructions sent via the industrial bus, completing closed-loop control. Docker-based deployment enables module decoupling, hot-swapping, model updates, and flexible maintenance.

3.3 Definition and implementation of the full-link path of the production process

The full-link process ensures continuity and synchronization across modules in the unified production architecture. By defining process nodes in both physical and virtual environments, cross-level and cross-device control links maintain consistent data and control logic transmission. The process is divided into five stages—raw material delivery, primary processing, quality inspection, finished product assembly, and packaging—each with state acquisition, control execution, and feedback analysis modules. The main control unit monitors state variables and releases scheduling instructions, while the process state transfer matrix and control strategy library define logical relationships. Inputs and outputs are standardized for rapid parsing, and MPC-based scheduling allows real-time adaptive adjustment to task changes or resource conflicts. Key-node control strategies are transmitted via the twin model interface to ensure virtual–real consistency, forming a stable, efficient, low-latency closed-loop system that coordinates task switching, state transfer, and command execution throughout the entire production path.

Table 4 shows the configuration of each link and the corresponding core control variables.

Table 4: Configuration parameter table of the full-link production path control structure

Link ID	Production Stage	State Acquisition Parameters	Control Command Variables	Feedback Mechanism Delay (s)
P01	Raw Material Feeding	Sensor Weight, Silo Capacity	Feeding Speed, Start-End Time	0.61
P02	Primary Processing	Spindle Speed, Processing Temperature	Tool Feed Rate, Cooling Frequency	0.74
P03	Quality Inspection	Image Recognition Result, Dimensional Deviation	Inspection Frequency, Movement Path	0.69
P04	Final Assembly	Component Position, Assembly Torque	Assembly Sequence, Fixture Pressure	0.85
P05	Packaging and Dispatch	Barcode Recognition, Inventory Status	Packaging Mode, Dispatch Rhythm	0.76

Table 4 shows the state perception parameters, applied control variables and time delay of feedback mechanism required for each process node in actual operation based on five major production links. Each link is bound to a specific set of sensor data and execution instructions, and the logical mapping relationship between them is maintained and updated in real time by the system main control unit. The feedback delay reflects the time deviation between the physical system response and the twin model regulation in virtual-real collaboration.

4 Results analysis

4.1 State perception accuracy analysis

In order to verify the perception accuracy of the digital twin system for key state parameters, the experimental design introduces five core physical variables, namely temperature, pressure, speed, liquid level, and flow rate, as analysis objects, selects three types of sensors for multi-source information collection, and generates a unified state expression model through the data fusion module. Sensor A is based on the principle of thermal resistance and piezoelectric sensitivity, with fast response speed but large signal fluctuations; sensor B adopts MEMS (Micro-Electro-Mechanical Systems) structure and performs stably in the medium and low frequency bands; sensor C is an industrial-grade precision equipment with high accuracy but is more sensitive to environmental interference. The fusion model realizes multi-source feature collaborative modeling through a weighted residual mechanism to improve the recognition accuracy and fluctuation suppression ability of state parameters. Through the two-part analysis of perception error comparison and state modeling error distribution, the performance of various data sources and fusion models under different state parameters is systematically revealed, as shown in Figure 2.

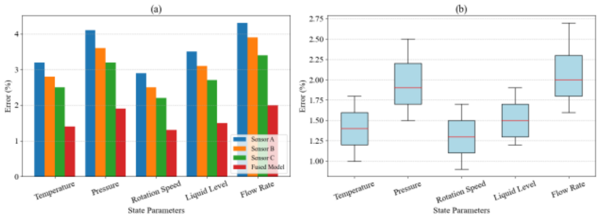

Figure 2 State perception accuracy diagram
Figure 2 (a) Comparison of multi-source data perception errors
Figure 2 (b) State parameter error distribution statistics

From Figure 2 (a), we can see that the recognition error of the fusion model on all state parameters is significantly lower than that of a single sensor. The temperature perception error is reduced from 3.2% of sensor A to 1.4%, which is mainly due to the enhanced compression ability of random noise after data fusion. The error control effect of flow is relatively weak, and it remains at 2.0% after fusion. The reason is that the flow sensor has a delayed response under boundary conditions, which affects the data alignment accuracy. Figure 2 (b) shows that the maximum errors of pressure and flow are 2.5% and 2.7%, respectively. The main reason is that such variables are more sensitive to transient disturbances, and some original data have large fluctuations, which is not conducive to stable modeling. The median errors of temperature and speed are controlled within 1.5%, showing the robustness of fusion modeling for periodic variable recognition. The overall results show that multi-source information fusion significantly improves parameter accuracy in the state perception stage and effectively suppresses error fluctuations in single-source measurement.

