IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 1 July 2022, accepted 24 July 2022, date of publication 1 August 2022, date of current version 5 August 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3195905

RESEARCH ARTICLE

A Digital Twin-Based Automatic Programming Method for Adaptive Control of Manufacturing Cells

CHAO ZHANG, GUANGHUI ZHOU, (Member, IEEE), YANZHEN JING, RUI WANG, AND FENGTIAN CHANG

School of Mechanical Engineering, Xi'an Jiaotong University, Xi'an 710049, China
State Key Laboratory for Manufacturing Systems Engineering, Xi'an Jiaotong University, Xi'an 710054, China

Corresponding author: Guanghui Zhou (ghzhou@mail.xjtu.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51975463 and Grant 52105530; in part by the China Postdoctoral Science Foundation under Grant 2021M692556; and in part by the Young Talent Fund of University Association for Science and Technology in Shaanxi, China, under Grant 20210409.

ABSTRACT The booming personalized and customized demands of customers in Industry 4.0 pose a great challenge for manufacturing enterprises in terms of flexibility and responsiveness. Nowadays, many effective dynamic scheduling approaches have been proposed for manufacturing systems to quickly respond to changes in customer demands, where, however, the implementation of an automatic programming method with high control accuracy and low control delay is still challenging. The above unaddressed issue brings about a lot of labor-intensive and time-consuming manual offline programming work when adjusting the scheduling scheme to meet dynamic customer demands, resulting in limited flexibility and responsiveness in current manufacturing systems. To bridge this gap, a bi-level adaptive control architecture enabled by an automatic programming method is proposed and embedded into a digital twin manufacturing cell (DTMC). The bi-level architecture aims to automatically map an input task scheduling scheme with a batch of jobs into a group of control programs through a behavior model network and a set of event models embedded in DTMC. It also provides an adaptive program modification mechanism to quickly adapt to the dynamic adjustment of the scheduling scheme caused by the changing of customer demands or production conditions, thus equipping DTMC with strong flexibility and responsiveness. Based on the bi-level architecture, a DTMC prototype system is developed, where its application and evaluation examples demonstrate the feasibility and effectiveness of the proposed method.

INDEX TERMS Digital twin, automatic programming, adaptive control, behavior model, event model, industry 4.0.

I. INTRODUCTION

The vigorous development of new generation information technologies [1], [2] provides unprecedented opportunities for customers to participate in modern globalized manufacturing networks to transform their demands into personalized products. This promotes the transformation of production modes of industrial enterprises to mass-customization manufacturing [3], [4]. In mass-customization manufacturing, dynamic changes in personalized and customized demands of customers together with increasingly reduced product life-cycle pose a great challenge for manufacturing enterprises in terms of flexibility and responsiveness [5], [6].

Fortunately, Industry 4.0 comes with intelligent manufacturing cells to meet the booming personalized and customized demands by offering a flexible and automatic manufacturing process [7]–[9]. An intelligent manufacturing cell [10] aims to transform diverse customer demands into manufacturing

The associate editor coordinating the review of this manuscript and approving it for publication was Min Wang.

tasks with a batch of scheduled jobs [11], where each job is automatically processed by manufacturing devices controlled with a group of control programs. Nowadays, many dynamic scheduling approaches [12], [13] have been developed to produce an optimal scheduling scheme for manufacturing tasks, while providing dynamic adjustment mechanisms to quickly respond to changes in customer demands (such as delivery time and product quantity) and production conditions (such as machine fault and tool wear). Nevertheless, little research has been conducted to develop an automatic programming method that maps each of jobs in an original or adjusted scheduling scheme into a group of control programs to automatically control the operation process of the manufacturing cell. Here, high mapping accuracy and low time delay are two main challenges for the development of an automatic programming method for quickly adapting to the dynamic adjustment of the task scheduling scheme. The above unaddressed issue leads to a lot of manual offline programming work [14] when adjusting the scheduling scheme to meet dynamic customer demands, resulting in limited flexibility and responsiveness for current manufacturing cells.

Digital twin (DT) [15], defined as "an integrated multi-physics, multi-scale, probabilistic simulation of a vehicle or system that uses the best available physical models, sensor updates, fleet history, etc., to mirror the life of its flying twin", could equip manufacturing cells with powerful real-time perception, optimization and control capabilities, showing great potential for addressing the above issue in the context of Industry 4.0. Actually, DT has achieved initial success in autonomously controlling of a single manufacturing device. For example, Zhao *et al*. [16] established a DT-driven cyber physical system for the context-aware autonomously controlling of a micro-dots punching machine tool; Kaigom and Roßmann [17] introduced a concept for the development of robotic digital twins that help companies gain a competitive edge through an intelligent robotized automation; Liu *et al*. [18] introduced a DT-based framework for time-varying error prediction and compensation for the movement axis of a CNC machine tool. Comparatively speaking, DT-driven system-level control is still in its infancy, where most current related researches [19]–[21] focus on providing control suggestions based on simulation or optimization results to guide the manufacturing process. That is, autonomously controlling of a manufacturing cell with a DT-driven system-level control method is still challenging due to the lack of a latency-aware adaptive control mechanism as well as a high-accuracy automatic programming method in current DT systems.

To bridge the gap, this paper takes a digital twin manufacturing cell (DTMC) introduced in our previous works [7], [10], [22] as basis, on which a novel DT-driven adaptive control mechanism enabled by an automatic programming method is proposed and embedded into DTMC. The proposed approach takes a real-time task scheduling scheme generated by a dynamic scheduling strategy [23] as input. Then, a batch of jobs in the scheduling scheme are automatically mapped into a group of control programs through a high-fidelity simulation process empowered by a behavior model network (BMN) and a set of event models (EMs) embedded in DTMC. BMN and EMs also provide an adaptive program modification mechanism to quickly respond to changes in the scheduling scheme to meet dynamic customer demands, thus equipping DTMC with strong flexibility and responsiveness. In addition, a DTMC prototype system is implemented, where its application and evaluation results demonstrate the feasibility and effectiveness of the proposed approach.

The remainder of the paper is organized as follows. Section II proposes a novel DT-driven bi-level adaptive control architecture for DTMC. In section III, BMN is constructed based on a timed automata theory. Section IV constructs an EM-driven automatic programming model for DTMC. Section V discusses a thus implemented prototype system to demonstrate the effectiveness of the approach. The conclusions and future works are found in section VI.

