IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received January 11, 2022, accepted February 6, 2022, date of publication February 11, 2022, date of current version February 22, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3151170

Predictive Maintenance Decision Making Based on Reinforcement Learning in Multistage Production Systems

MAOMAO FENG[1] AND YANG LI[2]
[1]School of Foreign Language Studies, Chang'an University, Xi'an, Shaanxi 710054, China
[2]Department of Industrial Engineering, School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an, Shaanxi 710072, China

Corresponding author: Maomao Feng (mfeng7@chd.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 52175485.

ABSTRACT Predictive maintenance has become increasingly prevalent in modern production systems that are challenged by high-mix low-volume production and short production life cycle. It is very helpful to prevent costly equipment failures, and reduce significant production loss caused by unscheduled machine breakdown. Although important, decision models for joint predictive maintenance and production in manufacturing systems have not been fully explored. Therefore, we propose a reinforcement learning based decision model, that brings together production system modeling and approximate dynamic programming. We start from the development of a state-based model by analyzing the dynamics of a multistage production system with predictive maintenance. It provides an approach to quantitatively evaluate the various disruptions as well as the maintenance decision's impact on production. Then a reinforcement learning method is proposed to explore optimal maintenance policies, that optimize the production and maintenance cost. To further improve the performance of the production system, machine stoppage bottlenecks are defined. An event-based indicator is proved to identify bottlenecks with production data. We test the proposed models in simulation case studies. The proposed predictive maintenance decision model is compared with three policies, which are state-based policy (SBP), time-based policy (TBP) and greedy policy (GP). The numerical studies show that the proposed decision model outperforms the policies, and it has the lowest system cost that is 9.68%, 39.07%, and 39.56% lower than SBP, TBP, and GP, respectively. In addition, the research shows that bottleneck identification and mitigation could help manufacturing systems to achieve more than 9.00% throughput improvement.

INDEX TERMS Production system analysis, Markov chain model, predictive maintenance, decision making, approximate dynamic programming, bottleneck.

I. INTRODUCTION

Production systems must maintain high productivity and low production cost to succeed in the highly competitive business environment. However, unexpected machine random failures can significantly impact the operation of the production systems. The systems are forced to stay in transient states, which causes the systems suffering from more unscheduled machine breakdown with extended durations, and hence leads to low productivity and quality rate [1].

Maintenance plays a central role to reduce machine failures, improve productivity and keep the functional level of products [2]. Predictive maintenance has been recognized as one of the most promising maintenance strategies for production systems because of high efficiency and low cost [3]. Predictive maintenance is an approach that makes maintenance decisions based on the real machine health conditions [4]. It can reduce unscheduled equipment breakdown and prevent maintenance events that are not necessary [5]. Predictive maintenance is recognized as a promising technique that revolutionizes production industry [6].

An optimal decision model is essential to successfully deploy predictive maintenance in production systems. In single stage production systems, all machines are independent of one another. Conducting predictive maintenance based on each machine's health degradation can optimize system performance. However, it is not trivial to make decisions in multistage production systems considering the

The associate editor coordinating the review of this manuscript and approving it for publication was Agustin Leobardo Herrera-May.

complex interdependencies among machines. In the systems, a machine's health degradation cannot only result in the breakdown of the machine, but also starve or block the adjacent machines. It may cause production loss. Predictive maintenance decision models should be adapted to take into consideration of the interdependencies in order to reach a system-wide optimal policy [7].

This research proposes a decision model for integrated production and maintenance decision making in multistage production systems. First, we propose a Markov chain model through analyzing the transient dynamics of multistage production systems with predictive maintenance control. In the existing analytical models, systems mainly use corrective maintenance to bring downtime machines back to operation while ignoring predictive maintenance. We extend the models to production systems where machines have multiple deterioration states, and predictive maintenance is employed to restore machines to better health conditions. Second, we integrate approximate dynamic programming and the Markov chain model. The existing research usually considers the modeling and control of multistage production systems separately. We bring together the research efforts, and investigate their applications in predictive maintenance decision making.

Machine stoppage bottlenecks refer to the machines whose random failure most strongly impedes throughput. The performance of the production system can be effectively improved if we give higher maintenance priorities to bottleneck machines. The challenge is to define and identify bottleneck machines. Therefore, we propose in this manuscript the definition and identification method for machine stoppage bottlenecks.

The remaining of the paper is presented as follows: literature review is discussed in Section II. Section III introduces system's descriptions and assumptions. Section IV proposes the Markov chain model. Sections V and VI describe the dynamic maintenance decision model. Section VII defines machine stoppage bottlenecks and establishes an event-based identification method. Section VIII carries out the numerical case study to validate the proposed models. Conclusions are summarized in Section IX.

II. LITERATURE REVIEW

In an effort to improve production and maintenance efficiency, extensive studies have been carried out in the past decades on joint maintenance and production control [8], [9]. Zhang *et al.* [10] established a reinforcement learning based algorithm to exploit optimal maintenance policies. The model can be applied to systems whose degradation modes can be either known or not. Yang *et al.* [11] investigated the integrated optimization of preventive maintenance and production scheduling in a multi-stage sing-machine production system. A Markov decision process framework is formulated and R-learning algorithm is applied to optimize the system rewards. In [12], a repairable multistate system is considered to optimize the achievement over a finite time period with maintenance resources constrain. A Markov decision process is established, and it is solved with a reinforcement learning based algorithm. Peng [13] proposed a Markov decision process model for systems with continuously degradation states. The transition and value function of system states are approximated with the Gaussian process regression. Although the aforementioned models are very useful to improve system performance in their applications, they are mostly developed for systems with one or two stages. The methods are difficult to be extended to multistage manufacturing systems.

Although some research has been carried out in multistage manufacturing systems to establish predictive maintenance decision models, most of them relies on heuristic rules without a systematic understanding of system dynamics [14]. As a result, the models may not adequately exploit optimal maintenance decisions. It is desired to integrate production system models into maintenance decision making algorithms to make comprehensive decisions in multistage production systems. For example, Iravani and Duenyas [15] discussed the application of reinforcement learning in production maintenance. It has the potential to improve $5-20\%$ maintenance efficiency comparing with conventional maintenance policies Xia *et al.* [16] proposed a predictive maintenance control policy for serial production systems, which utilizes a global-objective model (GOM) to make machine-level decisions and a maintenance time window method (MTW) to make system-level decision. The multi-unit production system is simplified by assuming that the entire production system needs to be stopped to perform predictive maintenance. Chang *et al.* [17] introduced a supervisory control algorithm to schedule predictive maintenance in maintenance opportunity window (MOW). The algorithm assumes that all the machine failures can be predicted. It strives to satisfy the maintenance needs while only slightly impede production efficiency.

Bottlenecks in production systems have been analyzed in numerous references. Cui *et al.* [18] studied bottlenecks in a serial production system, and established an identification method with accumulated starvation and blockage time. However, the identification method can only be analytically proved in production systems without random machine failure. Zhang *et al.* [19] analyzed three types of bottlenecks in Markovian production systems where machines can have multiple states. An approximation method is proposed to find bottlenecks. Li *et al.* [20] utilized a Markov chain model to identify market demand bottlenecks. A continuous improvement algorithm is established based on the bottleneck identification method. This manuscript will extend the bottleneck identification methods to production systems with maintenance.

In summary, the current literature fails to provide a systematic approach to the joint production and maintenance decision making in multistage production systems. Although extensive research efforts have been reported to optimize maintenance policies, they are mainly for single stage production systems. Many current decision models for multistage production systems rely on tribal knowledge or ad-hoc rules. The complex nature of a production system makes it difficult

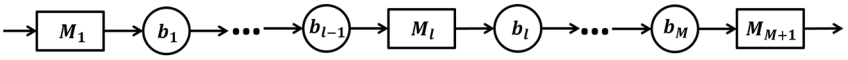


FIGURE 1. A multistage serial production line. It consists of $M+1$ machines separated by M buffers, where $M > 0$.

to utilize these methods to optimize system performance. The ever-growing complex business environment puts huge pressure on production systems. Dynamic decision models that integrate production system modeling and optimal control method is imperative to reduce maintenance cost and improve productivity. And this is the main concern of the paper.