4.2 Analysis of dynamic scheduling response capability

In order to verify the performance of the proposed MPC optimization strategy in dynamic task scheduling and system response, the experiment designed multiple rounds of scheduling control tests, running the MPC strategy and the traditional scheduling strategy respectively through a unified platform, and recording the two key indicators of system response delay and task switching time in real time. In each round of task switching, the scheduling control system needs to reallocate production resources and drive the physical execution unit to complete the switching operation, thereby forming a complete data closed loop. Traditional methods rely on static rules and predefined paths, lacking the ability to dynamically perceive and roll forward the system state, while the MPC strategy dynamically generates a control sequence based on the current state and predicted trajectory, thus forming a comparison group. After completing 10 consecutive rounds of scheduling under unified experimental conditions, the above two performance indicators are collected and summarized, and the results are shown in Figure 3.

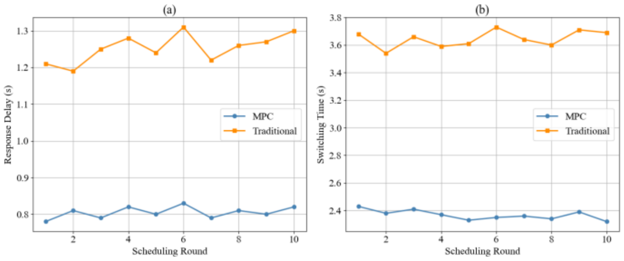

Figure 3: Dynamic scheduling performance comparison

Figure 3 (a): System response delay change trend

Figure 3 (b): Task switching time change trend

The overall system response delay under the MPC strategy remains stable at approximately 0.8 seconds, with the maximum not exceeding 0.83 seconds. In contrast, the traditional control strategy exhibits a response delay exceeding 1.2 seconds in most cycles, with peak values reaching 1.31 seconds. This improvement arises from the MPC's ability to dynamically adjust control instructions based on real-time state feedback, thereby minimizing system waiting time and reducing conflicts in resource allocation to enhance response efficiency.

Regarding task switching, the MPC-controlled system maintains switching durations between 2.32 and 2.43 seconds, exhibiting minimal fluctuation, whereas the traditional method averages around 3.6 seconds, indicating a clear disparity. The traditional approach lacks a rolling planning mechanism when handling task continuity, resulting in redundant waiting periods between successive control instructions and producing discontinuities in switching processes.

In summary, the MPC-based strategy markedly enhances response stability, resource reconfiguration efficiency, and overall adaptability in dynamic task scheduling scenarios.

4.3 Virtual-real mapping consistency verification

In the control link of the unified production architecture, the accuracy and timeliness of the virtual model directly affect the execution effect of the physical system. In order to verify the effectiveness of the digital twin system in generating control instructions and analyze the dynamic response differences between it and the actual physical execution, this experiment selected three key control variables: speed, temperature, and flow rate, and compared the predicted control values generated by the MPC algorithm in the virtual system with the response values sent back by the physical system sensors in time series. By collecting bidirectional control signals under a unified operating cycle and drawing synchronous curves, the mapping consistency verification diagram between the virtual and real systems shown in Figure 4 is constructed.