II. DT-DRIVEN BI-LEVEL ADAPTIVE CONTROL ARCHITECTURE

This section takes DTMC as basis, on which a novel DT-driven bi-level adaptive control architecture enabled by an automatic programming method is proposed. The proposed architecture aims to automatically map a batch of scheduled jobs generated by a dynamic scheduling strategy [23] into a group of executable control programs for autonomously controlling of DTMC. In addition, the architecture also provides an adaptive program modification mechanism to quickly adapt to dynamic changes in the scheduling scheme during the job execution process.

A. BRIEF INTRODUCTION OF DTMC

In the context of Industry 4.0, we introduced a new kind of intelligent manufacturing systems – DTMC [7], [10], [22]. DTMC offers powerful learning and cognitive capacities for autonomous manufacturing enabled by three functional layers including perception layer (PL), optimal-state control layer (OSCL) and service layer (SL), as shown in Fig. 1.

PL collects real-time data (such as device status, tool wear condition, job execution progress) from physical space with a sensor network, which is published to OSCL for further analysis. PL also subscribes control programs from OSCL through smart gateways for autonomously controlling of DTMC.

OSCL aims to keep physical space operating at its optimizal state with the cooperation of data space, virtual space and knowledge space. Specifically, data space is responsible for real-time data parsing and standardization through an OPC-UA information model embedded in driver agent, which also provides an unified interface in access library for virtual space and knowledge space to access data [24]. Virtual space contains a set of DT models, such as CNC machine tool DT, robotic DT, etc., which provide a high-fidelity simulation capacity to understand the current or future performance of physical space. Knowledge space acts as the brain of DTMC to handle various manufacturing problems in physical space and virtual space based on the dynamic knowledge base and knowledge models (such as a dynamic scheduling model).

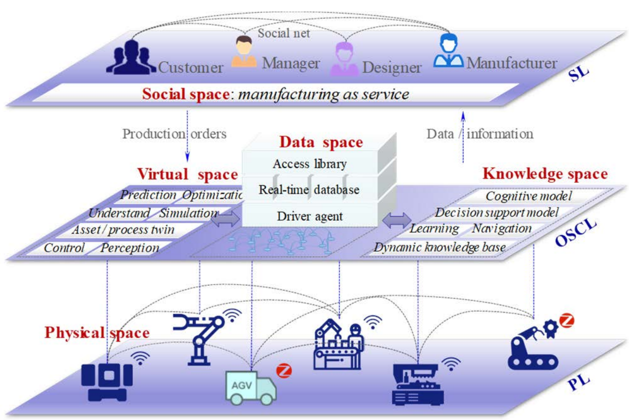

FIGURE 1. A reference framework for DTMC [22].

SL packages manufacturing capacities and resources as services to serve customers, managers, designers, etc., in social space [25]. SL integrates a variety of mature service systems, such as the customer relation management system and advanced production scheduling system, which bridge the gap between the supply of DTMC and diverse demands of customers.

DTMC could be one of the popular solutions for the construction of a DT-enhanced new-generation intelligent manufacturing workshop in mass-customization manufacturing. However, DTMC is more of a concept than a mature system. The implementation of DTMC is still made difficult due to the lack of a latency-aware adaptive control mechanism as well as an automatic programming method.

B. DESIGN OF BI-LEVEL ARCHITECTURE

To bridge the gap, a DT driven bi-level adaptive control mechanism enabled by an automatic programming method is proposed for DTMC. To this end, a BMN is incorporated into the virtual space of DTMC, which maps a set of jobs in a scheduling scheme into a group of discrete manufacturing events through a high-fidelity simulation process. On that basis, an EM-driven automatic programming method is proposed to transfer each event into a group of control programs for autonomously controlling of DTMC.

Specifically, given a task scheduling scheme $S = \{J_1, J_2, \ldots, J_n\}$ containing n jobs scheduled by the advanced production scheduling system in social space, the bi-level adaptive control mechanism could be defined by a discontinuous nonlinear mapping:

$$C(t) = E\left(B\left(\bigcup_{i=1}^{n} J_i + \delta_{ext}(t)\right) + \delta_{int}(t)\right) \tag{1}$$

where $C(t)$ is a discontinuous nonlinear mapping algorithm; $B(\cdot)$ in (1) represents an external mapping that transfers $J_i \in S$ into a trace containing a set of verified events expressed as $<e_1[t_1], e_2[t_2], \ldots, e_q[t_q]>$ through BMN; $E(\cdot)$ in (1) represents an internal mapping that transfers all verified events in the trace into a group of control programs executed in physical space of DTMC through an event-driven automatic programming model; $\delta_{ext}(t)$ is the external deviation of job schedules caused by the changes in customer demands and production conditions at time t; $\delta_{int}(t)$ is the internal deviation that only influences the execution time and sequence of manufacturing events at time t. The real-time deviation of $\delta_{ext}(t)$ and $\delta_{int}(t)$ could be derived by a dynamic scheduling model [23].

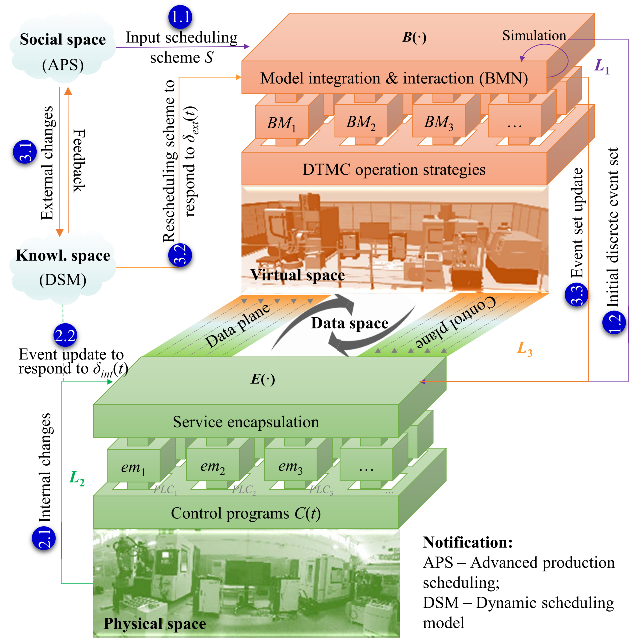

FIGURE 2. Bi-level adaptive control architecture for DTMC.