III. SYSTEM ASSUMPTIONS AND NOMENCLATURE

A. SYSTEM ASSUMPTIONS AND BACKGROUND

In this paper, we consider a serial production system as shown in Figure 1. We use $M_l, 1 \le l \le M$, to represent the lth production machine and b_l to represent the lth buffer. The end-of-line machine M_{M+1} is the market demand virtual machine. The last buffer b_M is the finished-goods buffer. The system makes the definitions and assumptions.

1) The capacity of each buffer is finite.
2) The cycle time of each machine is identical and equals to a time unit [21], [22].
3) Virtual machine M_{M+1} has $N_{M+1}+1$ states, i.e. $\alpha_{M+1} \in \{0, 1, \ldots, N_{M+1}\}$. When virtual machine M_{M+1} is in state α_{M+1}, it has a rated speed of $v_{M+1}(\alpha_{M+1})$, which is determined by the actual market demand. Machine $M_{M+1}, 1 \le l \le M$, can transfer from state $\alpha_{M+1} = i$ to state $\alpha_{M+1} = j$ with a probability of $p_{M+1}(i, j)$ at each time unit.
4) Machine $M_l, 1 \le l \le M$, has $N_l + 1$ deterioration states, which are denoted as $\alpha_l \in \{0, 1, \ldots, N_l\}$. State $\alpha_l = 0$ represents as good as new state, while state $\alpha_l = N_l$ represents failure state. Each machine has one failure state.
5) Without maintenance, machine $M_l, 1 \le l \le M$, will eventually fail and almost never revives on its own. We assume that machine $M_l, 1 \le l \le M$, can transfer from state $\alpha_l = i$ to state $\alpha_l = j$ with a probability of $\theta_{ij}, i \ne N_l$.
6) When machine $M_l, 1 \le l \le M$, fails, corrective maintenance (CM) shall be performed to bring the machine to one of the non-failure states. We use $\beta_{N_l j}, j \ne N_l$, to denote the transition probability of machine M_l from state N_l to state j with CM.
7) Before machine $M_l, 1 \le l \le M$, fails, predictive maintenance (PM) can be performed to restore the machine to a healthier state. We assume that the duration of PM is shorter than that of CM on average. PM usually has lower cost rate than CM. With PM, the transition probability of machine M_l from state i to state j is denoted as $\gamma_{ij}, i \ne j \ne N_l$.
8) $\vec{c}(t) = [c_1(t), \ldots, c_M(t)]$ denotes the maintenance decision of each machine at time t, where

$$c_l(t) = \begin{cases} 1, & \text{performing PM on machine } M_l \\ -1, & \text{performing CM on machine } M_l \\ & l = 1, \ldots, M. \\ 0, & \text{Otherwise} \end{cases}$$

9) $p_l(\alpha_l(t-1), \alpha_l(t), c_l(t-1))$ denotes the probability that machine M_l transfers from state $\alpha_l(t-1)$ to state $\alpha_l(t)$ given maintenance decision $c_l(t-1)$, where

$$p_l(\alpha_l(t-1), \alpha_l(t), c_l(t-1)) = \begin{cases} \gamma_{\alpha_l(t-1)\alpha_l(t)}, & c_l(t-1) = 1 \\ \beta_{\alpha_l(t-1)\alpha_l(t)}, & c_l(t-1) = -1 \quad l = 1, \ldots, M. \\ \theta_{\alpha_l(t-1)\alpha_l(t)}, & c_l(t-1) = 0 \end{cases}$$

10) $TH_l(t), 1 \le l \le M+1$, denotes the throughput of machine M_l at time t.
11) $g_l(c_l(t))$ denotes the maintenance cost of machine $M_l, l = 1, \ldots, M$, at time t. It is assumed that $g_l(c_l(t)) = 0$ if $c_l(t) = 0$ and $g_l(c_l(t)) > 0$ if $c_l(t) \ne 0$.

Remark 1: The expected length of the corrective maintenance performed on machine $M_l, l = 1, \ldots, M$, is $\frac{1}{\sum_{0 \le j < N_l} \beta_{N_l j}}$.

Remark 2: If predictive maintenance is performed on machine $M_l, l = 1, \ldots, M$, when it is in state $\alpha_l = i$, the maintenance continues until machine M_l successfully transfers to another state. The expected length of the predictive maintenance is estimated as $\frac{1}{\sum_{j \ne i} \gamma_{ij}}$.

B. NOMENCLATURE

$b_l, l = 1, \ldots, M$	the lth buffer
$B_l, l = 1, \ldots, M$	capacity of buffer b_l
$b_l(t), l = 1, \ldots, M$	buffer level of buffer b_l at time t
$M_l, l = 1, \ldots, M+1$	the lth machine
$\alpha_l(t), l = 1, \ldots, M+1$	state of machine M_l at time t
$v_{M+1}(\alpha_{M+1}(t))$	rated speed of machine M_{M+1} in state $\alpha_{M+1}(t)$
s	production system state
$s(t)$	system's state at time t
$\vec{c}$	maintenance decision of the system
$\vec{c}(t)$	maintenance decision of the system at time t
$c_l(t), 1 \le l \le M$	maintenance decision of machine M_l at time t

$r(s(t), \vec{c}(t))$	cost of the system when it is in state $s(t)$ with control decision $\vec{c}(t)$
$\vec{\theta}$	real valued weights of the neural network function approximator
π	maintenance policy
$V^{\pi}(s)$	value function in state s with maintenance policy π
$V\left(s, \vec{\theta}\right)$	optimal value function in state s
$L\left(\vec{\theta}\right)$	mean-squared error in the Bellman equation

IV. ANALYTICAL ANALYSIS OF MULTISTAGE MANUFACTURING SYSTEMS

At each time t, the production system is modelled by a Markov chain with state defined as $s(t) = [\alpha_1(t), \ldots, \alpha_{M+1}(t), b_1(t), \ldots, b_M(t)]$. The production system dynamics are described with:

$$b_l(t) = b_l(t-1) + TH_l(t) - TH_{l+1}(t), \quad l = 1, \ldots, M+1 \tag{1}$$

where $TH_l(t)$ and $TH_{l+1}(t)$ are the throughputs of machines M_l and M_{l+1} at time t.

When M_l, $1 < l \leq M+1$, is not the first machine in the system, its throughput $TH_l(t)$ at time t can be determined by comparing its rated speed $v_l(\alpha_l(t))$ in state $\alpha_l(t)$, the buffer level $b_{l-1}(t-1)$ of the immediate upstream buffer b_{l-1} at time $t-1$, and the available buffer space $B_l - b_l(t-1) + TH_{l+1}(t)$ of the immediate downstream buffer b_l at time t, i.e.

$$TH_l(t) = \min\{v_l(\alpha_l(t)), b_{l-1}(t-1), B_l - b_l(t-1) + TH_{l+1}(t)\}, \quad 1 < l \leq M. \tag{2}$$

For the first machine M_1, the throughput $TH_1(t)$ is determined by comparing the rated speed $v_1(\alpha_1(t))$ of the machine in state $\alpha_1(t)$, and the available buffer space $B_1 - b_1(t-1) + TH_2(t)$ of the immediate downstream buffer b_1 at time t, i.e.

$$TH_1(t) = \min\{v_1(\alpha_1(t)), B_1 - b_1(t-1) + TH_2(t)\}. \tag{3}$$

For the last virtual machine M_{M+1}, the throughput $TH_{M+1}(t)$ is determined by comparing its rated speed $v_{M+1}(\alpha_{M+1}(t))$ in state $\alpha_{M+1}(t)$, and the buffer level $b_M(t-1)$ of the immediate upstream buffer b_{M-1} at time $t-1$, i.e.

$$TH_{M+1}(t) = \min\{v_{M+1}(t), b_M(t-1)\}. \tag{4}$$

Therefore, the throughput of each machine is summarized in the following Lemma 1.