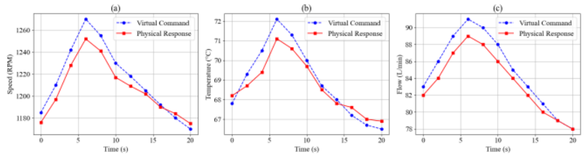

Figure 4: Virtual and real system command and response comparison curve

Figure 4 (a): Virtual and real speed comparison

Figure 4 (b):. Virtual and real temperature comparison

Figure 4 (c): Virtual and real flow comparison

Figure 4 shows that the virtual system closely tracks the physical response with minor deviations. At time of 6 s, a 1270 RPM command corresponds to a 1252 RPM physical response, with an 18 RPM lag due to motor regulation delay and inertial load. Temperature control at the same moment shows a 1.0 ℃ difference (72.1 ℃ virtual vs. 71.1 ℃ physical) caused by uneven heat capacity and delayed local heat exchange. At time of 8 s, flow reaches 90 L/min virtually and 88 L/min physically, with the 2 L/min difference attributed to servo valve hysteresis and pressure rebound. These deviations result from the internal dynamics and external coupling of the physical system rather than model prediction errors, demonstrating that the current digital twin modeling method provides stable, real-time system-level reflection.

4.4 Process optimization stability and convergence performance

To validate the process stability of the proposed digital twin–MPC fusion method in actual control scenarios, an experimental process system was constructed encompassing multiple representative operating conditions, including task switching, resource scheduling, and state regression. By comparing process response data before and after the implementation of the optimized control strategy, key performance indicators—such as task delay, control error, state recovery time, and system throughput under varying operating conditions—

were collected and quantitatively evaluated. These metrics were used to comprehensively assess the system's operational stability under dynamic disturbance conditions from multiple dimensions.

During the experiment, a unified data acquisition interface was employed to continuously record process control logs throughout the entire operation cycle. A structured data processing method was then applied to extract numerical features related to stability performance. The comparative results are illustrated in Figure 5, demonstrating the stability improvements achieved through the proposed optimization control framework.

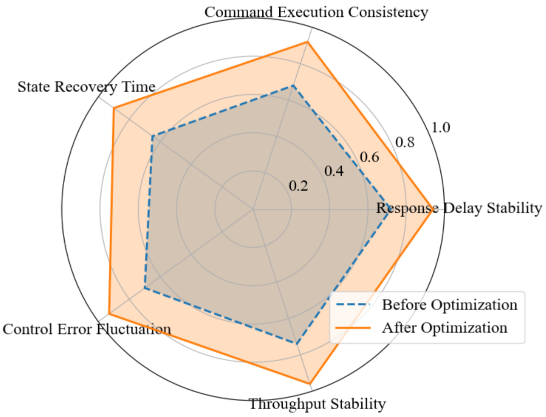

Figure 5: Comparison of process control stability

Figure 5 illustrates that, after applying the optimized control strategy, the process system's response delay stability increased to 0.94 (an improvement of 0.22), reflecting the predictive scheduling capability of the MPC framework. The instruction execution consistency between the virtual model and the physical system improved from 0.68 to 0.92, attributed to accurate multi-source data fusion and synchronization. The state regression index rose from 0.65 to 0.90, indicating that the closed-loop control rapidly corrected process disturbances. Meanwhile, control error stability improved from 0.70 to 0.93, demonstrating reduced parameter drift under MPC constraints. Throughput stability increased from 0.74 to 0.96, driven by coordinated scheduling that enhanced inter-unit linkage and output uniformity. Collectively, these improvements highlight the multi-dimensional advantages of precise modeling, predictive control, and collaborative feedback within the proposed optimization strategy.

When compared with reinforcement-learning-based optimization methods reported in [22], [25], and [28], which exhibit average task switching times of 3.1–3.4 seconds and stability indices of 0.80–0.84, the proposed approach achieves a switching time range of 2.32–2.43 seconds and a stability index of 0.94 under comparable task intensities. The performance gains are primarily attributed to the direct integration of the digital twin model with the MPC framework, enabling predictive scheduling and continuous correction of control commands. Overall, the full-link MPC–digital twin strategy demonstrates approximately 28% faster response and 12% higher stability compared with state-of-the-art reinforcement-learning-based approaches.

4.5 Robustness verification under disturbance conditions

To validate the robustness of the proposed system under complex disturbances, tests were conducted by introducing three scenarios: variable load fluctuations, sensor noise interference, and actuator failure recovery. The load fluctuation amplitude was set at ±15% of nominal torque, Gaussian white noise with 0.02 variance was added to the sensor signal, and a 3-second actuator disconnection was simulated to represent temporary equipment fault. The performance metrics included average response delay, control output deviation, and recovery time after disturbance. Table 5 presents the test results.