The semantics of the discontinuous nonlinear mapping in (1) could be expressed via a bi-level architecture as shown in Fig. 2. The top level corresponds to an external mapping $B(\cdot)$ carried out by a BMN that consists of a set of behavior models (BMs). BMN could automatically generate a trace containing a set of verified discrete events for the input S through a logic simulation. The bottom level corresponds to an internal mapping $E(\cdot)$ carried out by an EM-driven automatic programming model. In this model, each manufacturing device in physical space acts as an actuator controlled by a programmable logic controller (PLC). To this end, an event pool containing a set of EMs is constructed to establish correspondences between an event and a group of PLC programs. In addition, EMs are encapsulated as services to subscribe events in a trace and turn them into PLC programs to logically control manufacturing devices in physical space. With the in-depth integration of cyber-physical spaces in the bi-level architecture based on DT, three control loops including the automatically programming loop l_1, internal change control loop l_2 and external change control loop l_3 are formed to adaptively handle external or internal changes.

Specifically, l_1 takes a task schedule scheme as input and generates an initial discrete event set through $B(\cdot)$ enabled by BMN, which is further transformed to physical via control plane in data space. The event set is automatically resolved and mapped into control programs through $E(\cdot)$ enabled by an EM-driven automatic programming method to guide the operation of DTMC. During manufacturing process, l_2 aims to update the execution time and sequence of events to quickly respond to the internal deviation $\delta_{int}(t)$ by continuously simulating the future state of tasks to seek optimal event execution time and sequence based on the current task execution state with BMN. l_3 aims to quickly generate a new discrete event set based on the rescheduling scheme to respond to the external deviation $\delta_{ext}(t)$, which further updates the control programs in millisecond level for autonomously controlling of DTMC.

With the above observations, the following key enabling technologies of the architecture could be summarized, namely the behavior modeling method for BMN construction and EM-driven automatic programming model for autonomously controlling of DTMC.

III. BEHAVIOR MODELING METHOD

DTMC is a typical discrete event dynamic system that finishes manufacturing tasks with a series of discrete events encoded in control programs. Behavior modeling aims to fully characterize the operation logic of manufacturing devices in physical space of DTMC digitally, through which a task scheduling scheme with a set of jobs could be automatically mapped into a series of verified discrete manufacturing events. Besides, timed automata (TA) [26] have been a popular theory widely used in modeling and analyzing characteristics and behaviors of discrete event dynamic systems. TA enjoys significant advantages in high modeling accuracy, strong interpretability/ verifiability, and fast responding speed. Consequently, TA is utilized in this section for behavior modeling of manufacturing devices according to the operation strategies of DTMC.

A. DEFINING OPERATION STRATEGIES FOR DTMC

Operation strategies serve as the basis for behavior modeling of DTMC, which are defined by the following interactive mechanisms and conflict resolving rules: 1) A single job (corresponding to a workpiece in workshop) exists independently in a manufacturing device, which is not allowed to be occupied by multiple manufacturing devices at the same time; 2) Job processed on a manufacturing device needs to be completed within a limited period of time defined in the scheduling scheme; 3) Multiple jobs are processed with a specific sequence, not disorderly; 4) Each manufacturing device handles jobs with First Come First Serve (FCFS) rule, which means jobs are preferentially processed according to the order of arrival time; 5) Shortest Job First (SJF) rule would be adopted when a manufacturing device receives multiple jobs at the same time and; 6) Jobs will be randomly selected for processing when a manufacturing device receives multiple jobs with the same processing and arrival time.

Notice that the above strategies used in this paper are flexible to be revised or replaced by other strategies to adapt to different types of DTMC.

B. TA-BASED BEHAVIOR MODELLING

Based on the operation strategies, a TA-based behavior modelling method is proposed for the construction of BMN. To begin with, we define TA-based concepts for DTMC.

Definition I: Attribute constraints qualitatively represent the capacity and availability of manufacturing resources in physical space of DTMC, which determine whether a mapping between a job in the task and events executed by several manufacturing resources could be approved. Given a finite set of attributes $A = \{a_1, a_2, \ldots, a_m\}$, the set $\varphi(A)$ of all possible attribute constraints over A is defined by:

$$\varphi(A) = \{a_1 \sim \upsilon_{a1}, a_2 \sim \upsilon_{a2}, \ldots, a_m \sim \upsilon_{am}\} \tag{2}$$

where $\sim \in \{=, \geq, >, \leq, <\}$; $a_m \sim \upsilon_{am}$ indicates indicates the m-th attribute of DTMC satisfies the constraint defined by υ_{am}. For example, if the highest turning precision of a lathe is 0.01mm, its attribute constraint – maximum turning precision is defined as $a_1 = 0.01$mm.

Definition II: Clock constraints indicate the processing period for each of jobs based on its scheduling scheme. Given a finite set of clocks $C = \{c_1, c_2, \ldots, c_n\}$, the set $\varphi(C)$ of all possible clock constraints over C is defined by the grammar:

$$\varphi(C) = \{c_1 \sim \tau_{c1}, c_2 \sim \tau_{c2}, \ldots, c_n \sim \tau_{cn}\} \tag{3}$$

where $\sim \in \{=, \geq, >, \leq, <\}$; $c_n \sim \tau_{cn}$ indicates the n-th clock for processing a job satisfies the time constraint defined by τ_{cn}. For example, if the processing time of a job is 1min, the clock constraint for that job is defined as $c_1 = 1$min.

Based on Definition I and II, we employ attribute and clock constraints $\varphi(C) = \varphi(A) \cup \varphi(C)$ to represent all possible constraints for DTMC to complete a batch of jobs in the task scheduling scheme.

Definition III: Locations & status represent a finite set of relative spatial positions or operation status of manufacturing resources in DTMC, which is defined as:

$$LS = \{ls_1, ls_2, \ldots, ls_i\} \tag{4}$$

where ls_i indicates a specific location or status for a manufacturing resource. For example, LS_1 represents all possible end locations of AGV in physical space, which is defined as $LS_1 = \{ls_{buffer1} = (1, 5), \ldots, ls_{bufferi} = (5, 10)\}$.

Definition IV: Actions refer to a finite set of operations performed by DTMC to finish jobs, which are defined as:

$$\Sigma = \left\{\alpha_1, \alpha_2, \ldots, \alpha_j\right\} \tag{5}$$

where α_j indicates a specific operation of a manufacturing resource. For example, the startup, waiting, door open, processing and shutdown of a machine tool.

