IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal
IEEE RELIABILITY SOCIETY SECTION

Received August 20, 2021, accepted September 6, 2021, date of publication September 9, 2021, date of current version September 17, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3111575

Colored Resource-Oriented Petri Nets for Deadlock Control and Reliability Design of Automated Manufacturing Systems

ADEL AL-SHAYEA[1], HUSAM KAID[1], ABDULRAHMAN AL-AHMARI[1], EMAD ABOUEL NASR[1,2], ALI K. KAMRANI[3], AND HAITHAM A. MAHMOUD[1,2]

[1]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[2]Mechanical Engineering Department, Faculty of Engineering, Helwan University, Cairo 11732, Egypt
[3]Department of Industrial Engineering, College of Engineering, University of Houston, Houston, TX 77204, USA

Corresponding authors: Husam Kaid (yemenhussam@yahoo.com) and Emad Abouel Nasr (eabdelghany@ksu.edu.sa)

This work was supported by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, under Award 14-ELE69-02.

ABSTRACT Reliability modeling and deadlock control have become important issues in automated manufacturing systems (AMSs) with reliable and unreliable resources. Therefore, this paper develops a reliability modelling approach and deadlock control based on colored resource-oriented timed Petri net (CROTPN) and neural networks to find crucial reliability measures in AMSs. In the first step, a CROTPN with considering resource failures is developed to obtain ‘‘sufficient and necessary conditions’’ for the CROTPN liveness. In the second step, a fault diagnosis and treatment approach is proposed that integrates the obtained unreliable net with neural networks to ensure that the system is reliable. Furthermore, a neural unreliable CROTPN is used to analyze the system reliability. A simulation is carried out to illustrate the approach and compare the results to those found in the literature. The developed approach has been proved to be simpler in structure and can solve the deadlock problem and model AMS reliability.

INDEX TERMS Failures, deadlock avoidance, reliability, neural network, colored Petri net, automated manufacturing system.

NOTATIONS

MTTF	Mean time to failure.
MTTR	Mean time to repair.
N	A colored resource-oriented timed Petri net.
N_{RNi}	A marked recovery subnet.
N_U	an unreliable CROTPN.
N_{NP}	A neural Petri net model.
N_{NU}	A neural unreliable CROTPN.
PPCs	Production process circuits.
n	No. of transitions.
m	No. of places.
π	No. of part types.
ε_l	No. of part paths.
θ	No. of PPCs.
μ	No. of input neurons in the N_{NP}.
α	No. of output neurons in the N_{NP}.
β	No. of fault detection transitions in the N_{NP}.
γ	No. of fault treatment transitions in the N_{NP}.
ζ	No. of inputs neuron pattern in the N_{NP}.
i	The index of a place, $i = 1, 2, \ldots, m$.
j	The index of a transition, $j = 1, 2, \ldots, n$.
l	The index of part types, $l = 1, 2, \ldots, \pi$.
z	The index of part paths, $z = 1, 2, \ldots, \varepsilon_l$.
r	The index of PPCs, $r = 1, 2, \ldots, \theta$.
ii	The index of input neurons, $ii = 1, 2, \ldots, \mu$.
jj	The index of output neurons, $jj = 1, 2, \ldots, \alpha$.
ll	The index of fault detection transitions, $ll = 1, 2, \ldots, \beta$.
zz	The index of fault treatment transitions, $zz = 1, 2, \ldots, \gamma$.
rr	The index of inputs neuron pattern, $rr = 1, 2, \ldots, \zeta$.
p_o	The single idle place.
p_r	The single transportation resource.
p_{xii}	The input neuron of the N_{NP}.

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

p_{yjj}	The output pattern of the N_{NP}.
P_R	The resource places set.
T	The transitions set.
$C(p)$	The colors set of a place p.
$C(t)$	The colors set of a transition t.
u_i	The number of color types in a place p_i.
v_j	The number of color types in a transition t_j.
a_{iui}	A color type ui in a place p_i.
b_{jvj}	A color type v_j in a transition t_j.
$I(p, t)$	The input function of N.
$O(p, t)$	The output function of N.
D	The delay function, which adds a firing delay $D(t_j)$ to each transition t_j.
K	The capacity function, which adds the maximum tokens number $K(p_i)$ to each place p_i.
M_o	The initial marking function, which assigns tokens to each place p_i with color a_{iui}.
M_{RNio}	The initial marking function of a marked recovery subnet.
M_{Uo}	The function of an unreliable CROTPN initial marking.
M_{NPo}	The function of a neural Petri net initial marking.
M_{NUo}	The function of a neural unreliable CROTPN initial marking.
e_r	A circuit in PPCs.
e_r^c	An interactive subnet in the CROTPN that consisting of c PPCs, $c > 0$.
$P(e_r)$	The places set in an e_r.
$T(e_r)$	The transitions set in an e_r.
$R(N, M_o)$	The reachable markings set of the CROTPN.
$R_L(N, M_o)$	The reachable markings set of the controlled CROTPN.
$M(p_i, e_r)$	The tokens number in a place p_i, which enables t_j in an e_r.
$M(p_i, e_r^c)$	The tokens number in a place p_i, which enables t_j in an e_r^c.
$\eta(e_k^n, M)$	The enabled PPCs in an e_r^c.
$P(e_r^c)$	The places set in an e_r^c.
$T(e_r^c)$	The transitions set in an e_r^c.
$T_I(e_r)$	The input transitions set in an e_r.
$T_O(e_r)$	The output transitions set in an e_r.
$T_I(e_r^c)$	The input transitions set in an e_r^c.
$T_O(e_r^c)$	The output transitions set in an e_r^c.
$S'(e_r)$	The current spaces number in an e_r.
$S(e_r)$	The free spaces number in an e_r.
$V_{en}(t_j)$	The PPCs set in the CROTPN, where t_j be an input transition of these PPCs.
T_d	The transitions set in the CROTPN.
r_u	An unreliable resource in CROTPN.
$p_{combined}$	The single recovery place.
t_{fi}	The failure transition of a place p_i.
t_{ri}	The recovery transition of a place p_i.
t_{dll}	The fault detection transition in the neural model.
t_{tzz}	The fault treatment transition in the neural model.
c_{rni}	The color that maps p_i into colors.
T_F	The failure transitions set of an unreliable CROTPN.
T_R	The recovery transitions set of an unreliable CROTPN.
X_ζ	The inputs neuron patterns set ζ of the neural model.
Y_ζ	The outputs set of patterns ζ of the neural model.
Z_{ll}	The input of fault detection neuron layers of the neural model.
δ	The number of elements in the input learning pattern X_ζ that are equal to 1.
T_{uptime}	Up time of the neural unreliable CROTPN.
$T_{downtime}$	Down time of the neural unreliable CROTPN.
$N_{failures}$	Total number of occurred failures in the neural unreliable CROTPN.
T_{MTTF}	MTTF of the neural unreliable CROTPN.
T_{MTTR}	MTTR of the neural unreliable CROTPN.
As	Availability of the neural unreliable CROTPN.