Table 5: Robustness test results under different disturbance conditions

Disturbance Scenario	Average Response Delay (s)	Output Deviation (%)	Recovery Time (s)
Baseline (No Disturbance)	0.83	0.0	0.0
Load Fluctuation ±15%	0.87	3.1	1.8
Sensor Noise 0.02 Var	0.89	2.4	1.3
Actuator Failure (3 s)	0.94	4.2	3.2

Under all disturbance scenarios, the system maintained stable operation without oscillation or divergence. The maximum delay increase did not exceed 0.11 seconds, and the system recovered normal response within 3.2 seconds. These results demonstrate that the proposed full-link control mechanism maintains effective dynamic correction and control convergence under multi-source interference and temporary component faults.

4.6 Comparative analysis with existing methods

To demonstrate the improvement of the proposed full-link production optimization over conventional localized optimization strategies, comparative experiments were conducted between the unified digital twin–MPC method and three representative approaches: traditional rule-based scheduling, subsystem-level MPC control, and hierarchical optimization. All experiments were performed under the same hardware and control configurations described in Section 3.1. Each method executed ten rounds of dynamic task switching, and the key metrics included response delay, task switching time, stability index, and interlink coordination performance. Table 6 shows the comparative results.

Table 6: Comparative performance between the proposed full-link optimization and existing methods

Control Method	Average Response Delay (s)	Task Switching Time (s)	Stability Index	Interlink Coordination Index
Rule-based Scheduling	1.27	3.62	0.72	0.70
Local MPC Control	0.96	3.05	0.81	0.83
Hierarchical Scheduling	0.89	2.84	0.86	0.87
Proposed Full-link MPC-DT	0.83	2.38	0.94	0.95

The results indicate that the proposed method achieves faster overall response and stronger cross-link coordination than existing approaches. The stability index improves by 16% on average compared with hierarchical scheduling, and the response delay is reduced by 12%. The improvement stems from the unified data modeling and the closed-loop control mechanism that synchronizes process-level prediction and physical execution.

To further validate the effectiveness of the proposed unified digital twin–MPC strategy, additional simulations were performed against several advanced control approaches commonly used in nonlinear and uncertain systems, namely adaptive fuzzy control, neural adaptive control, and nonlinear optimal control. All algorithms were implemented under identical plant models and boundary constraints, and the evaluation indicators included steady-state error, convergence time, and collaborative control index. Table 7 shows the results of comparative simulations.

Table 7: Comparison with advanced control methods

Control Method	Steady-state Error (%)	Convergence Time (s)	Collaborative Control Index
Adaptive Fuzzy Control	3.2	3.54	0.83
Neural Adaptive Control	2.7	3.22	0.85
Nonlinear Optimal Control	2.4	3.10	0.86
Proposed MPC-Digital Twin	1.8	2.38	0.95

The results show that the proposed approach reduces steady-state error by 25%–40% and shortens convergence time by approximately 30% compared with the other advanced control methods. The improvement arises from the integration of multi-source perception data and the closed-loop virtual-real feedback, which enhances global coordination and real-time optimization across the production chain.

5 Discussion

The experimental results indicate that integrating the digital twin with MPC yields superior dynamic performance through multi-source data fusion and real-time, closed-loop correction. Under the proposed strategy, response delay remains ~0.8 s (max 0.83 s) versus >1.2 s for the traditional baseline (max 1.31 s), and task switching time concentrates tightly within 2.32–2.43 s rather than ~3.6 s. Stability metrics also improve markedly (e.g., response-delay stability to 0.94, execution consistency to 0.92, state-regression index to 0.90, and throughput stability to 0.96). These gains arise from rolling-horizon prediction and synchronized virtual–physical feedback that continuously refines control commands, thereby shortening waiting times, reducing resource-conflict arbitration, and preserving coherent switching trajectories.