Definition V: The invariant clock constraint is a subset of clock constraints, which indicates the condition for one of two types of state transitions of TA that can occur in DTMC, namely a time transition where the location or status stays the same while the clock valuation τ advances $\delta \in \mathbb{R}^{\geq 0}$ units to the valuations $\tau + \delta$. Given a finite set of invariant clock constraint $I(ls)$, a time transition is defined by the grammar:

$$\langle ls\ ,\ \tau\rangle \xrightarrow{\delta} \langle ls\ ,\ \tau+\delta\rangle \text{ if } \forall 0 \leq \delta' \leq \delta, \tau+\delta' \models I(ls) \quad (6)$$

where the semantics of (6) is that, for example, the location of a job stays invariant when it is processed in a machine tool until finished or specified time constrain reached.

Definition VI: The edge of TA $e = (ls, \alpha, g, \lambda, ls^{'})$ indicates the condition of another type of state transition - the action transition where location or status of a manufacturing resource would be changed from ls to $ls^{'}$, if ν satisfies the action command α and guard g containing a set of constraints defined in $\varphi(\mathbb{C})$, while the clocks in λ are reset to 0 and the invariant of $ls^{'}$ must be satisfied after the clocks are reset. Given a finite set of edges E, an action transition is defined by the grammar:

$$\langle ls\ ,\ \tau\rangle \xrightarrow{\alpha,g,\lambda} \langle ls'\ ,\ \tau\rangle \text{ if} \tau \models g, \tau' \models I(ls') \quad (7)$$

where the semantics of (7) is that, for example, the location of a job could be changed from ls to $ls^{'}$ when a logistics action α is performed and its transition time satisfies the constraints in guard g defined in the schedule plan.

Based on Definition I-VI, a TA-based behavior model for a manufacturing resource is defined by a tuple:

$$BM = \{LS, ls, \Sigma, \mathbb{C}, I, E\} \quad (8)$$

where LS is a finite set of locations & status that represent all possible operation positions or status of manufacturing resources in DTMC; ls is the initial location & status; Σ is a finite set of actions; $\mathbb{C}$ is a finite set of attributes and clocks; I is a finite set of invariant clock constraints; $E \subseteq LS \times \Sigma \times \varphi(\mathbb{C}) \times 2^{\mathbb{C}} \times LS$ is a finite set of edges.

C. BM INTEGRATION AND INTERACTION

A set of BMs could be integrated into BMN, thus equipping DTMC with a high-fidelity operation logic simulation capacity for the automatic generation of discrete manufacturing events based on the input scheduling scheme.

Specifically, let $BM = \{BM_1, BM_2, \ldots, BM_N\}$ be a set of BMs representing each of all manufacturing resources in DTMC, where $BM_i = \{LS_i, ls_i, \Sigma_i, \mathbb{C}_i, I_i, E_i\}$. BMs could be integrated into BMN through the dot product of BM_i and BM_j based on TA theory, which is calculated as:

$$\begin{aligned} BMN &= \{LS, ls_0, \Sigma, \mathbb{C}, I, E\} \\ &= BM_1 || BM_2 || \ldots || BM_N \\ LS &= \bigcup\nolimits_{i=1}^{N} LS_i \\ ls_0 &= \left\langle l_{10}, l_{20}, \ldots, l_{N0} \right\rangle \\ \Sigma &= \bigcup\nolimits_{i=1}^{N} \Sigma_i \\ \mathbb{C} &= \bigcup\nolimits_{i=1}^{N} \mathbb{C}_i \\ I\left(\left\langle LS_1, LS_2, \ldots, LS_N \right\rangle\right) &= \wedge_{i=1}^{N} I_i(LS_i) \\ E &= \bigcup\nolimits_{i=1}^{N} E_i \end{aligned} \quad (9)$$

BMN takes a task scheduling/rescheduling scheme as input and generates a set of verified actions as discrete manufacturing events in a trace that could be performed by manufacturing devices in the physical space of DTMC to finish the task. BMN is operated on four atomic rules widely used in TA, including the sequence rule, spilt rule, choice rule and if-then-else rule. Let T_i and T_j be the i-th and j-th transitions, respectively, in BMN, the atomic rules are defined as: 1) the sequence rule indicates that T_i could be performed only when T_j is finished; 2) the spilt rule indicates that T_i and T_j could be performed concurrently without barrier synchronization; 3) the choice rule indicates that T_i could be performed only when T_j does not start, and vice versa; 4) if-then-else rule indicates that T_i could be executed when if condition holds, and T_j could be executed when if condition does not hold. In addition, atomic rules are the concrete embodiment of operation strategies of DTMC in BMN, where the selection of atomic rules is determined by the input scheduling scheme and operation strategies.

IV. EM-DRIVEN AUTOMATIC PROGRAMMING MODEL

This section introduces an automatic programming model enabled by a set of EMs in the event pool and an event-driven hybrid control network, which could dynamically transfer the discrete manufacturing events generated by BMN into a group of control programs.

A. DESIGN OF EVENT MODEL AND EVENT POOL

A set of events in a trace are transferred into control programs with logic self-consistency and enforceability through EMs. EM is defined based on the discrete event system specification [27]:

$$em = \{O, EM_{ext}, S_{int}, S_{out}, C_{out}, \psi\} \quad (10)$$

where O refers to an objective for executing an event, such as a robot or a machine tool; EM_{ext} is a set of external events that would influence the execution of the current event; S_{int} indicates a group of internal states of manufacturing resources in DTMC, which are the prerequisites for the execution of the current event; S_{out} is a group of output states of manufacturing resources after the execution of em, which may influence the execution of other events; C_{out} is a group of control programs that could control the operation process of the objective when executing the event; ψ is an execution algorithm that enables EM with the capacity of logic self-consistency and enforceability.

As shown in Table 1, ψ evaluates the influence of external events for the execution of the current event based on S_{int}. ψ further maps the event to the executable control programs to guide the operation of DTMC, while outputting a group of states influencing the execution of other events.

Based on EM, an event pool is defined as a collection of a finite set of EMs, that is $EP = \{em_1, em_2, \ldots, em_k\}$, where em_k refers to the k-th EM in the event pool EP. The event pool records all possible events executed in DTMC, which serves as the prior knowledge for bridging the gap between events generated by BMN and control programs executed in physical space of DTMC.