Lemma 1: In the production system defined in Section III.A, the throughput of each machine is expressed as:

$$TH_l(t) = \begin{cases} \min\{v_l(\alpha_l(t)), B_l - b_l(t-1) + TH_{l+1}(t)\}, \\ \qquad l = 1 \\ \min\{v_l(\alpha_l(t)), b_{l-1}(t-1), B_l - b_l(t-1) + TH_{l+1}(t)\}, \\ \qquad 2 \leq l \leq M \\ \min\{v_l(\alpha_l(t)), b_{l-1}(t-1)\} \\ \qquad l = M+1 \end{cases} \tag{5}$$

We suppose that at time t, the system is in state $s(t) = [\alpha_1(t), \ldots, \alpha_{M+1}(t), b_1(t), \ldots, b_M(t)]$, and the maintenance decision is $\vec{c}(t) = [c_1(t), \ldots, c_M(t)]$. The probability that machines are in states $\vec{M}(t+1) = [\alpha_1(t+1), \ldots, \alpha_{M+1}(t+1)]$ at the following time step $t+1$ is estimated as:

1)] at the following time step $t+1$ is estimated as:

$$P(\vec{M}(t), \vec{M}(t+1), \vec{c}(t)) = p_{M+1}(\alpha_{M+1}(t), \alpha_{M+1}(t+1)) \prod_{l=1}^{M} p_l(\alpha_l(t), \alpha_l(t+1), c_l(t)). \tag{6}$$

At time $t+1$, give n the machines' states $\vec{M}(t+1)$, the buffer level $\vec{B}(t+1) = [b_1(t+1), \ldots, b_M(t+1)]$ can be calculated with Equations 1 and 2. Therefore, the transition probability from states $s(t)$ to $s(t+1) = [\vec{M}(t+1), \vec{B}(t+1)]$ is determined as:

$$P(s(t), s(t+1), \vec{c}(t)) = P(\vec{M}(t), \vec{M}(t+1), \vec{c}(t)). \tag{7}$$

The process is repeated until all the transition probabilities are obtained. The process is summarized in Algorithm 1.

Algorithm 1

Step 0. Input system state $s(t)$ and maintenance decision $\vec{c}(t)$

Step 1. For all machines' states $\vec{M}(t+1)$

Compute the buffer levels for the following time step $t+1$ with Equations 1 and 5.

Compute the transition probability from $s(t)$ to $s(t+1)$ with Equations 6 and 7.

End For

Remark 3: According to Equation 7, it is noted that the transition of system state is determined based on the transition of machines' states. When system is in state $s(t)$, it can transfer to at most $\prod_{l=1}^{M+1} N_l$ other states, where $\prod_{l=1}^{M+1} N_l$ is the total number of possible machines' states combinations. Therefore, each time when we compute the transition probability, we only need to take into consideration of the $\prod_{l=1}^{M+1} N_l$ transitions. The probabilities for the system to transfer to all other states equal to zero.

V. PREDICTIVE MAINTENANCE DECISION MODEL FORMULATION

The maintenance decision making problem is a typical Markov Decision Process (MDP), which can be described with a 4-tuple [S, C, P, R]:

- S denotes the state space,
- C denotes the action space,
- P: $S \times S \times C \rightarrow [0, 1]$ denotes the transition probability matrix with element $P(s, s', \vec{c})$, $\forall s, s' \in S$ and $\forall \vec{c} \in C$,
- R: $S \times C$ denotes the cost function, where $r(s, \vec{c}) \in R$, $\forall s \in S$ and $\forall \vec{c} \in C$, represents the immediate cost of the system in state s and having maintenance decision $\vec{c}$.

System cost $r(s(t), \vec{c}(t))$ at each time t constitutes inventory cost, backlog cost, and maintenance cost, which is calculated as follows:

$$r(s(t), \vec{c}(t)) = g^b \sum_{l=1}^{M} b_l(t) + g^- x^-(t) + g^m(t), \quad (8)$$

where g^b is the inventory cost per part, and g^- is the backlog cost per part. The backlog incurred in the system is measured as the production loss in the last virtual machine M_{M+1}, i.e. $x^-(t) = v_{M+1}(\alpha_{M+1}(t)) - TH_{M+1}(t)$. $g^m(t)$ is the maintenance cost, i.e. $g^m(t) = \sum_{l=1}^{M} g_l(c_l(t))$.

A maintenance policy is a function from system states to maintenance decisions. Policy $\pi(s, \vec{c})$ denotes executing maintenance decision $\vec{c} \in C$ in state $s \in S$. The value function of the policy is $V^\pi(s)$, which estimates the discounted cost when the system is in state s and it follows policy π thereafter:

$$V^\pi(s) = E_\pi[\sum_{t=0}^{\infty} \gamma^t r(s(t), \vec{c}(t)) | s(0) = s], \quad (9)$$

where $s(0)$ refers to the initial system state and γ, $0 < \gamma < 1$, is the discount factor.

The objective of the decision model is to find an optimal maintenance policy π^* that minimizes the expected discounted cost over an infinite horizon, i.e. $\forall \pi \neq \pi^*$, $V^{\pi^*}(s) \leq V^\pi(s)$. The optimal policy satisfies Bellman's optimality equation, which can be expressed as:

$$V(s) = \min_{\vec{c}} \left[r(s, \vec{c}) + \gamma \sum_{s}^{'} P(s, s', \vec{c}) V(s') \right], \forall s, \quad s' \in S. \quad (10)$$

Solving Equation 10 encounters the curses of dimensionality especially when system state space S and the action space C become large. It is difficult, if not impossible, to apply traditional dynamic programming methods to solve the problem. In the following section, we will introduce an approximate dynamic programming algorithm for exploring the optimal maintenance policy.

VI. PREDICTIVE MAINTENANCE DECISION MODEL FORMULATION

The recent breakthrough of reinforcement learning (RL) in AlphaGo proves the method as a good approach to handle dynamic programming with large state and action spaces [23], [24]. Actually, some pioneer works have been presented that apply RL to solve optimal control problems in production systems. McDonnell *et al.* [25] employed a reinforcement learning approach for specifying the payoffs in reconfiguration games in a heterarchical production system Csáji *et al.* [26] presented an adaptive iterative distributed scheduling algorithm in a market-based production control system. The algorithm uses a triple-level learning mechanism, and is based on deep reinforcement learning.

Therefore, we propose to use reinforcement learning (RL) to perform maintenance decision making. Instead of directly exploring maintenance policies, the aim of the method is to learn the optimal value function $V^*(s)$, $s \in S$. The optimal policies are obtained based on the value function estimations. A neural network function approximator is adopted to represent the value function. The neural network function approximator is organized in a standard multilayer perception architecture, which is parameterized by real valued weights $\vec{\theta} = [\theta_1, \ldots, \theta_n]$. In the network, the value function $V\left(s, \vec{\theta}\right)$ is computed with a feed-forward flow of activation from the input neurons to the output neurons, passing through one or more layers of hidden neurons [27].

The neural network function approximator is trained with a reinforcement learning algorithm. However, conventional reinforcement learning algorithms with neural network function approximator can oscillate or diverge when they learn directly from consecutive samples [28]. To overcome the limitations, we adopt a mechanism named experience replay, which can randomize the training samples and break the strong correlations among them. The idea behind experience replay is to randomly sample a mini-batch Ψ of previous experiences of the system, and smooth out learning over many historical experiences. The training process is summarized in Algorithm 2.