I. INTRODUCTION

Deadlock control and reliability analysis are important features in the development and management of AMSs. Some operations in AMSs cannot be carried out due to deadlock if resources are shared. Therefore, AMSs need to control the deadlock. Furthermore, resource failures can lead to new deadlocks. The system reliability is significantly affected by failures, which in turn dictates the system's efficiency. To detect and treat failures and improve the reliability of the AMS, it is important to design an approach in the AMS with unreliable resources.

An effective mathematical modeling tool to control and analyze deadlock [1], [2] and reliability [3]–[5] in AMSs is Petri Nets (PNs). PNs are used to describe the dynamic behaviors of AMSs, such as sequencing, synchronization, concurrency, and conflict. In the literature, several approaches are proposed based on Petri nets, which focus on the detection and recovery of deadlocks, the prevention of deadlocks, and the prevention of deadlocks [6], [7]. Most of these approaches suppose resources are reliable [8]–[15], and others suppose they are unreliable [16]–[27]. For designing deadlock control approaches there are two analysis techniques in PNs: the reachability graph [28]–[30] and the structural analysis [1], [8]. Three criteria, including the computational complexity [1], [31], structural complexity [1], [9], [32], and behavioral permissiveness [6], [27], [33], are required to develop the AMS supervisor.

Over the last several years, various techniques are developed on deadlock prevention and detection and treatment of faults of unreliable resources in AMSs [18], [20], [25], [34]–[38]. Neural networks are also used to detect and treat faults [39]–[46]. In addition, several quantitative analysis techniques and tools are used for system reliability modeling and analysis, such as the fault tree analysis (FTA), the HAZard and OPerability study (HAZOP), the reliability block diagram (RBD), the Markov analysis, and the failure modes effects and criticality analysis (FMECA).

It is well-known that several deadlock control and reliability estimation methods have been proposed. However, the disadvantage of these methods is that there are no methods can propose an integrated approach to solve the deadlock control and reliability estimation. Therefore, the aim of this paper is to propose an approach for deadlock control and reliability modeling based on CROTPN and neural networks to find important reliability measures in AMSs, including MTTF, MTTR, and availability. First, a CROTPN [47]–[64] with considering resource failures is developed to ensure that the CROTPN is live. In the second step, an approach is proposed for detection and treatment of faults, which combines neural networks with the CROTPN to ensure that the system is reliable. In addition, using neural unreliable CROTPN, the system reliability are analyzed. The main research contributions are listed below.

1. A new solution is designed for reliability modelling and deadlock control of complex AMSs.
2. The proposed strategy represents a neural unreliable CROTPN for detection and treatment of faults in AMSs.
3. A general simulation code for the proposed CROTPN is designed with the GPenSIM Tool, which is used to model, validate and performance comparisons.

The rest of this research is organized as follows. In Section II, the CROTPN construction and its deadlock avoidance policy are presented. The unreliable CROTPN and neural networks integration are shown in Section III for faults detection and treatment. The reliability model of a neural unreliable CROTPN is illustrated in In Section IV. Section V illustrates the AMS example from the literature, which shows the experimental results of the proposed approach. Section VI shows the conclusions and future work of the research.

II. DESIGN OF CROTPN AND ITS DEADLOCK AVOIDANCE POLICY

The CROTPN is an extended version of Petri nets that consisting of places (sketched by circles), transitions (sketched by bars), arcs (connect places to transitions or transitions to places), tokens (dots), and colors [53]. Each place contains a colored token describing the modeled system state. The formal definitions of the CROTPN and its deadlock avoidance policy are presented in this section.