TABLE 1. **EM execution algorithm.**

Algorithm I: EM execution algorithm ψ
Input: *em* execution command, EM_{ext}, S_{int}
Output: S_{out} and C_{out}
For each $em_i \in EM_{ext}$
$S \leftarrow em_i.S_{out}$
End for
If $S \cap S_{int} = S_{int}$
Then
For each $s_i \in S_{out}$
s_i.*setStateController*
End for
For each $c_j \in C_{out}$
c_j.*setController*
End for
Return S_{out}, C_{out}
Else break
End if

B. EVENT-DRIVEN HYBRID CONTROL NETWORK

Traditional centralized control mechanism may significantly influence the real-time performance of DTMC caused by stochastic errors or latency, especially when the control programs are frequently modified. Distributed control mechanism may address this issue through a decentralized control network, which, however, may not guarantee the control stability and robustness. Hence, an event-driven hybrid control network is proposed, which takes advantage of both centralized and decentralized control mechanisms. As shown in Fig. 3, a set of EMs in the event pool are encapsulated as a service based on their topic related to production process, such as warehousing, logistics, processing, inspection, etc. Here, each service acts as a leader and a set of events contained in that service act as the followers in the hybrid control network. Each leader and its followers consist of a centralized control network triggered by a trace containing a set of discrete manufacturing events generated by BMN. Each of followers with its controller in the device end constitutes a distributed control network with logic self-consistency enabled by an execution algorithm ψ. The entire hybrid control network is operated based on a publish-subscribe architecture that equips the hybrid control network with greater network scalability and a more dynamic network topology.

Specifically, a service is defined as a tuple:

$$s = \{t_l, EM_f\} \tag{11}$$

where t_l represents the l-th topic classified by the production process, and EM_f refers to a event set included in this topic.

As shown in Fig. 3, The entire hybrid control network subscribes a verified trace $\{<e_1[t_1], e_2[t_2], \ldots, e_q[t_q]>, cst\}$ generated by BMN as input. Here, $<e_1[t_1], e_2[t_2], \ldots, e_q[t_q]>$ is an event set with total q discrete manufacturing events, where each event $e_q[t_q]$ is labelled with its event identifier e_q and execution time t_q. cst is a constraint set for the execution of events, which is defined by the relationships between each event, including sequence, spilt, choice and if-then-else relationships. The hybrid control network could be triggered only when the events and constraints in the trace are the subsets of EP and constraint set defined in the publish-subscribe architecture, respectively. Then, the hybrid control network generates a group of control programs that are published to the corresponding controllers to control the production process of DTMC. With the above observations, the publish-subscribe architecture is defined as:

$$PSA = \{EP, CST, Sub, Pub, \zeta\} \tag{12}$$

where EP is an event pool designed for a specific DTMC; CST defines a constraint set to provide guidance for the execution of events; $Sub = t_l(e_i[t_i], em_i)$ is a subscription interface, where em_i is the corresponding EM for event $e_i[t_i]$ with t_l as topic; $Pub = t_l(S_{outi}, C_{outi})$ is a publish interface, which publishes control programs to PLC deployed at device end after the execution of em_i; ζ represents the operation strategies of DTMC defined in Section III.

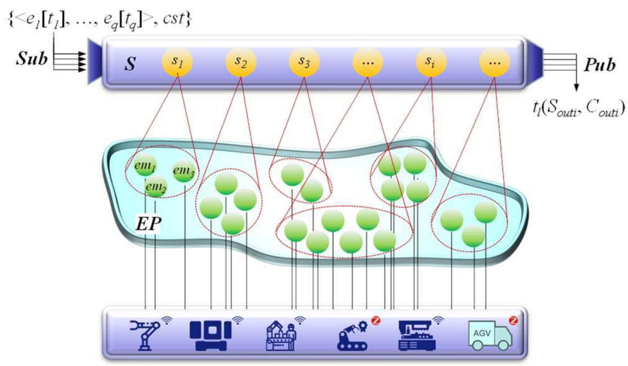

FIGURE 3. **Event-driven hybrid control network.**

V. PROTOTYPE SYSTEM IMPLEMENTATION

This section develops a prototype system named BE-DTMC, where its application and evaluation results demonstrate the feasibility and effectiveness of the proposed approach.

A. DEVELOPMENT ENVIRONMENT

As shown in Fig. 4, a hardware and software integrated development environment is constructed based on the reference framework of DTMC, which serves as the basis for BE-DTMC implementation. In addtion, an advanced production scheduling system and a dynamic scheduling model [23] are embedded into social space and knowledge space, respectively, to generate a scheduling/ rescheduling scheme that is taken as the input for autonomously controlling of BE-DTMC. The onstruction details of the development environment could be found in [7], [10], [22]. Its operation mechanism is introduced as follows.

FIGURE 4. A typical DTMC and its operation environment.

PL perceives the real-time status of the physical manufacturing cell by a sensor network, where manufacturing resources, including two robots, a CNC lathe, a CNC milling machine, AGV, a warehouse and several IoT devices, are deployed for automatic task execution. The perceived data and real-time control programs are published and subscribed by smart gateways, respectively.

OSCL makes physical manufacturing cell operate at its optimal state with the cooperation of data space, virtual space, and knowledge space. Data space collects real-time data from physical manufacturing cell, which is further stored in real-time database and accessed by other spaces through an OPC-UA interface. Virtual space contains virtual manufacturing resources, including virtual manufacturing cell in global view, and DT robot, DT lathe, DT milling machine in device view, which could simulate and understand the performance of physical manufacturing cell. Knowledge space integrates dynamic knowledge base and knowledge models (for example, the dynamic scheduling model) that act as the brain of DTMC to handle various manufacturing problems in physical space or virtual space. In addition, real-time database, device twins and partial knowledge models are deployed near devices to handle time-sensitive control issues. Dynamic knowledge base, knowledge models and virtual manufacturing cell are deployed at remote servers to handle computationally intensive global optimization issues.

SL packages manufacturing capacities and resources as services in social space to handle personalized and customized demands of customers, while transferring them into manufacturing task scheduling scheme processed adaptively by BE-DTMC.

B. BE-DTMC IMPLEMENTATION

Based on the development environment, BMN and EMs are constructed with the proposed approach. As shown in Fig. 5, BMN and EMs are embedded into DTMC to formalize a BE-DTMC prototype system, which could effectively support the bi-level adaptive control of physical manufacturing cell. In addition, the data and control flow between BMN, EM and physical manufacturing cell is enabled by a publish-subscribe interface.