Algorithm 2

Step 1. Randomly determine replay memory D_1, and the network parameters $\vec{\theta}$. Set $\vec{\theta}^- = \vec{\theta}$.

Step 2. For iteration $\varsigma = 1 : \Omega$

Randomly determine initial state s_0.
For $\tau = 1 : \Theta$
Select $\vec{c}(\tau)$ based on ε-greedy, and execute the maintenance decision.
Record $\{s(\tau), \vec{c}(\tau), r(\tau), s(\tau + 1)\}$ into D_ς.
Sample random mini-batch from D_ς.
For each $\{s, \vec{c}, r, s'\} \in$, update
$V(s)$ with Equation 10 and Algorithm 1.
Update $\vec{\theta}$ with gradient descent, and update $\vec{\theta}^- = \vec{\theta}$ every certain steps.

End For

End For

The main steps of the algorithm are introduced as follows.

1. *Replay Memory Generation and Update.* In each time step t, the production system selects and executes maintenance actions according to ε-greedy policy. It means that with probability ε, the system selects and executes a random action $\vec{c}(t)$. And with probability $1 - \varepsilon$, it selects and executes action $\vec{c}(t)$ by solving

$$\vec{c}(t) = arg \min_{\vec{c}} (r(s(t), \vec{c}(t)) + \gamma \sum_{s(t+1)} P(s(t), \vec{c}(t), s(t+1)) V\left(s(t+1), \vec{\theta}\right) \quad (11)$$

Then the experience $e_t = (s(t), \vec{c}(t), r(t), s(t+1))$ is stored in a data set $D_t = (e_1, \ldots, e_t)$, which is named as replay memory data set. The size of the replay memory data set is fixed. It stores the latest N experiences.

2. *Approximation Update.* The network is trained through iteratively adjusting the parameters $\vec{\theta}$ to reduce the mean-squared error in the Bellman equation, which is defined as

$$L\left(\vec{\theta}\right) = E[\min_{\vec{c}}(r(s,\vec{c}) + \gamma V\left(s', \vec{\theta}^{-}\right)) - V\left(s, \vec{\theta}\right)]^2. \tag{12}$$

where $\vec{\theta}^{-}$ are the parameters $\vec{\theta}$ from the previous iteration.

Stochastic gradient descent is applied to reduce the value of Error Equation 12. During each iteration, a set of experiences are randomly drawn from the replay memory pool, and utilized to compute the target

$$y_s = \min_{\vec{c}}(r(s,\vec{c}) + \gamma V\left(s', \vec{\theta}^{-}\right)).$$

Then $\vec{\theta}$ is updated with a gradient descent step on $\left(y_s - V\left(s, \vec{\theta}\right)\right)^2$ as

$$\vec{\theta} = \vec{\theta} - \alpha(y_s - V\left(s, \vec{\theta}\right))\nabla_{\vec{\theta}} V\left(s, \vec{\theta}\right), \tag{13}$$

where $\alpha = 1/n$ in the nth training iteration.

3. *Stopping Criteria.* The algorithm determines the final neural network function approximator and the corresponding maintenance policy by repeating the aforementioned steps for certain iterations.

Note to Practitioner. In order to apply the proposed maintenance decision model, the production systems should be able to continuously track the health status of machines. In addition, the production systems should have access to production data including the states of machines, buffer levels, maintenance decisions, etc. Then, the current system state $s(t)$ is adopted as the input to the trained neural network function approximator. Equation 11 is adopted to determine the optimal maintenance control. The decision-making process is shown in the following Procedure 1.

Procedure 1

Step 1. Collect the current system state $s(t)$.

Step 2. Compute $V\left(s, \vec{\theta}\right)$ by running the forward propagation of the trained network

Step 3. Determine the optimal maintenance decision according to Equation 11.

VII. MACHINE STOPPAGE BOTTLENECKS

If the end-of-line machine M_{M+1} is starved, the production system fails to satisfy the market demand. The products that are not satisfied are defined as unsatisfied market demand (USMD). It can be calculated as the production loss of M_{M+1} caused by starvation.

Machine stoppage is the most direct cause of USMD. Identifying and mitigating machine stoppage bottlenecks have been considered as one of the most cost-effective approaches to reduce USMD. Machine stoppage bottlenecks refer to the machines that their stoppage most significantly impacts USMD. It is defined in the following Definition 1.

Definition 1: Machine stoppage bottleneck (MSB) is machine M_m, $1 \le m \le M$, if

$$\left|\frac{\partial USMD}{\partial DTD_m}\right| \ge \left|\frac{\partial USMD}{\partial DTD_l}\right|, \tag{14}$$

where $1 \le l \le M$ and $l \ne m$. In the definition, DTD_m refers to the average downtime duration (DTD) of machine M_m, $1 \le m \le M$, because of random machine failures or predictive maintenance. It measures the average time that the system takes to recover M_m from down states to working states.

According to the definition, machine M_m is MSB if a small amount decrease of its DTD could lead to the greatest decrease of USMD. However, identifying MSB with its definition is challenge because there is no close-form expression of USMD in multistage production systems, let alone to estimate its derivatives. It is necessary to develop a method that can identify MSB with production or simulation data.

We adopt $\vec{\omega}_m^i = (t_i, d_i)$ to represent the i th starvation event that the end-of-line machine M_{M+1} is starved by machine M_m, $1 \le m \le M$, from time t_i to $t_i + d_i$. $W_m(T) = \{\vec{\omega}_m^1, \ldots, \vec{\omega}_m^{n_m}\}$ denotes all the starvation events resulted from machine M_m during time $(0, T]$. USDM can be calculated with the starvation event information as

$$USDM = \sum_{m=1}^{M} \sum_{k=1}^{n_m} \sum_{\tau=t_k}^{t_k+d_k} v_{M+1}(\alpha_{M+1}(\tau)). \tag{15}$$

The identification method for MSB can be established based on Equation 15.

Proposition 1: For machines M_i and M_j, $i \ne j$, if $n_i \ge n_j$, then they have $\left|\frac{\partial USMD}{\partial DTD_i}\right| \ge \left|\frac{\partial USMD}{\partial DTD_j}\right|$.

Proof: USDM can be calculated as

$$USDM = \sum_{m=1}^{M} \sum_{k=1}^{n_m} \sum_{\tau=t_k}^{t_k+d_k} v_{M+1}(\alpha_{M+1}). \tag{16}$$

If the repair time of machine M_i is reduced by a small amount of δDTD_i, then average USDM becomes

$$\begin{aligned} USDM' = &\sum_{k=1}^{n_i} \sum_{\tau=t_k}^{t_k+d_k-\delta DTD_i} v_{M+1}(\alpha_{M+1}) \\ &+ \sum_{m\ne i} \sum_{k=1}^{n_m} \sum_{\tau=t_k}^{t_k+d_k} v_{M+1}(\alpha_{M+1}). \end{aligned} \tag{17}$$

When time T is long enough, equations 16 and 17 can be approximated with the average speed of the end-of-line machine M_{M+1}, i.e. $\bar{v}_{M+1}$, as

$$USDM = \bar{v}_{M+1} \sum_{m=1}^{M} \sum_{k=1}^{n_m} d_k, \tag{18}$$

$$\begin{aligned} USDM' = &\bar{v}_{M+1} \sum_{k=1}^{n_i} (d_k - \delta DTD_i) \\ &+ \bar{v}_{M+1} \sum_{m\ne i} \sum_{k=1}^{n_m} d_k. \end{aligned} \tag{19}$$

TABLE 1. Parameters of the production system.

Machines	M_1	M_2	M_3	M_4	M_5	M_6
s(parts/min)	0.2	0.2	0.2	0.2	0.2	0.2
Buffers	b_1	b_2	b_3	b_4	b_5	
Capacity	5	3	4	3	4	

TABLE 2. Transition probabilities of machines M_1 to M_5.