Definition 1: Let N is said to be a colored resource-oriented timed Petri net (CROTPN) with $N = (P, T, C, I, O, D, K, M_o)$ if

1. $P = \{p_o\} \cup \{p_r\} \cup P_R$, is a finite set of places, where $P_R = \cup_{i\in m}\{p_i\}$;
2. $T = \cup_{j\in n}\{t_j\}$, is a finite set of transitions such that $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$;
3. $C(p)$ and $C(t)$ are the sets of colors associated with place $p \in P$ and transition $t \in T$. We let $C(p_i) = \{a_{i1}, a_{i2}, \ldots, a_{iui}\}$ and $C(t_i) = \{b_{j1}, b_{j2}, \ldots, b_{jvj}\}$ where $u_i = |C(p_i)|$ and $v_j = |C(t_i)|$;
4. $I(p, t)$: $C(p)\times C(t) \rightarrow \mathbf{IN}$ and $O(p, t)$: $C(p)\times C(t) \rightarrow \mathbf{IN}$, where $\mathbf{IN} = \{0, 1, 2, \ldots\}$;
5. $D: T \rightarrow \mathbf{TS}$, where $\mathbf{TS} > 0$;
6. $K: P \rightarrow \mathbf{IN}$;
7. $M: P \rightarrow N$ is a marking function, which assigns tokens to the places. $M = (M(p_1), M(p_2), \ldots, M(p_m))^T$. $M(p_i)$ represents the number of tokens in p_i, regardless of their color, while $M(p_i, a_{ij})$ represents the number of tokens in p_i that have the color a_{ij}. The initial marking is denoted by M_o.

Definition 2: Let (N, M_o) be a CROTPN with $N = (P, T, C, I, O, D, K, M_o)$. For a place $p \in P, {}^\bullet p = \{t: t \in T$ and $O(p, t) > 0\}$ is called the preset of p and $p^\bullet = \{t: t \in T$ and $I(p, t) > 0\}$ is called the postset of p. Similarly, t's preset ${}^\bullet t = \{p \in P: I(p, t) > 0\}$ and postset $t^\bullet = \{p \in P: O(p, t) > 0\}$. This notation can be extended to a set of nodes as follows: given a set $S \subseteq P \cup T$, the preset and postset of S are respectively defined as ${}^\bullet S = \cup^\bullet_{a\in S} a$ and $S^\bullet = \cup_{a\in S} a^\bullet$.

The transitional rules for enabling and firing can be presented as below.

Definition 3: Let (N, M_o) be a CROTPN. A transition t_j is called a process-resource-enabled if

$$M(p_i, a_{ih}) \geq I(p_i, t_j)(a_{ih}, b_{jk}), \forall p_i \in P, \forall p_i \in^\bullet t_j, \quad a_{ih} \in C(p_i), \quad b_{ik} \in C(t_j) \tag{1}$$

and

$$K(p_i) \geq M(p_i, a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk}) - I(p_i, t_j)(a_{ih}, b_{jk}), \quad \forall p_i \in P, \quad \forall p_i \in t_j^\bullet, \ a_{ih} \in C(p_i), \ b_{ik} \in C(t_j) \tag{2}$$

Definition 4: Let (N, M_o) be a CROTPN. If the t_j is enabled at marking M, then it can fire and the marking transforms from M to M' (expressed by $M[t_j\rangle M'$) as follows.

$$M'(p_i, a_{ih}) = M(p_i, a_{ih}) + O(p_i, t_j)(a_{ih}, b_{jk}) - I(p_i, t_j)(a_{ih}, b_{jk}), \quad \forall p_i \in P, \ a_{ih} \in C(p_i), \ b_{ik} \in C(t_j) \tag{3}$$

The parts processing paths in a CROTPN are defined as the following.

Definition 5: Let $N = (P, T, C, I, O, D, K, M_o, PP)$ be a CROTPN, where $PP = \{PP_1, PP_2, PP_3, \ldots, PP_\pi\}$ represents all processing paths of all part types π. PR_l is $R_{lo} \rightarrow R_{l1} \rightarrow R_{l2} \rightarrow \cdots\cdots \rightarrow R_{lz} \rightarrow R_{lo}$. R_{lo} represents a load/unload station of the CROTPN, and $R_l(l \neq 0)$ represents a resource. The operation path begins at R_o and ends at R_o. If the part shifts from R_{l1} to R_{l2}, a transportation resource is needed to move the part.

Due to the high connectivity of the CROTPN, it has several circuits. A production process circuits are CROTPN special circuits and play a critical role in the CROTPN's liveness. In production process circuits, there is no idle place p_o and represented as PPCs $= \{e_1, e_2, \ldots, e_r\}$. If a circuit e_r moves from a node a, via numerous nodes, then returns to the initial node a, and no node is duplicated, it is called an elementary circuit. When a circuit e_r does not back to the starting node a, then the circuit e_r is called a nonelementary. In addition, the places number in e_r must be equal to the number of transitions on e_r such that $|P(e_r)| = |T(e_r)|$, and the transition input places for e_r ($^{\bullet}t_j \in P(e_r)$, $p_i \in {}^{\bullet}t_j$) must be on e_r.