FIGURE 5. BE-DTMC prototype system.

As shown in the top of Fig. 5, BM for each of manufacturing resources in physical manufacturing cell is constructed based on TA. A set of BMs are integrated into BMN that equips BE-DTMC with high-fidelity operation logic simulation capacity. BMN are implemented in UPPAAL, which provides an integrated tool environment for modeling, validation and verification of the manufacturing cell modeled as networks of TA through the following steps. Firstly, the behaviour of BE-DTMC for task execution is described as a network of TAs extended with clock and data variables based on a non-deterministic guarded command language in UPPAAL. Then, possible dynamic executions of BMN are examined via a simulator in UPPAAL for the quick detection and repair of faults in early modeling stage. Finally, the exhaustive dynamic behavior of BMN is checked in terms of its invariant and reachability properties by exploring the state-space of BMN via a model-checker in UPPAAL. Through the above strategies, a complete and effective BMN with the invariant and reachability properties checking mechanism is constructed and embedded in the virtual manufacturing cell.

As shown in the right of Fig. 5, a three-layer event-driven hybrid control network is constructed. Here, the bottom layer contains several physical controllers that directly control the corresponding devices. Middle layer defines thirteen basic events through EM, such as out of stock, warehousing, RFID scan, etc. The basic events are further encapsulated as five services based on their topic related to task execution process, including warehousing, logistics, loading/ unloading of workpieces, turning, and milling represented by s1∼s5, respectively. In addition, each service with the events (marked with the same color) consists of a centralized control network triggered by upper-level events generated by BMN; each event with its linked controller in PL constitute a distributed control network for latency-aware device control.

Based on the constructed BMN and EMs, a task-driven adaptive control flow enabled by an automatic programming method could be carried out with the following steps. Firstly, BE-DTMC takes a task scheduling scheme as input and generates a trace containing a set of verified events through BMN. Then, the trace is resolved and mapped into PLC programs through EMs for autonomously controlling of BE-DTMC. During the operation process, BE-DTMC simulates the future state of tasks to seek optimal trace according to the current task execution state, thereby quickly responding to the internal changes. In addition, BE-DTMC could quickly generate a new trace when the scheduling scheme is adjusted, which is further mapped into PLC programs through EMs to control the real-time operation state of BE-DTMC.

C. APPLICATION EXAMPLE OF BE-DTMC

Fig. 6 shows a typical application example for task-driven adaptive control of BE-DTMC during the production process of a batch of disc-type parts.

Speficially, Fig. 6 (a) illustrates a task for disc-type part processing, which includes two flange parts, two end cup parts and one impeller. The scheduling scheme of the task generated by the advanced production shceduling system is

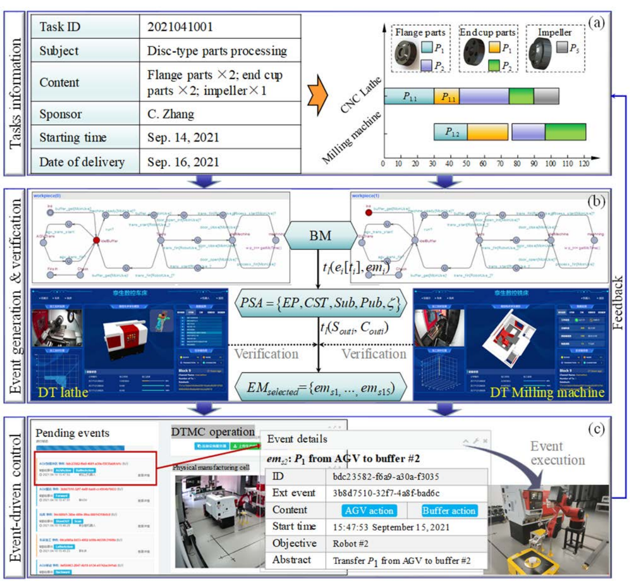

FIGURE 6. An event-driven adaptive control example for BE-DTMC.

as shown in the right of Fig. 6 (a), which is taken as the input for BE-DTMC. As shown in Fig. 6 (b), based on the scheduling scheme, a trace containing a set of events labeled with timestamp is first generated through a simulation and verification process of BMN. Secondly, each of events in the trace is subscribed by a service defined in the hybrid control network based on its topic. Thirdly, each event in a service is mapped into a group of PLC programs through its corresponding EM. Finally, PLC programs along with the linked EMs are verified by device twins in terms of the capacity and availability of physical devices. In addition, the verified PLC programs are published to the corresponding controllers for the logic control of BE-DTMC.

D. DISCUSSION

This section evaluates the performance of BE-DTMC in terms of control accuracy and control delay. Then contributions and challenges ahead of the paper are discussed.

Control accuracy refers to the matching degree between the control programs generated by BMN and EMs, and the actual programs used to complete the same event in physical space of BE-DTMC. For accuracy evaluation purpose, 50 groups of events for disc-type part processing are employed, where 50 groups of PLC programs are generated through the EM execution algorithm ψ. By comparing with the pre-fixed programs, an average control accuracy of 100% are obtained. This may attribute to the triple verification mechanism provided by BE-DTMC, namely invariant and reachability properties checking via model-checker in BMN, logic self-consistent checking via EMs, as well as capacity and availability checking of manufacturing devices via device twins, which ensures the reliability and stability of BE-DTMC. Control delay is represented by time difference between the input timestamp of a task and device action timestamp for finishing that task. Control delay is divided into three subclasses, including global control delay (t_{gcd}), control program generation delay (t_{cpgd}), and device responding delay (t_{drd}). Here, t_{gcd} is calculated by the difference between task input timestamp in social space and device responding timestamp in physical space. t_{cpgd} is calculated by the difference between task arrival timestamp in BMN and event publish timestamp in EMs. t_{drd} is calculated by the difference between event publish timestamp and device responding timestamp. As shown in Fig. 7, 70 groups of tasks are tested for control delay evaluation purpose. Here, the average global control delay, control program generation delay and device responding delay are 403.39ms, 2.47ms and 242.34ms, respectively, which are suitable for industrial applications.