		Without Maintenance					With PM					With CM			
M_1	$p_{i,j}$	0	1	2	3	$\beta_{i,j}$	0	1	2	3	$\gamma_{i,j}$	0	1	2	3
	0	0.55	0.30	0.10	0.05	0	1.00	-	-	-	0	1.00	-	-	-
	1	-	0.65	0.20	0.15	1	0.25	0.75	-	-	1	-	1.00	-	-
	2	-	-	0.60	0.40	2	0.20	0.10	0.70	-	2	-	-	1.00	
	3	-	-	-	1.00	3	-	-	-	1.00	3	0.10	0.10	0.15	0.65
M_2	$p_{i,j}$	0	1	2	3	$\beta_{i,j}$	0	1	2	3	$\gamma_{i,j}$	0	1	2	3
	0	0.60	0.30	0.09	0.01	0	1.00	-	-	-	0	1.00	-	-	-
	1	-	0.70	0.20	0.10	1	0.20	0.80	-	-	1	-	1.00	-	-
	2	-	-	0.60	0.40	2	0.20	0.10	0.70	-	2	-	-	1.00	-
	3	-	-	-	1.00	3	-	-	-	1.00	3	0.10	0.10	0.10	0.70
M_3	$p_{i,j}$	0	1	2	3	$\beta_{i,j}$	0	1	2	3	$\gamma_{i,j}$	0	1	2	3
	0	0.65	0.25	0.09	0.01	0	1.00	-	-	-	0	1.00	-	-	-
	1	-	0.70	0.20	0.10	1	0.20	0.80	-	-	1	-	1.00	-	-
	2	-	-	0.55	0.45	2	0.15	0.10	0.75	-	2	-	-	1.00	-
	3	-	-	-	1.00	3	-	-	-	1.00	3	0.05	0.10	0.10	0.75
M_4	$p_{i,j}$	0	1	2	3	$\beta_{i,j}$	0	1	2	3	$\gamma_{i,j}$	0	1	2	3
	0	0.60	0.30	0.09	0.01	0	1.00	-	-	-	0	1.00	-	-	-
	1	-	0.70	0.20	0.10	1	0.20	0.80	-	-	1	-	1.00	-	-
	2	-	-	0.60	0.40	2	0.20	0.10	0.70	-	2	-	-	1.00	-
	3	-	-	-	1.00	3	-	-	-	1.00	3	0.10	0.10	0.10	0.70
M_5	$p_{i,j}$	0	1	2	3	$\beta_{i,j}$	0	1	2	3	$\gamma_{i,j}$	0	1	2	3
	0	0.60	0.30	0.09	0.01	0	1.00	-	-	-	0	1.00	-	-	-
	1	-	0.75	0.15	0.10	1	0.25	0.75	-	-	1	-	1.00	-	-
	2	-	-	0.60	0.40	2	0.15	0.10	0.75	-	2	-	-	1.00	-
	3	-	-	-	1.00	3	-	-	-	1.00	3	0.05	0.10	0.15	0.70

The partial differential equation as demonstrated in Definition 1 can be estimated as

$$\left|\frac{\partial USMD}{\partial DTD_i}\right| = \frac{USMD - USMD'}{\delta DTD_i} = \frac{\bar{v}_{M+1} n_i \delta DTD_i}{\delta DTD_i} = \bar{v}_{M+1} n_i. \quad (20)$$

It is noted that $\bar{v}_{M+1}$ is constant. Therefore, n_i can be utilized as the indicator to determine the value of $\left|\frac{\partial USMD}{\partial DTD_i}\right|$. ■

n_i measures the number of times that machine M_i causes the starvation of the end-of-line machine M_{M+1}. It can be determined directly from the collected production information or simulation results. Therefore, Proposition 1 presents the indicator for MSB identification, that the machine with the highest n_i is MSB.

Remark 4: Since the transition of end-of-line machine M_{M+1} is assumed to be time dependent, the probability distribution of the machine can be obtained by solving a sequence of balance equations:

$$P(\alpha_{M+1} = j) = \sum_{i=0}^{N_{M+1}} p_{M+1}(i,j) P(\alpha_{M+1} = i), \quad (21)$$

and $\sum_{i=0}^{N_{M+1}} P(\alpha_{M+1} = i) = 1$.

TABLE 3. Transition probability of machine M_6 without maintenance.

$p_{i,j}$	0	1
0	0.30	0.70
1	0.30	0.70

The average speed $\bar{v}_{M+1}$ is determined as $\bar{v}_{M+1} = \sum_{\alpha_{M+1}} P(\alpha_{M+1}) v_{M+1}(\alpha_{M+1})$, and it is constant.

Note to Practitioner. The application of the bottleneck identification method is shown in Procedure 2.

Procedure 2

Step 1. Collect production information.

Step 2. Compute n_i of each machine.

Step 3. The machine with the highest n_i is identified as MSB.

VIII. NUMERICAL STUDIES

This section performs numerical studies to analyze the proposed maintenance decision model and the bottleneck identification method. The simulation environment is Tensorflow

TABLE 4. Parameters of costs.

	Inventory cost ($/part/cycle)	Backlog cost ($/part)	Predictive maintenance cost ($/cycle)	Reactive maintenance cost ($/cycle)
Cost	10	100	100	300

TABLE 5. Sizes of replay memory and mini-batch.

Case number	1	2	3	4	5	6	7	8
Replay memory	5,000	10,000	15,000	20,000	25,000	30,000	35,000	40,000
Mini-batch	10	15	20	25	30	35	40	45

TABLE 6. CPU time and system cost in all cases.

Case number	1	2	3	4	5	6	7	8
Average CPU time (s)	253	312	374	429	533	583	647	713
Standard deviation	13.92	6.65	4.49	9.87	20.35	52.47	43.49	45.18
Average system cost ($)	83,997,960	80,105,640	81,459,840	81,996,120	83,966,880	83,160,720	83,922,000	84,171,360
Standard deviation	940,800	656,928	937,385	1,084,157	557,293	1,271,728	2,845,003	934,389

TABLE 7. Comparison results of four maintenance policies.

	Maintenance cost ($)	Inventory cost ($)	Backlog cost ($)	Total cost ($)
GP	74,131,200	9,330,480	4,438,800	87,900,480
SBP	71,126,400	9,058,800	4,363,200	84,548,400
TBP	73,575,600	9,090,240	4,364,400	87,030,240
RLP	66,753,600	9,051,240	4,300,800	80,105,640

1.14.0 with Python 3.6, and the simulation is performed on a laptop with Intel i7-5500U CPU, and 16.0 GB RAM.

A. ANALYSIS OF MAINTENANCE DECISION MODEL

We now consider a serial production system that consists of 6 machines and 5 buffers. The system parameters as demonstrated in Tables 1-3. In the system, machine $M_l, 1 \leq l \leq 5$, has 4 states, which are denoted as $\alpha_l \in \{0, 1, 2, 3\}$. For ease of discussion, we assume that the machine's health condition degrades from state $\alpha_l = 0$ to state $\alpha_l = 3$. The speed of machine $M_l, 1 \leq l \leq 5$, is $v_l(\alpha_l) = \begin{cases} 1, & \alpha_l = 0, 1, 2 \\ 0, & \alpha_l = 3 \end{cases}$.