When a transition t_j in e_r is fired and the tokens depart e_r, the tokens in e_r are said to be the departing tokens, and called the cycling tokens if the tokens do not depart e_k, and can be denoted as

$$M(e_r) = \Sigma M(p_i, e_r), \quad p_i \in P(e_r) \tag{4}$$

A circuit e_r^c is called interactive if its places and transitions are shared with at least another PPC and it is strongly connected. If a transition t_j in e_r^c ($t_j \in (p_i^{\bullet} \cap T(e_r^c))$ and $p_i \in P(e_r^c)$) is fired and the tokens depart e_r^c, the tokens in e_r^c are said to be the departing tokens, and called the cycling tokens if the tokens do not depart e_r^c, can be denoted as

$$M(e_r^c) = \Sigma M(p_i, e_r^c), \quad p_i \in P(e_r^c) \tag{5}$$

A circuit e_r has no free space in its places at marking M if

$$\Sigma M(p_i) = \Sigma K(p_i) = K(e_r), \quad p_i \in P(e_r) \tag{6}$$

The control policy and necessary conditions of the deadlock-free in a CROTPN are presented in the following theorems.

Theorem 1: Let (N, M_o) be a CROTPN. N is not live if

$$M_o(p_o) \geq K(e_r). \tag{7}$$

Proof: See [52].

In Definition 3, when conditions 1 and 2 are achieved, a transition t_j is said to be a controlled transition. N is said to be a controlled net when it has at least one controlled transition. If a circuit e_r in the N is process-resource-enabled, then it called enabled. If a transition $t_j \in T(e_r)$ is live, then a circuit e_r is said to be a live transition. When a transition $t_j \notin T(e_r^c)$ and $t_j^{\bullet} \in P(e_r^c)$, then t_j is called an input transition of e_r^c. If a transition $t_j \notin T(e_r^c)$ and $^{\bullet}t_j \in P(e_r^c)$, then t_j is said to be an output transition of e_r^c.

The S(e_r) and S$'$ (e_r) in a circuit e_r can be formulated as

$$S(e_r) = \sum\nolimits_{p_i \in P(\mathbf{e_r})} (K\,(p_i) - M(p_i)) \tag{8}$$

$$S'(e_r) = K(e_r) - M(e_r) \tag{9}$$

Theorem 2: Let (N, M_o) be a CROTPN. A circuit e_r is live at any marking $M \in R_L(N, M_o)$ if

$$S'(e_r) \geq 1. \tag{10}$$

Proof: See [52].

Wu and Zhou [65] presented a deadlock-free control policy (DFC-Policy) based on the given condition in Theorem 2 condition, which makes the model deadlock-free.

A. DFC-POLICY [65]

At any reachable marking M, transitions in $T_I(e_r)$ and $T(e_r)$ are controlled if the condition given in Theorem 2 is satisfied.

Theorem 3: Let (N, M_o) be a CROTPN. A circuit e_r^c is live at any marking $M \in R_L(N, M_o)$ reachable from M_o if

$$\text{for any } e_r, \; S'(e_k) \geq 1, \tag{11}$$

and

$$\eta(e_r^c, M) \geq 1 \tag{12}$$

Proof: See [52].

Theorem 4: Let (N, M_o) be a CROTPN. N is always live if it has no the PPC.

Proof: See [52].

Places and transitions inside the interactive subnet e_r^c may be shared. These shared places are connected together by shared transitions, forming a shared direct place path (SDPP). The first (last) place on the SDPP contains input (output) transitions from several PPCs. These are referred to as intercircuit input (output) transitions (IITs and IOTs) [52].

Theorem 5: Let (N, M_o) be a CROTPN. A circuit e_r^c is live at marking M if the following conditions are satisfied

(a) any transition $t_j \in T_I(e_r^c)$ and any IIT in the subnet are controlled;
(b) before a controlled transition t_j fires, for any $S'(e_r) \geq 2$, $e_r \in V_{en}(t_j)$;
(c) the marking M is changed to M' after t_j fires, such that $\eta(e_r^c, M') \geq 1$.

Proof: See [52].

If the control law provided in Theorem 5 is implemented, then firing any $t_j \in T_I(e_r^c)$ or IIT ensures that 3 is Theorem achieved. As a result of Theorem 3, such firing ensures the subnet's liveness. Moreover, no other transitions can be used to move tokens and spaces from or to a PPC in the subnet. As a result, the firing of other transitions has no effect on the subnet's liveness.

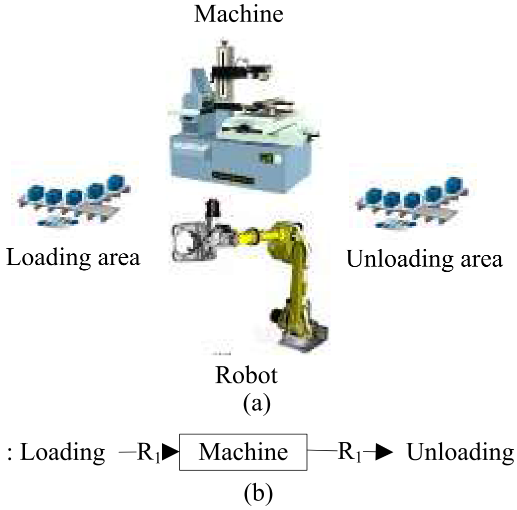

FIGURE 1. **(a) AMS example [66] and (b) The path of operation.**

Algorithm 1 Modeling of the CROTPN

Input: $PP_l: = R_{lo} \to R_{l1} \to R_{l2} \to \cdots\cdots \to R_{lz} \to R_{lo}$ and the operation duration $\mathrm{D}(t_{lz})$ to perform process R_{lz};.
Output: The CROTPN model;
Initialization: Design the common idle place p_o, the common material handling resource p_r and their initial marking M_o, i.e., $P = \{p_o, p_r\}$, $M_o = \{c_{p1}, c_{p2}, \ldots c_{pl}, c_{t1}, c_{t2}, c_{t3}, \ldots\}$ and $l = 0$, and $z = 0$;