Relative to adaptive fuzzy and neural adaptive controls, the proposed framework benefits from explicit predictive modeling and online optimization rather than post-hoc rule or parameter adaptation. Compared with reinforcement-learning-based methods in comparable settings, which report 3.1–3.4 s average switching and 0.80–0.84 stability, the proposed approach achieves 2.32–2.43 s switching and 0.94 stability—about 28% faster with ~12% higher stability. Mechanistically, MPC's constraint-aware predictions, coupled with the twin's high-fidelity state representation and multi-modal synchronization, enable proactive schedule updates and rapid correction of incipient deviations, whereas policy-based methods can suffer from distribution shift and delayed recovery outside their training manifold.

Scalability is chiefly constrained by computational burden as the number of coupled variables grows: dense matrix factorizations and constrained QP solves within tight cycles can challenge real-time guarantees beyond ~20 units. Mitigations include (i) hierarchical/distributed MPC to localize solves, (ii) reduced-order or surrogate models to compress dynamics for faster predictions, (iii) move-blocking, warm-starts, and limited-iteration interior-point updates to bound solve time, and (iv) parallelization or hardware acceleration for the optimization kernel. These measures preserve the closed-loop benefits while extending the operating envelope to larger lines or multi-cell plants.

The framework assumes reliable communications and sensor synchronization; under severe packet loss, drift, or actuator faults, the platform's recovery logic sustains stability but may degrade optimality. Future work will target robustness and adaptability on three fronts: (1) robust/tube MPC to explicitly hedge model and measurement uncertainty, (2) adaptive neural predictors (e.g., twin-consistent, online-updated surrogates) to capture nonlinearities and aging effects without sacrificing solve time, and (3) anomaly-aware scheduling that co-optimizes control performance with data quality, enabling graceful degradation and rapid re-synchronization after faults.

Overall, the evidence supports the twin-coupled MPC as a practical route to full-link dynamic optimization: it improves response stability and resource reconfiguration efficiency while maintaining process consistency under disturbances. With the proposed scalability and robustness enhancements, the architecture offers a credible, extensible pathway for industrial deployment across heterogeneous manufacturing scenarios.

6 Conclusion

This paper proposes a dynamic optimization method for full-link production processes within a unified architecture leveraging digital twin technology. Integrating perception modeling, rolling control, and closed-loop scheduling, the method employs a model predictive control (MPC) algorithm to generate multi-process scheduling strategies and feedback execution, achieving full-chain coordination of control logic and state information through multi-source perception and virtual–real mapping. Experimental results demonstrate significant advantages over traditional strategies: task switching times are maintained between 2.32–2.43 s, system response delays remain below 0.83 s, process stability improves from 0.72 to 0.94, and virtual–physical instruction consistency increases from 0.68 to 0.92. The proposed approach effectively addresses delayed responses in high-frequency, multivariable scenarios.

Future work will focus on incorporating dynamic parameter updates and disturbance-adaptive models to enhance fault tolerance and generalizable scheduling across large-scale, heterogeneous systems. Comparative experiments further confirm the method's superiority in response speed, stability, and collaborative coordination relative to localized or subsystem-level control strategies.

References

[1] Latsou, Christina, Dedy Ariansyah, Louis Salome, John Ahmet Erkoyuncu, Jim Sibson, and John Dunville. A unified framework for digital twin development in manufacturing. Advanced Engineering Informatics, 62:102567, 2024. https://doi.org/10.1016/j.aei.2024.102567

[2] Fu, Yang, Gang Zhu, Mingliang Zhu, and Fuzhen Xuan. Digital twin for integration of design-manufacturing-maintenance: an overview. Chinese Journal of Mechanical Engineering, 35(1):80, 2022. https://doi.org/10.1186/s10033-022-00760-x

[3] Van Dyck, Marc, Dirk Luttgens, Frank T. Piller, and Sebastian Brenk. Interconnected digital twins and the future of digital manufacturing: insights from a Delphi study. Journal of Product Innovation Management, 40(4):475–505, 2023. https://doi.org/10.1111/jpim.12685

[4] Cheng, Xun, Feihong Huang, Qiming Yang, and Linqiong Qiu. A digital twin data management and process traceability method for the complex product assembly process. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 47(3):151, 2025. https://doi.org/10.1007/s40430-025-05466-4