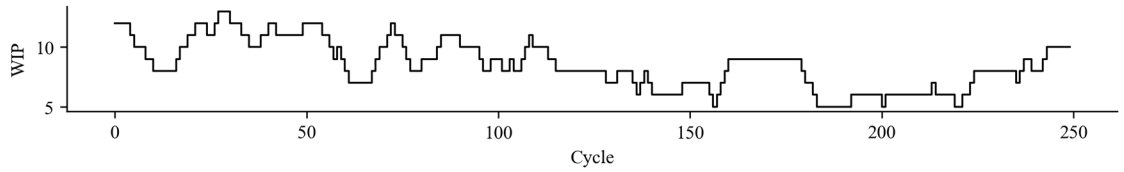
We assume the market demand virtual machine M_6 has 2 states, i.e. $\alpha_6 \in \{0, 1\}$. The speed of the virtual machine is $v_6(\alpha_6) = \begin{cases} 1, & \alpha_6 = 0 \\ 0, & \alpha_6 = 1 \end{cases}$. The maintenance cost for each machine is summarized in Table 4. The inventory cost is \$60 per part and time unit, and backlog cost is \$100 per part. Each time step has 5 minutes and it equals to the cycle time of the machines.

First, the implementation of the proposed maintenance decision making algorithm is demonstrated. The core of the algorithm is the construction and training of the neural network, which uses system state $s(t)$ as input, and computes the corresponding value function $V(s(t))$. The neural network is a feed-forward network, which has an input layer, a hidden layer and an output layer. In this case, the input layer consists of 17 input neurons. The output layer is a fully-connected linear layer with a single output. The hidden layer is fully-connected and the number of its neurons is determined based on [29] as 9.

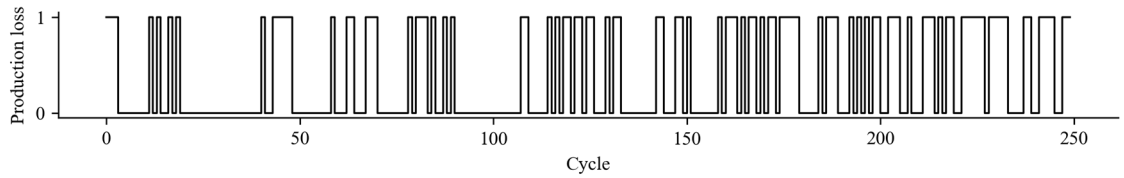
The maximum number of iterations is $J = 1000$. The probability ε in ε-greedy linearly decreases from 1 to 0.1 during the training process. The training process is repeated for 10 times. In each training, the training data is generated with simulation. The transition $\{s(t), \vec{c}(t), r(t), s(t+1)\}$ is recorded in the replay memory. The sizes of replay memory and mini-batch are demonstrated in Table 5. The average CPU time, the average system cost $r(t)$, as well as their standard deviation are demonstrated in Table 6. It is observed that the training time increases from 253s to 713s as the sizes of replay memory and mini-batch increase. The results also indicate that the average system cost decreases first and then increases when the sizes of replay memory and mini-batch increase. The proposed maintenance decision making algorithm achieves the least system cost when the replay memory has a size of 10000 and mini-batch has a size of 15. And the CPU time is 312s, which is the second least CPU time. Therefore, in this case, the sizes of replay memory and mini-batch are 10000 and 15, respectively.

Second, we compare the performance of the maintenance policies suggested by the proposed algorithm, which is denoted as reinforcement learning policy (RLP), to that of the following three widely accepted policies by manufacturers.

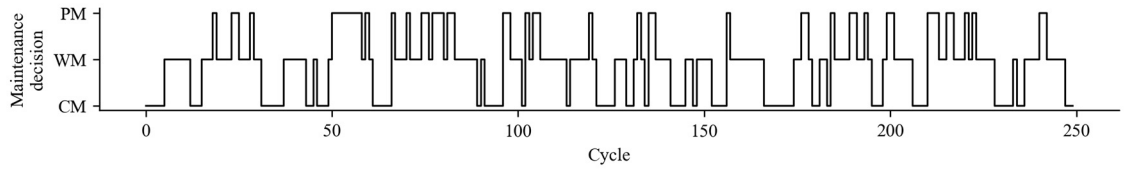
• *State-based policy (SBP):* Predictive maintenance is performed on machine $M_l, 1 \leq l \leq 5$, if it reaches state $\alpha_l = 2$. The policy is based on the observation that reactive maintenance is usually much more costly than predictive



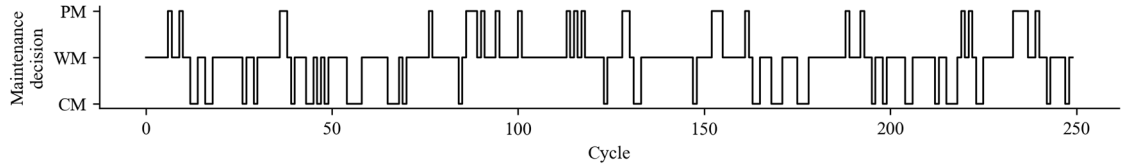
(a) WIP inventory



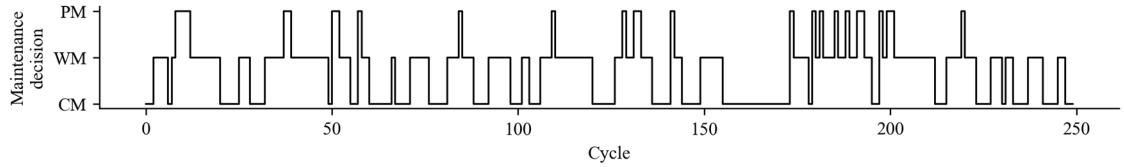
(b) Production loss of the end-of-line virtual machine M_6



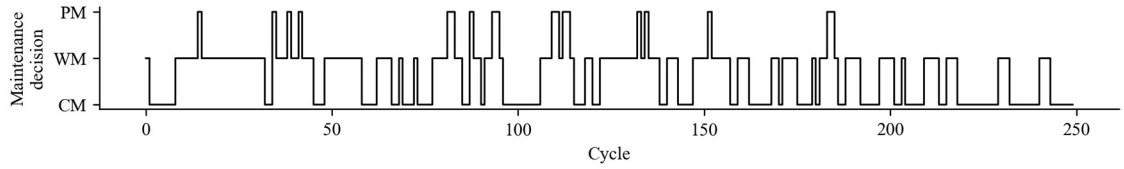
(c) Maintenance decision of machine M_1



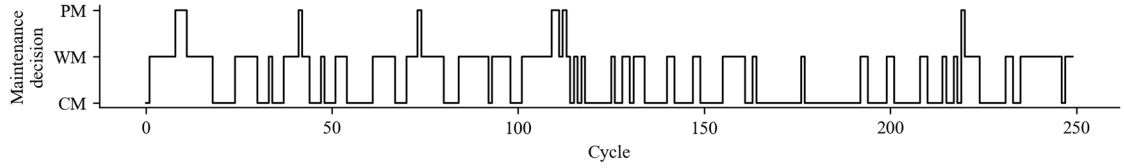
(d) Maintenance decision of machine M_2



(e) Maintenance decision of machine M_3



(f) Maintenance decision of machine M_4



(g) Maintenance decision of machine M_5

FIGURE 2. **WIP inventory, production loss and maintenance decisions with RLP. The figures plot the trajectories of WIP inventory, the production loss of the last virtual machine, and the maintenance decision of each machine with RLP during operations. The horizontal axis shows production cycles. The Vertical axis shows WIP, production loss, and maintenance decision, respectively.**

TABLE 8. MSB indicator of each machine.

	M_1	M_2	M_3	M_4	M_5
n_i	0	148	3514	13336	54018

TABLE 9. USMD and MSB after each improvement of M_5.

	Before improvement	1st improvement	2nd improvement	3rd improvement	4th improvement
USMD (Parts)	4083	3132	2048	1742	1683
MSB	M_5	M_5	M_5	M_4	M_4

TABLE 10. Parameters of the random generated production systems.