for all $1 \leq l \leq \pi$ ***do***
  for all $1 \leq z \leq \varepsilon_i$ ***do***
    Design a place p_x for $R_{l(z-1)}$, i.e., $P := P \cup \{p_x\}$;
    Design a place p_y, which represents the machine that will perform the operation R_{lz}, i.e., $P := P \cup \{p_y\}$;
    Design a transition t_{xyl} to perform the operation R_{lz}, i.e., $T := T \cup \{t_{xyl}\}$, $D := D \cup \mathrm{D}(t_{lz})$;
    Design the arcs (p_x, t_{xyl}) and (t_{xyl}, p_y), i.e., $F := F \cup \{(p_x, t_{xyl}), (t_{xyl}, p_y)\}$;
      if t_{xyl} needs p_r to move part l, ***then***
        Design the arcs (p_r, t_{xyl}) and (t_{xyl}, p_r), i.e., $F := F \cup \{(p_r, t_{xyl}), (t_{xyl}, p_r)\}$;
      end if
  end for
end for

Consider the automated manufacturing system [66] presented in Figure 1(a) to show the steps of the CROTPN construction by using Algorithms 1 and 2. The system operation path is illustrated in Figure 1(b). After implementing Algorithm 1, Figure 2 illustrates the developed CROTPN. In Figure 2, we have a place p_1 representing machine 1, one transportation place p_r representing robot 1 (R1). Transitions t_{01A} and t_{10A} respectively indicate the loading and unloading part A to/from p_1. The initial state of the CROTPN is designed as $M_o(p_o) = \{c_{p1}\}$, which represents that there is a raw part A with color c_{p1} in the load/unload station p_o, $M_o(p_r) = \{c_{t1}\}$, which represents the robot 1 with color c_{t1}, and $M_o(p_1) = 0$, which represents the state of machine 1. The operation path is stated as: a raw part A is assigned to the p_o, the robot 1 loads a part to the machine 1 p_1 by a transition t_{01A}, then the robot 1 unloads the finished part by t_{10A} to the unload station p_0. The CROTPN's behavior shown in Figure 2 can be described as follows. If t_{01A} enabled, then it fires and chooses a token c_{p1} from p_{o} and a token c_{t1} from p_{r}. When t_{01A} fired, it places a token c_{p1} to p_1 and a token c_{t1} to p_{r}. Finally, if t_{10A} enabled, then it fires and chooses a token c_{p1} from p_1 and a token c_{t1} from p_{r}. When t_{10A} fired, it places a token c_{p1} to p_o and a token c_{t1} to p_{r}. Based on Algorithm 2, the CROTPN shown in Figure 2 has no the PPC. Therefore, it is live.

III. DESIGN OF UNRELIABLE CROTPN BASED ON NEURAL NETWORK

The failure of a resource is a temporal uncertainty problem in automated manufacturing systems. When a resource failure occurs, we attempt to design a recovery subnet that

Algorithm 2 Policy of Deadlock Avoidance for a CROTPN

Input: The obtained CROTPN from Algorithm 1 and PPCs $= \{e_1, e_2, \ldots, e_r\}$;
Output: The controlled CROTPN;

if there is the PPC ***then***
  for all $1 \leq r \leq \theta$ ***do***
    if the e_r is not an interactive, ***then***
      for all $0 \leq w \leq |R\ (N, M_o)|$ ***do***
        $p \in P(e_r)$, $K(e_r) = \Sigma K(p, e_r)$;
        $p \in P(e_r)$, $M_w(e_r) = \Sigma M_w(p, e_r)$;
        $S'\ (e_r) = K(e_r)$- $M_w(e_r)$;
        if $S'(e_r) \geq$, ***then***
          The e_r is live;
        else if
          The e_r is not live;
          Apply the DFC-Policy [65] to avoid the deadlock
      end for
  else if /* the e_r is an interactive */.
    $w = 0$;
    for all $0 \leq w \leq |R\ (N, M_o)|$ ***do***
      $p \in P(e_r)$, $K(e_r) = \Sigma K(p, e_r)$;
      $p \in P(e_r)$, $M_w(e_r) = \Sigma M_w(p, e_r)$;
      $S'(e_r) = K(e_r) - M_w(e_r)$;
      if $S'(e_r) \geq 1$ and $\eta(e_r^c, M_w) \geq$, ***then***
        The e_r is live;
      else if
        The e_r is not live;
        Apply the conditions in Theorem 5 to avoid the deadlock;
    end for
  end if
end for
else if
The e_r is live;
end if

can repair the failed resource. The resource can then be reused. Additionally, early fault detection and treatment are crucial for AMSs to operate efficiently, safely, and reliably. Therefore, the formal definitions are introduced in this section to develop a single recovery and detection and treatment nets for all failures in an AMS based on neural networks.