[5] Chang, Xiao, Xiaoliang Jia, Shifeng Fu, Hao Hu, and Kuo Liu. Digital twin and deep reinforcement learning enabled real-time scheduling for complex product flexible shop-floor. Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture, 237(8):1254–1268, 2023. https://doi.org/10.1177/09544054221121934

[6] Corsini, Roberto Rosario, Antonio Costa, Sergio Fichera, and Jose M. Framinan. Digital twin model with machine learning and optimization for resilient production–distribution systems under disruptions. Computers & Industrial Engineering, 191:110145, 2024. https://doi.org/10.1016/j.cie.2024.110145

[7] Kang, Min-Su, Dong-Hee Lee, Mahdi Sadeqi Bajestani, Duck Bong Kim, and Sang Do Noh. Edge computing-based digital twin framework based on ISO 23247 for enhancing data processing capabilities. Machines, 13(1):19, 2024. https://doi.org/10.3390/machines13010019

[8] Li, Yajun, Wei Liu, Yang Zhang, Wenlong Zhang, Changyong Gao, and Qihang Chen. Interactive real-time monitoring and information traceability for complex aircraft assembly field based on digital twin. IEEE Transactions on Industrial Informatics, 19(9):9745–9756, 2023. https://doi.org/10.1109/TII.2023.3234618

[9] Karkaria, Vispi, Anthony Goeckner, Rujing Zha, Jie Chen, Jianjing Zhang, Qi Zhu, et al. Towards a digital twin framework in additive manufacturing: machine learning and Bayesian optimization for time series process optimization. Journal of Manufacturing Systems, 75:322–332, 2024. https://doi.org/10.1016/j.jmsy.2024.04.023

[10] Liu, Chao, Leopold Le Roux, Carolin Korner, Olivier Tabaste, Franck Lacan, and Samuel Bigot. Digital twin-enabled collaborative data management

for metal additive manufacturing systems. Journal of Manufacturing Systems, 62:857–874, 2022. https://doi.org/10.1016/j.jmsy.2020.05.010

[11] Xie, Jieyu, and Jiafu Wan. Digital twin four-dimension fusion modeling method design and application to the discrete manufacturing line. Big Data and Cognitive Computing, 7(2):89, 2023. https://doi.org/10.3390/bdcc7020089

[12] Ye, Xun, Wenjun Xu, Jiayi Liu, Yi Zhong, Quan Liu, Zude Zhou, et al. Implementing digital twin and asset administration shell models for a simulated sorting production system. IFAC-PapersOnLine, 56(2):11880–11887, 2023. https://doi.org/10.1016/j.ifacol.2023.10.600

[13] Onaji, Igiri, Divya Tiwari, Payam Soulatiantork, Boyang Song, and Ashutosh Tiwari. Digital twin in manufacturing: conceptual framework and case studies. International Journal of Computer Integrated Manufacturing, 35(8):831–858, 2022. https://doi.org/10.1080/0951192X.2022.2027014

[14] Yao, Jun-Feng, Yong Yang, Xue-Cheng Wang, and Xiao-Peng Zhang. Systematic review of digital twin technology and applications. Visual Computing for Industry, Biomedicine, and Art, 6(1):10, 2023. https://doi.org/10.1186/s42492-023-00137-4

[15] Attaran, Mohsen, Sharmin Attaran, and Bilge Gokhan Celik. The impact of digital twins on the evolution of intelligent manufacturing and Industry 4.0. Advances in Computational Intelligence, 3(3):11, 2023. https://doi.org/10.1007/s43674-023-00058-y

[16] Liu, Shimin, Pai Zheng, and Jinsong Bao. Digital twin-based manufacturing system: a survey based on a novel reference model. Journal of Intelligent Manufacturing, 35(6):2517–2546, 2024. https://doi.org/10.1007/s10845-023-02172-7

[17] Karkaria, Vispi, Ying-Kuan Tsai, Yi-Ping Chen, and Wei Chen. An optimization-centric review on integrating artificial intelligence and digital twin technologies in manufacturing. Engineering Optimization, 57(1):161–207, 2025. https://doi.org/10.1080/0305215X.2024.2434201