Parameters	Values
Number of Machines	$\{6, 8\}$
Number of Machine States	{3,4}
Rated Speed	{1,2}
Transition Probability	$p_{i,i} \in \{0.8,0.9\}$, $\sum_j p_{i,j} = 1$, and $p_{i,j} \geq 0$
Transition Probability with Predictive Maintenance	$\beta_{i,i} \in \{0.7,0.8\}$, $\sum_j \beta_{i,j} = 1$, and $\beta_{i,j} \geq 0$
Transition Probability with Corrective Maintenance	$\gamma_{i,i} \in \{0.85,0.95\}$, $\sum_j \gamma_{i,j} = 1$, and $\gamma_{i,j} \geq 0$
Buffer Capacity	$b_i \in \{5, 10, 15, 20\}$

TABLE 11. System throughput improvement by three improvement methods.

	Bottleneck	WMIM	RMIM
Improvement	9.08%	2.98%	2.83%

maintenance. The policy strives to reduce the reactive maintenance events and hence reduce the maintenance cost.

• *Time-based policy (TBP):* Preventative maintenance is periodically performed on machine $M_l, 1 \leq l \leq 5$. This policy is usually determined based on experience or equipment maintenance manual instead of actual machine health conditions. In this case, we use simulation to search for the maintenance frequency for each machine that leads to the lowest discounted system cost.

• *Greedy policy (GP):* It is also denoted as reactive maintenance only policy. All the machines are made to produce parts until they reach the breakdown states. No predictive maintenance is applied.

Each policy is simulated for 10,000h in the production system. Table 7 summarizes the results. It can be observed that the proposed RLP outperforms all the other policies in terms of mean maintenance cost, mean inventory cost, and mean backlog cost. To be specific, RLP causes 9.27% less maintenance cost, 0.08% less inventory cost, and 1.43% less backlog cost than SBP. It causes 9.27% less maintenance cost, 0.43% less inventory cost, and 1.46% less backlog cost than TBP. It causes 9.95% less maintenance cost, 2.99% less inventory cost, and 3.11% less backlog cost than GP.

The distinctive characteristics allow the proposed RLP to achieve the best performance among all the 4 policies. Figure 2 plots the trajectories of WIP inventory, the production loss of the last virtual machine, and the maintenance decision of each machine with RLP. The results are discussed as follows:

1. Maintenance actions are scheduled together. When machine $M_l, 2 \leq l \leq 5$, is stopped for maintenance, there exist opportunities to perform maintenance on machines $M_1, \ldots, M_{l-1}$ at the same time. On the one hand, turning off the machines will not result in additional production loss [30]. On the other hand, it helps to prevent the system from accumulating excess inventory, which leads to high inventory cost.

2. Predictive maintenance is scheduled more frequently on upstream machines. Machine $M_l, 1 \leq l \leq 5$, can cause the production loss of the last virtual machine M_6 if all the buffers between machines M_l and M_6 being empty, i.e. $\sum_{k=l}^{5} b_k(t) = 0$. At any moment, it takes less time for machine M_l to cause the production loss than its upstream machines $M_1, \ldots, M_{l-1}$. Therefore, stopping a downstream machine has more risk in causing the production loss of the last virtual machine M_6, which leads to high backlog cost.

3. Predictive maintenance is scheduled when the inventory level is sufficiently high. This indicates that RLP can balance the production and maintenance needs of the production system. When the system has a low inventory level, performing predictive maintenance can lead to high production loss and cause high backlog cost. When the inventory level is high, stopping a machine for predictive maintenance cannot only reduce the risk of high reactive maintenance cost, but also help to decrease inventory cost. RLP helps the production system to capture the opportunities, and optimize the system cost.

B. BOTTLENECK IDENTIFICATION AND IMPROVEMENT

The effect of improving MSB is investigated. The serial production system in the previous section is still adopted. The production line is simulated for 10,000h to generate production data. Since the end-of-line machine M_6 models the market demand, the n_l of machines M_1 to M_5 is calculated and demonstrated in Table 8. It is found that machine M_5 has the greatest n_l, and the machine is MSB. In addition, Definition 1 is also utilized to identify MSB. Both methods identify the same MSB.

The DTD of machine M_5 is reduced by 20%, and the process is repeated for four times. Table 9 records the USMD and the MSB before and after the improvement. The result indicates that the USMDs in the first two improvements are significantly higher than the USMDs in the last two improvements. MSB also transfers from machines M_5 to M_4 after the third improvement. It is because when machine M_5 is MSB, reducing its DTD can most effectively reduce USMD. And when MSB transfers to another machine, continuously reducing the DTD of machine M_5 cannot effectively reduce USMD.

To further validate the bottleneck identification method, 10, 000 production systems are randomly generated. The parameters of the production systems are randomly selected from Table 10. The identification indicator proposed in Proposition 1 is adopted to find bottlenecks. The production data are generated by simulations, that have a simulation time of 10,000 h. Definition 1 is also utilized to find bottlenecks. The results show that the methods find identical bottlenecks in all the cases.

A bottleneck improvement method (BIM) can be established by identifying and improving bottlenecks. The improvement method is summarized as follows.

1) Collect production information and compute the value of n_i.
2) Reduce the DTD of MSB by 5%.
3) Repeat step 2 until MSB transfers to another machine.

The method is compared with two improvement methods. The first method is denoted as the worst machine improvement method (WMIM), where the DTD of the machine with the highest breakdown frequency is reduced by the same percentage as that in BIM. The second method is denoted as random machine improvement method (RMIM), where the DTD of a randomly selected machine is reduced by the same percentage as that in BIM. The improvement using the three methods are demonstrated in Table 11. It is shown that the proposed iterative process can lead to the most throughput improvement, which is about 3.05 times higher than the improvement by WMIM, and is 3.21 times higher than the improvement by RMIM.

IX. CONCLUSION AND FUTURE WORK

This paper establishes an integrated decision model for joint production and maintenance decision making. A Markov chain model is developed to investigate the transient behavior of the production system. A reinforcement learning approach is proposed to optimize the production and maintenance cost. This research integrates production system modeling and approximate dynamic programming. It establishes a systematic approach of dynamic decision making in multistage production systems. The presented control method is compared with other three commonly applied maintenance policies. The result indicates that the proposed control method can balance the production and maintenance needs of the production system. It leads to the lowest production and maintenance cost, which is 5.25% lower than SBP, 7.96% lower than TBP, and 8.87% lower than GP. The research also investigates machine stoppage bottleneck. The numerical studies show that the identification and mitigation of machine stoppage bottleneck can effectively improve system throughput (approximate 9.00% improvement). It leads to an improvement that is about 3 times higher than both WMIM and RMIM.

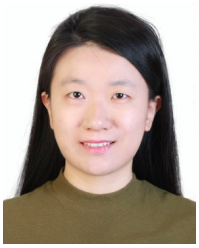
The research presents some managerial insights into the predictive maintenance decision-making and continuous improvement in the production systems. Firstly, when machine M_l is stopped for maintenance, there are maintenance opportunity windows for all the machines in the upstream of M_l. On the plant floor, operation managers can schedule the maintenance work of the machines together without causing excessive production loss. Secondly, predictive maintenance decision-making is closely related to the occupancies of the buffers. Plant floor managers should consider to performance predictive maintenance when the buffer levels between the machine and end-of-line machine are sufficiently high. Thirdly, when plant floor managers improve the performance of production systems through bottleneck identification and improvement, they should continuously identify bottlenecks, and only improve the performance of the current bottlenecks.

In the research, the standard feedforward neural network and stochastic gradient descent method can be improved to achieve higher computation efficiency and better maintenance policies. In addition, the current model assumes discrete machine deterioration states. However, it is not uncommon that a machine's health condition is described with continuous functions. For example, remaining useful life has been widely accepted as an indicator to denote machines' health condition. The proposed decision model should be extended to optimize production and maintenance when machines have continuous health states. We also plan to apply the decision model to broader areas. For instance, it has been proved in [31] that machines can be temporarily turned off for energy saving. The decision model will be extended to optimize the energy saving control such that energy efficiency improvement can be achieved without causing additional production loss. These will be the future work of the research.