Definition 6: Let N be a CROTPN. Let $N_{RNi} = (\{p_i, p_{combined}\}, \{t_{fi}, t_{ri}\}, F_{rni}, c_{rni})$ be a single recovery net of $p_i \in P_R$ and M_{RNio} its initial markings, where $F_{rni} = \{(p_i, t_{fi}), (t_{fi}, p_{combined}), (p_{combined}, t_{ri}), (t_{ri}, p_i)\}$, $M_{RNio}(p_i) \geq 0$ and $M_{RNio}(p_{combined}) = 0$. The integration of CROTPN with the single recovery net leads to an unreliable net, denoted as $(N_U, M_{Uo}) = (N_{RNi}, M_{RNio}) \parallel (N, M_o)$, where $\parallel$ means the net composition of (N_{RNi}, M_{RNio}) and (N, M_o).

In Definition 6, $p_{combined}$ is called a single recovery place of all $p_i \in P_R$. Transitions t_{fi} and t_{ri} indicate that an

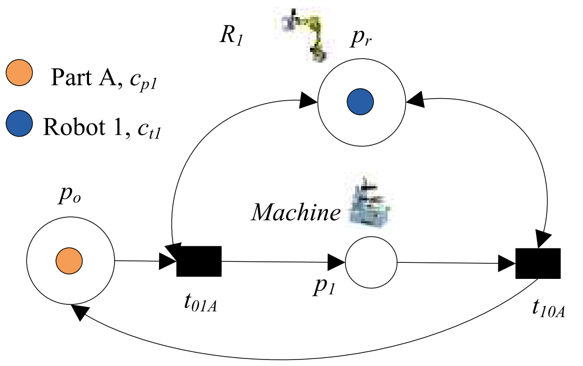

FIGURE 2. The CROTPN (N, M_o) of an AMS illustrated in Figure 1(a).

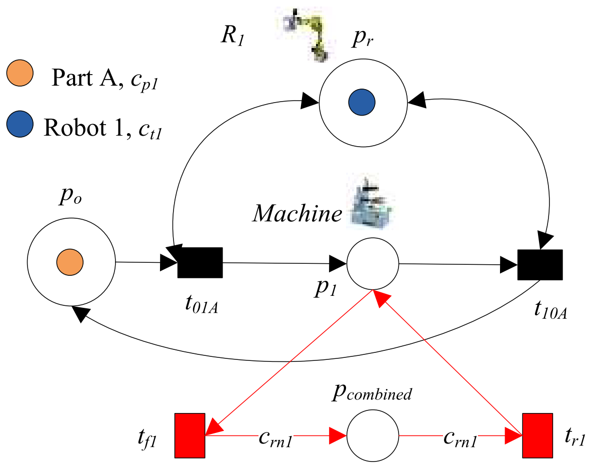

FIGURE 3. The unreliable CROTPN (N_U, M_{Uo}) of an AMS presented in Figure 2.

unreliable resource pi fails in p_i and recovers using $p_{combined}$, respectively. If an unreliable resource fails in p_i, the token in p_i moves into $p_{combined}$ by firing t_{fi}, which indicates that a resource failure happens; when the failed resource is repaired, the token in $p_{combined}$ moves into p_i by firing t_{ri}, indicating that a resource recovery is complete.

Definition 7: Let (N_U, M_{Uo}) be an unreliable CROTPN with $N_U = (P_U, T_U, C_U, I_U, O_U, D_U, K_U, M_{Uo})$ if

1. $P_U = P \cup \{p_{combined}\}$;
2. $T_U = T \cup T_F \cup T_R$, where $T_F = \cup_{i\in \mathbf{NP}}\{t_{fi}\}$, $T_R = \cup_{i\in \mathbf{NP}}\{t_{ri}\}$, and $\mathbf{NP} = \{i|p_i \in P_R\}$;
3. $C_U = C \cup C_F$, where $C_F = \cup_{i\in \mathbf{NP}}\{c_{rni}\}$;
4. $I_U(p_i, t_j)$: $C_U(p_i)\times C_U(t_j) \to \mathbf{IN}$ and $O_U(p_i, t_j)$: $C_U(p_i)\times C_U(t_j) \to \mathbf{IN}$;
5. D_U: $T_U \to \mathbf{TS}$;
6. K_U: $P_U \to \mathbf{IN}$;
7. M_{Uo}: $P_U \to \mathbf{IN}$.

Assume that the unreliable resource in the net presented in Figure 2 is $p1$. Adding recovery subnet for $p1$ by Definition 6 results in an unreliable CROTPN (N_U, M_{Uo}) presented in Figure 3, where $\mathbf{NP} = \{1\}$, $T_F = \{t_{f1}\}$, and $T_R = \{t_{r1}\}$, and $C_F = \{c_{rn1}\}$. If the machine 1 fails in p_1, then a transition t_{f1} enables and fires. If a transition t_{f1} fired, then it takes a token from p_1 and places a token c_{rn1} to $p_{combined}$. If the MTTR on machine 1 is elapsed, then a transition t_{r1} enables and fires. If a transition t_{r1} fired, then it takes a token c_{rn1} from $p_{combined}$ and places a token to p_1. Thus, the recovery maintenance on machine 1 is finished successfully.

Neural networks have been increasingly popular in recent years because they can learn complicated functions. Large numbers of simple and highly connected processors that comprise parallel and distributed processing systems can be described as neural networks. Using these networks, complex mapping such as identifying faults may be performed offline and can then be used successfully in the online environment. The following definitions provide the basis of the unreliable CROTPN model, which is based on neural networks for fault diagnosis and treatment.