[18] Wang, Yuchen, Xinheng Wang, Ang Liu, Junqing Zhang, and Jinhua Zhang. Ontology of 3D virtual modeling in digital twin: a review, analysis and thinking. Journal of Intelligent Manufacturing, 36(1):95–145, 2025. https://doi.org/10.1007/s10845-023-02246-6

[19] Lattanzi, Luca, Roberto Raffaeli, Margherita Peruzzini, and Marcello Pellicciari. Digital twin for smart manufacturing: a review of concepts towards a practical industrial implementation. International Journal of Computer Integrated Manufacturing, 34(6):567–597, 2021. https://doi.org/10.1080/0951192X.2021.1911003

[20] Elgebaly, Hamdy, Basma Elhariry, Amr Noureldin, and Doaa Stohy. Digital twin for maintenance and smart manufacturing: the mediating role of replacement maintenance in the Saudi industrial sector. Journal of Lifestyle and SDGs Review, 5(4):06107–06107, 2025. https://doi.org/10.47172/2965-730X.SDGSREVIEW.V5.N04.PE06107

[21] Alfaro-Viquez, David, Mauricio Zamora-Hernandez, Michael Fernandez-Vega, Jose Garcia-Rodriguez, and Jorge Azorin-Lopez. A comprehensive review of AI-based digital twin applications in manufacturing: integration across operator, product, and process dimensions. Electronics, 14(4):646, 2025. https://doi.org/10.3390/electronics14040646

[22] Liu, Qiang, Jiewu Leng, Douxi Yan, Ding Zhang, Lijun Wei, Ailin Yu, et al. Digital twin-based designing of the configuration, motion, control, and optimization model of a flow-type smart manufacturing system. Journal of Manufacturing Systems, 58:52–64, 2021. https://doi.org/10.1016/j.jmsy.2020.04.012

[23] Wang, Likun, Zi Wang, Kevin Gumma, Alison Turner, and Svetan Ratchev. Multi-agent cooperative swarm learning for dynamic layout optimisation of reconfigurable robotic assembly cells based on digital twin. Journal of Intelligent Manufacturing, 1–24, 2024. https://doi.org/10.1007/s10845-023-02229-7

[24] Marah, Hussein, and Moharram Challenger. Adaptive hybrid reasoning for agent-based digital twins of distributed multi-robot systems. Simulation, 100(9):931–957, 2024. https://doi.org/10.1177/00375497231226436

[25] Zhang, Rong, Jianhao Lv, Jinsong Bao, and Yu Zheng. A digital twin-driven flexible scheduling method in a human–machine collaborative workshop based on hierarchical reinforcement learning. Flexible Services and Manufacturing Journal, 35(4):1116–1138, 2023. https://doi.org/10.1007/s10696-023-09498-7

[26] Kalyani, Yogeswaranathan, and Rem Collier. The role of multi-agents in digital twin implementation: short survey. ACM Computing Surveys, 57(3):1–15, 2024. https://doi.org/10.1145/3697350

[27] Nie, Qingwei, Dunbing Tang, Haihua Zhu, and Hongwei Sun. A multi-agent and Internet of Things framework of digital twin for optimized manufacturing control. International Journal of Computer Integrated Manufacturing, 35(10–

11):1205–1226, 2022. https://doi.org/10.1080/0951192X.2021.2004619

[28] Choi, Hyekyung, Seokhwan Yu, DongHyun Lee, Sang Do Noh, Sanghoon Ji, Horim Kim, et al. Optimization of the factory layout and production flow using production-simulation-based reinforcement learning. Machines, 12(6):390, 2024. https://doi.org/10.3390/machines12060390

[29] Li, Yibing, Zhiyu Tao, Lei Wang, Baigang Du, Jun Guo, and Shibao Pang. Digital twin-based job shop anomaly detection and dynamic scheduling. Robotics and Computer-Integrated Manufacturing, 79:102443, 2023. https://doi.org/10.1016/j.rcim.2022.102443

[30] Liu, Lilan, Kai Guo, Zenggui Gao, Jiaying Li, and Jiachen Sun. Digital twin-driven adaptive scheduling for flexible job shops. Sustainability, 14(9):5340, 2022. https://doi.org/10.3390/su14095340