REFERENCES

[1] Y. Li, Q. Chang, G. Xiao, and S. Biller, "Event-based modeling of distributed sensor networks in battery manufacturing," *Int. J. Prod. Res.*, vol. 52, no. 14, pp. 4239–4252, 2013.

[2] S. Takata, F. Kirnura, F. J. V. Houten, E. Westkamper, M. Shpitalni, D. Ceglarek, and J. Lee, "Maintenance: Changing role in life cycle management," *CIRP Ann. Manuf. Technol.*, vol. 53, no. 2, pp. 643–655, 2004.
[3] A. Raza and V. Ulansky, "Modelling of predictive maintenance for a periodically inspected system," *Proc. CIRP*, vol. 59, pp. 95–101, Jan. 2017.
[4] R. Ahmad and S. Kamaruddin, "An overview of time-based and condition-based maintenance in industrial application," *Comput. Ind. Eng.*, vol. 63, no. 1, pp. 135–149, 2012.
[5] A. K. S. Jardine, D. Lin, and D. Banjevic, "A review on machinery diagnostics and prognostics implementing condition-based maintenance," *Mech. Syst. Signal Process.*, vol. 20, no. 7, pp. 1483–1510, Oct. 2006.
[6] E. Auschitzky, M. Hammer and A. Rajagopaul, "How big data can improve manufacturing," McKinsey Company, New York, NY, USA, 2014, p. 822. [Online]. Available: http://www.mckinsey.com/insights/operations/how_big_data_can_improve_manufacturing
[7] A. V. Horenbeek and P. Liliane, "A dynamic predictive maintenance policy for complex multi-component systems," *Rel. Eng. Syst. Saf.*, vol. 120, pp. 39–51, Dec. 2013.
[8] C. S. Wong, F. T. S. Chan, and S. H. Chung, "A joint production scheduling approach considering multiple resources and preventive maintenance tasks," *Int. J. Prod. Res.*, vol. 51, no. 3, pp. 883–896, Feb. 2013.
[9] N. M. Najid, M. Alaoui-Selsouli, and A. Mohafid, "An integrated production and maintenance planning model with time Windows and shortage cost," *Int. J. Prod. Res.*, vol. 49, no. 8, pp. 2265–2283, Apr. 2011.
[10] P. Zhang, X. Zhu, and M. Xie, "A model-based reinforcement learning approach for maintenance optimization of degrading systems in a large state space," *Comput. Ind. Eng.*, vol. 161, Nov. 2021, Art. no. 107622.
[11] H. Yang, W. Li, and B. Wang, "Joint optimization of preventive maintenance and production scheduling for multi-state production systems based on reinforcement learning," *Rel. Eng. Syst. Saf.*, vol. 214, Oct. 2021, Art. no. 107713.
[12] Y. Chen, Y. Liu, and T. Xiahou, "A deep reinforcement learning approach to dynamic loading strategy of repairable multistate systems," *IEEE Trans. Rel.*, early access, Jan. 25, 2021, doi: 10.1109/TR.2020.3044596.
[13] S. Peng and Q. Feng, "Reinforcement learning with Gaussian processes for condition-based maintenance," *Comput. Ind. Eng.*, vol. 158, Aug. 2021, Art. no. 107321.
[14] Y. Li, Q. Tang, Q. Chang, and M. P. Brundage, "An event-based analysis of condition-based maintenance decision-making in multistage production systems," *Int. J. Prod. Res.*, vol. 55, no. 16, pp. 4753–4764, Aug. 2017.
[15] S. M. R. Iravani and I. Duenyas, "Integrated maintenance and production control of a deteriorating production system," *IIE Trans.*, vol. 34, no. 5, pp. 423–435, May 2002.
[16] T. Xia, L. Xi, X. Zhou, and J. Lee, "Condition-based maintenance for intelligent monitored series system with independent machine failure modes," *Int. J. Prod. Res.*, vol. 51, no. 15, pp. 4585–4596, Aug. 2013.
[17] Q. Chang, J. Ni, P. Bandyopadhyay, S. Biller, and G. Xiao, "Maintenance opportunity planning system," *J. Manuf. Sci. Eng.*, vol. 129, no. 3, pp. 661–668, Jun. 2007.
[18] P.-H. Cui, J.-Q. Wang, and Y. Li, "Data-driven modelling, analysis and improvement of multistage production systems with predictive maintenance and product quality," *Int. J. Prod. Res.*, vol. 2021, pp. 1–18, Sep. 2021.
[19] L. Zhang, C. Wang, J. Arinez, and S. Biller, "Transient analysis of Bernoulli serial lines: Performance evaluation and system-theoretic properties," *IIE Trans.*, vol. 45, no. 5, pp. 528–543, 2013.
[20] Y. Li, Q. Chang, M. P. Brundage, S. Biller, J. Arinez, and G. Xiao, "Market demand oriented data-driven modeling for dynamic manufacturing system control," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 1, pp. 109–121, Jan. 2014.
[21] J. Li, "Performance analysis of manufacturing systems with rework loops," *IIE Trans.*, vol. 36, no. 8, pp. 755–765, 2004.
[22] S. M. Meerkov and L. Zhang, "Product quality inspection in Bernoulli lines: Analysis, bottlenecks, and design," *Int. J. Prod. Res.*, vol. 48, no. 16, pp. 4745–4766, 2010.
[23] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, pp. 529–533, Feb. 2014.
[24] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.
[25] P. McDonnell, S. Joshi, and R. G. Qiu, "A learning approach to enhancing machine reconfiguration decision-making games in a heterarchical manufacturing environment," *Int. J. Prod. Res.*, vol. 43, no. 20, pp. 4321–4334, Oct. 2005.
[26] B. C. Csáji, L. Monostori, and B. Kádár, "Reinforcement learning in a distributed market-based production control system," *Adv. Eng. Informat.*, vol. 20, no. 3, pp. 279–288, Jul. 2006.
[27] G. Tesauro, "Temporal difference learning and TD-Gammon," *Commun. ACM*, vol. 38, no. 3, pp. 58–68, Mar. 1995.
[28] J. Shahrabi, M. A. Adibi, and M. Mahootchi, "A reinforcement learning approach to parameter estimation in dynamic job shop scheduling," *Comput. Ind. Eng.*, vol. 110, pp. 75–82, Aug. 2017.
[29] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85–117, Jan. 2014.
[30] C. Qing, G. Xiao, S. Biller, and L. Li, "Energy saving opportunity analysis of automotive serial production systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 10, no. 2, pp. 334–342, Apr. 2013.
[31] Y. Li, Q. Chang, J. Ni, and M. Brundage, "Event-based supervisory control for energy efficient manufacturing systems," *IEEE Trans. Automat. Sci. Eng.*, vol. 15, no. 1, pp. 92–103, May 2018.



MAOMAO FENG received the Ed.D. degree from Binghamton University, Binghamton, in 2018. Currently, she is a Lecturer with the School of Foreign Studies, Chang'an University, Xi'an, China. Her research interests include modeling and analysis of engineering systems, and quantitative/qualitative research.

YANG LI received the B.S. degree in process equipment and control engineering from Xi'an Jiaotong University, Xi'an, Shaanxi, China, in 2010, and the Ph.D. degree in mechanical engineering from Stony Brook University, Stony Brook, in 2014.

Currently, he is an Associate Professor with the Department of Industrial Engineering, School of Mechanical Engineering, Northwestern Polytechnical University, Xi'an. He has participated in many different projects related to production system modeling and continuous improvement. His research interests include analysis and real-time control of production systems, and modeling and analysis of supply-chain systems.