This paper explores the key aspects of a DT-based automatic programming method for adaptive control of DTMC. However, there are still some challenges needed to be addressed in near future. According to the experimental results, t_{gcd} and t_{drd} are relatively high, which may be caused by the transmission delay. Fortunately, multi-access edge computing [28], characterized by ultra-low latency and

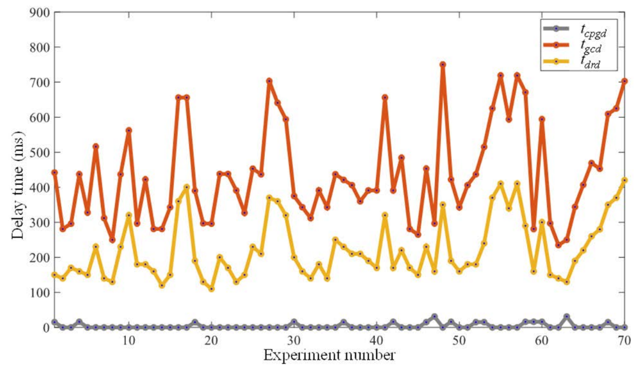

FIGURE 7. Evaluation of the control delay of BE-DTMC.

high bandwidth, might be introduced in DTMC to significantly improve its transmission efficiency. In addition, the increase in the number of manufacturing resources may affect scale and computational complexity of BMN and EMs, where a more effective edge-cloud collaboration mechanism should be established to optimize the real-time performance of DTMC.

VI. CONCLUSION

This paper proposes a novel DT-based automatic programming method for adaptive control of DTMC which could automatically map an input task scheduling scheme with a batch of jobs into a group of control programs through BMN and EM embedded in DTMC. In addition, the proposed approach also provides an adaptive program modification mechanism to quickly adapt to the adjustment of the scheduling scheme caused by the changing of customer demands or production conditions, thus equipping DTMC with strong flexibility and responsiveness. Based on the experimental results obtained in this paper, the following contributions of the paper could be summarized.

1) To bridge the gap between limited flexibility and responsiveness in current manufacturing cells and dynamically changeable customer demands in mass-customization manufacturing, a novel bi-level adaptive control architecture enabled by an automatic programming method is proposed for autonomously controlling of DTMC. To our best knowledge, this is the first time that combines DT with automatic programming to construct an intelligent manufacturing cell with strong flexibility and responsiveness.
2) A behavior modeling method for BMN construction and an EM-driven automatic programming model for autonomously controlling of DTMC are introduced as the key enabling technologies for the bi-level architecture. Here, a BMN is incorporated into the virtual space of DTMC to generate a set of discrete manufacturing events for finishing jobs in an input scheduling scheme. On that basis, a set of EMs are performed to map each event into a group of control programs for autonomously controlling of DTMC. In addition, BMN and EM also provide a triple verification mechanism, including invariant and reachability properties checking via BMN, logic self-consistent checking via EM, and capacity & availability checking of manufacturing devices via device twins, which guarantees the control accuracy and robustness of DTMC.
3) A BE-DTMC prototype system is implemented, which could provide an insight into the industrial implementation of an intelligent manufacturing cell towards Industry 4.0. Its application and evaluation results show the superiority of the proposed approach with a very high control accuracy performance within the reasonable delay at the millisecond level.

Potential medium-term future studies related to this paper are as follows. We plan to combine multi-access edge computing with digital twins to significantly improve the transmission efficiency of DTMC. In addition, considering that the increase of manufacturing resources may affect the scale and computational complexity of DTMC, we plan to establish a more effective edge-cloud collaboration mechanism to optimize its real-time performance.

REFERENCES

[1] C.-C. Chen, M.-H. Hung, P.-Y. Li, Y.-C. Lin, Y.-Y. Liu, and F.-T. Cheng, "A novel automated construction scheme for efficiently developing cloud manufacturing services," *IEEE Robot. Autom. Lett.*, vol. 3, no. 3, pp. 1378–1385, Jul. 2018.
[2] S. Jeong, W. Na, J. Kim, and S. Cho, "Internet of Things for smart manufacturing system: Trust issues in resource allocation," *IEEE Internet Things J.*, vol. 5, no. 6, pp. 4418–4427, Dec. 2018.
[3] T. Li, T. He, Z. Wang, and Y. Zhang, "An approach to IoT service optimal composition for mass customization on cloud manufacturing," *IEEE Access*, vol. 6, pp. 50572–50586, 2018.
[4] C. Jiang and J. Wan, "A thing-edge-cloud collaborative computing decision-making method for personalized customization production," *IEEE Access*, vol. 9, pp. 10962–10973, 2021.
[5] D. Guo, S. Ling, H. Li, D. Ao, T. Zhang, Y. Rong, and G. Q. Huang, "A framework for personalized production based on digital twin, blockchain and additive manufacturing in the context of Industry 4.0," in *Proc. IEEE 16th Int. Conf. Autom. Sci. Eng. (CASE)*, Aug. 2020, pp. 1181–1186.
[6] F. Tao and Q. Qi, "New IT driven service-oriented smart manufacturing: Framework and characteristics," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 49, no. 1, pp. 81–91, Jan. 2019.
[7] C. Zhang, G. Zhou, J. He, Z. Li, and W. Cheng, "A data- and knowledge-driven framework for digital twin manufacturing cell," in *Proc. CIRP*, vol. 83, Jan. 2019, pp. 345–350.
[8] W. Dai, C. Pang, V. Vyatkin, J. H. Christensen, and X. Guan, "Discrete-event-based deterministic execution semantics with timestamps for industrial cyber-physical systems," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 3, pp. 851–862, Mar. 2020.
[9] J. Leng, D. Yan, Q. Liu, K. Xu, J. Zhao, R. Shi, L. Wei, D. Zhang, and X. Chen, "ManuChain: Combining permissioned blockchain with a holistic optimization model as bi-level intelligence for smart manufacturing," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 50, no. 1, pp. 182–192, Jan. 2020.
[10] G. Zhou, C. Zhang, Z. Li, K. Ding, and C. Wang, "Knowledge-driven digital twin manufacturing cell towards intelligent manufacturing," *Int. J. Prod. Res.*, vol. 58, no. 4, pp. 1034–1051, Feb. 2020.
[11] C. Wang, G. Zhou, and Z. Zhu, "Service perspective based production control system for smart job shop under Industry 4.0," *Robot. Comput.-Integr. Manuf.*, vol. 65, Oct. 2020, Art. no. 101954.
[12] J. Zhao, T. Wang, W. Pedrycz, and W. Wang, "Granular prediction and dynamic scheduling based on adaptive dynamic programming for the blast furnace gas system," *IEEE Trans. Cybern.*, vol. 51, no. 4, pp. 2201–2214, Apr. 2021.