Definition 8: Let N_{NP} be a neural model with $N_{NP} = (P_{NP}, T_{NP}, F_{NP}, X_{rr}, Y_{rr}, W_{NP}, M_{NPo})$ if

1. $P_{NP} = \cup_{ii\in\mu}\{p_{xii}\} \cup (\cup_{jj\in\alpha}\{p_{yjj}\})$;
2. $T_{NP} = \cup_{ll\in\beta}\{t_{dll}\} \cup (\cup_{zz\in\gamma}\{t_{tzz}\})$;
3. $F_N \subseteq (P_{NP} \times T_{NP}) \cup (T_{NP} \times P_{NP})$;
4. $X_{rr} = \cup_{ii\in\mu}\{x_{ii}^{rr}\}$, where each x_{ii} is assigned to the p_{xii};
5. $Y_{rr} = \cup_{jj\in\alpha}\{y_{jj}^{rr}\}$, where each y_{jj} is assigned to the p_{yjj};
6. $W_{NP} \to [0,1]$ represents the synaptic weight matrix of the neural network from p_{xii} to t_{dll}, and can be represented as

$$W_{NP} = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1\beta} \\ w_{21} & w_{22} & \cdots & w_{2\beta} \\ \vdots & \vdots & \vdots & \vdots \\ w_{\mu 1} & w_{\mu 2} & \cdots & w_{\mu\beta} \end{bmatrix} \quad (13)$$

where the constraint condition is:

$$\sum_{ll=1}^{\beta}\sum_{ii=1}^{\mu} w_{ii\,ll} = 1 \quad (14)$$

7. $M_{NPo} : P_{NP} \to \mathbf{IN}$.

The main difference between traditional neural networks and neural Petri nets (NPNs) is that in neural Petri nets, the Petri layer and transition layer indicate the configuration of the NPNs model of failure diagnosis and treatment. The Petri layer's input x_{ii}^{rr} (actual devices failures) represents the NPNs' input, and the output of each node in this layer represents tokens with acquisition systems that collect signals from input sensors and a connectivity matrix W_{NP}.

Definition 9: Let (N_{NP}, M_{NPo}) be a neural model. If a transition t_{dll} is fired, then the Z_{ll} can be formulated as:

$$Z_{ll} = \sum_{ii}^{\mu} w_{ii\,ll} x_{ii}^{rr} \quad (rr = 1, 2 \ldots, \zeta \text{ and } ll = 1, 2 \ldots, \beta) \quad (15)$$

The transition layer's input is the Petri layer's output, which is connected to the neural network's middle layer.

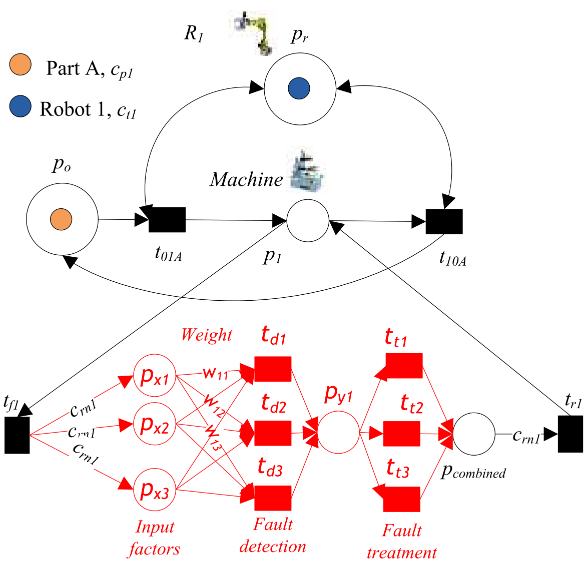

FIGURE 4. **The neural unreliable CROTPN (N_{NU}, M_{NUo}) of an AMS presented in Figure 3.**

This layer is designed to create tokens in accordance with the following competition laws:

Definition 10: Let (N_{NP}, M_{NPo}) be a neural model. If the value of Z_{ll} is the largest relative to the values of the other inputs, it is called the winner and its output value y_{jj} is indicated as 1; and the other y_{jj} output values are indicated as 0, expressed by

$$\left[\begin{array}{l} y_{ll} = 1\ Z_{ll} > Z_i\ (ll \text{ and } i = 1, 2, \ldots, \beta;\ i \neq ll) \\ \qquad\qquad y_i = 0\ (i \neq ll) \end{array}\right. \tag{16}$$

To ensure accurate online estimation of NPNs, we formulate an update law based on the synaptic weight of the winner neuron $w_{ii\,ll}$ as follows:

Definition 11: Let (N_{NP}, M_{NPo}) be a neural model. The winner neuron's $w_{ii,ll} \in W_{NP}$ synaptic weight can be expressed as

$$w_{ii\,ll} = w_{ii\,ll} + w_{ii\,ll} \quad (ii = 1, 2 \ldots, \mu \text{ and } ll = 1, 2 \ldots, \beta) \tag{17}$$

$$\Delta w_{ii\,ll} = \lambda \left(\frac{x_{ii}^{rr}}{\delta} w_{ii\,ll} \right) \quad (rr = 1, 2 \ldots, \zeta,\ ii = 1, 2 \ldots, \mu \text{ and } ll = 1, 2 \ldots, \beta) \tag{18}$$

where $\lambda \rightarrow$[0,1] represents a learning rate.