[13] Y. Zhao, Y. Wang, Y. Tan, J. Zhang, and H. Yu, "Dynamic jobshop scheduling algorithm based on deep Q network," *IEEE Access*, vol. 9, pp. 122995–123011, 2021.

[14] M. A. Sehr, M. Lohstroh, M. Weber, I. Ugalde, M. Witte, J. Neidig, S. Hoeme, M. Niknami, and E. A. Lee, "Programmable logic controllers in the context of Industry 4.0," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3523–3533, May 2021.

[15] R. Rolle, V. Martucci, and E. Godoy, "Architecture for digital twin implementation focusing on Industry 4.0," *IEEE Latin Amer. Trans.*, vol. 18, no. 5, pp. 889–898, May 2020.

[16] R. Zhao, D. Yan, Q. Liu, J. Leng, J. Wan, X. Chen, and X. Zhang, "Digital twin-driven cyber-physical system for autonomously controlling of micro punching system," *IEEE Access*, vol. 7, pp. 9459–9469, 2019.

[17] E. Guiffo Kaigom and J. Roßmann, "Value-driven robotic digital twins in cyber–physical applications," *IEEE Trans. Ind. Informat.*, vol. 17, no. 5, pp. 3609–3619, May 2021.

[18] K. Liu, L. Song, W. Han, Y. Cui, and Y. Wang, "Time-varying error prediction and compensation for movement axis of CNC machine tool based on digital twin," *IEEE Trans. Ind. Informat.*, vol. 18, no. 1, pp. 109–118, Jan. 2022.

[19] C. Gehrmann and M. Gunnarsson, "A digital twin based industrial automation and control system security architecture," *IEEE Trans. Ind. Informat.*, vol. 16, no. 1, pp. 669–680, Jan. 2020.

[20] J. Li, G. Zhou, and C. Zhang, "A twin data and knowledge-driven intelligent process planning framework of aviation parts," *Int. J. Prod. Res.*, to be published, doi: 10.1080/00207543.2021.1951869.

[21] D. Ding, Q.-L. Han, X. Ge, and J. Wang, "Secure state estimation and control of cyber-physical systems: A survey," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 51, no. 1, pp. 176–190, Jan. 2021.

[22] C. Zhang, G. Zhou, H. Li, and Y. Cao, "Manufacturing blockchain of things for the configuration of a data- and knowledge-driven digital twin manufacturing cell," *IEEE Internet Things J.*, vol. 7, no. 12, pp. 11884–11894, Dec. 2020.

[23] G. Zhou, Z. Chen, C. Zhang, and F. Chang, "An adaptive ensemble deep forest based dynamic scheduling strategy for low carbon flexible job shop under recessive disturbance," *J. Cleaner Prod.*, vol. 337, Feb. 2022, Art. no. 130541.

[24] A. Angrish, B. Starly, Y.-S. Lee, and P. H. Cohen, "A flexible data schema and system architecture for the virtualization of manufacturing machines (VMM)," *J. Manuf. Syst.*, vol. 45, pp. 236–247, Oct. 2017.

[25] K. Ding, P. Jiang, and S. Su, "RFID-enabled social manufacturing system for inter-enterprise monitoring and dispatching of integrated production and transportation tasks," *Robot. Comput.-Integr. Manuf.*, vol. 49, pp. 120–133, Feb. 2018.

[26] B. Amel and M. Ramedane, "From OWL-S to timed automata network: Operational semantic," *Proc. Comput. Sci.*, vol. 83, pp. 409–416, Jan. 2016.

[27] B. P. Zeigler, "Discrete event system specification framework for self-improving healthcare service systems," *IEEE Syst. J.*, vol. 12, no. 1, pp. 196–207, Mar. 2018.

[28] T. Taleb, K. Samdanis, B. Mada, H. Flinck, S. Dutta, and D. Sabella, "On multi-access edge computing: A survey of the emerging 5G network edge cloud architecture and orchestration," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 3, pp. 1657–1681, 3rd Quart., 2017.

CHAO ZHANG received the B.E. degree in mechanical engineering and automation from Sichuan University, Chengdu, China, in 2015, and the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, in 2020.

He is currently an Assistant Professor with the School of Mechanical Engineering, Xi'an Jiaotong University. He has authored two monographs, 30 journals and conference papers, and 15 inventions. His research interests include digital twins, big data-driven intelligent manufacturing, deep learning, and decision-making support systems.

GUANGHUI ZHOU (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in mechanical engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, in 1996, 1999, and 2003, respectively.

From 2010 to 2011, he was a Visiting Scholar with the Department of Industrial and Manufacturing Systems Engineering, Florida State University. He is currently a Professor with the School of Mechanical Engineering, Xi'an Jiaotong University. He is the author of three monographs, more than 100 journals and conference papers, and 20 inventions. His research results won the First Prize of the Natural Science of the Ministry of Education of China. His research interests include service-oriented networked digital manufacturing, low-carbon manufacturing, intelligent manufacturing, and knowledge-based manufacturing systems.

YANZHEN JING received the B.E. and M.E. degrees in mechanical manufacturing and automation from the Xi'an University of Technology, Xi'an, Shaanxi, China, in 2018. She is currently pursuing the Ph.D. degree in mechanical engineering with Xi'an Jiaotong University, Xi'an. Her research interests include design and manufacturing collaboration, knowledge-driven decision making, and deep learning.

RUI WANG received the B.Eng. degree in mechanical engineering from the Xi'an University of Technology, in 2006, and the M.E. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2010, where she is currently pursuing the Ph.D. degree in mechanical engineering. Her research interests include digital twin, intelligent manufacturing systems, and workshop scheduling.

FENGTIAN CHANG received the B.E. degree in mechanical engineering and automation from Zhengzhou University, Zhengzhou, China, in 2014, and the Ph.D. degree in mechanical engineering from Xi'an Jiaotong University, Xi'an, China, in 2020.

From 2021 to 2022, he was a Lecturer with Chang'an University, Xi'an. His research interests include advanced manufacturing paradigms, smart product service systems, predictive maintenance service, and collaborative maintenance decision-making.