Definition 12: Let (N_U, M_{Uo}) and (N_{NP}, M_{NPo}) respectively be an unreliable CROTPN and a neural Petri net. The integration of (N_U, M_{Uo}) and (N_{NP}, M_{NPo}) leads to a neural unreliable CROTPN (N_{NU}, M_{NUo}) with N_{NU} = (P_{NU}, T_{NU}, C_{NU}, I_{NU}, O_{NU}, D_{NU}, K_{NU}, X_{rr}, Y_{rr}, W_{NP}, M_{NUo}), were

1. $P_{NU} = P_U \cup P_{NP}$;
2. $T_{NU} = T_U \cup T_{NP}$;
3. $C_{NU} = C_U$;
4. $I_{NU}(p, t)$: $C_{NU}(p) \times C_{NU}(t) \rightarrow$ **IN**, $O_{NU}(p, t)$: $C_{NU}(p) \times C_{NU}(t) \rightarrow$ **IN**;
5. D_{NU}: $T_{NU} \rightarrow$ **TS**;
6. K_{NU}: $P_{NU} \rightarrow$ **IN**;
7. M_{NUo}: $P_{NU} \rightarrow$ **IN**.

Finally, to show how to detect and treat faults, consider the unreliable net illustrated in Figure 3. Different faults types are recognized by Algorithm 3 according to certain criteria. The neural unreliable net is illustrated in Figure 4. Figure 4 contains three input continuous factors: x_1, x_2, and x_3 and a single output y_1. The output variable is defined as

1. fault type 1 $\rightarrow$ [1 0 0];
2. fault type 2 $\rightarrow$ [0 1 0];
3. fault type 3 $\rightarrow$ [0 0 1].

The treatments include t_{t1}, t_{t2}, or t_{t3}. Note that in the Section V, the behavior of neural networks will be presented in details for diagnosis and treatment of faults.

Algorithm 3 shows the construction of the neural unreliable CROTPN.

Algorithm 3 The Construction of the Neural Unreliable CROTPN

Input: The obtained controlled CROTPN from Algorithm 2, $w_{ii,jj} \rightarrow$[0,1], X_ζ, and ψ (target weight);
Output: A neural unreliable CROTPN and fault type Y_ζ;
Initialization: Design the common recovery place $p_{combined}$;

for all $1 \leq i \leq |P_R|$ ***do***
Design the transitions t_{fi} and t_{ri};
Design the arcs and weights (p_i, t_{fi}), (t_{fi}, $p_{combined}$), ($p_{combined}$, t_{ri}), and (t_{ri}, p_i);
Define a color c_{rni} for t_{fi};
end for
for all $1 \leq i \leq |T_F|$ ***do***
if t_{fi} fires ***then***
while $w_{ii,jj} < \psi$ ***do***
for all $1 \leq rr \leq \zeta$ ***do***
for all $1 \leq ii \leq \mu$ ***do***
for all $1 \leq jj \leq \beta$ ***do***
Calculate the Z_{jj};
Calculate the winner y_{jj}
Update the winner neuron weight $w_{ii,jj}$
end for
end for
end for
end while
end if
end for

IV. DESIGN THE RELIABILITY MODEL OF NEURAL UNRELIABLE CROTPN

The developed neural unreliable net are improved to predict the reliability parameters of an AMS. Note that we integrated the neural unreliable net with the reliability modeling approach presented in [3] to predict the reliability parameters of the system. The failure and recovery transitions t_{fi} and t_{ri} of the neural unreliable CROTPN presented in Definition 12 are

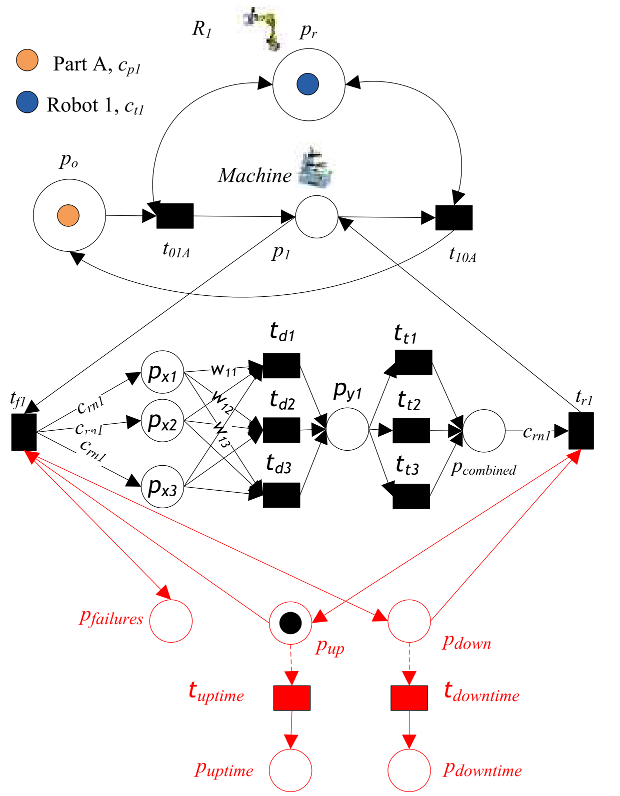

FIGURE 5. Reliability model for the model presented in Figure 4.

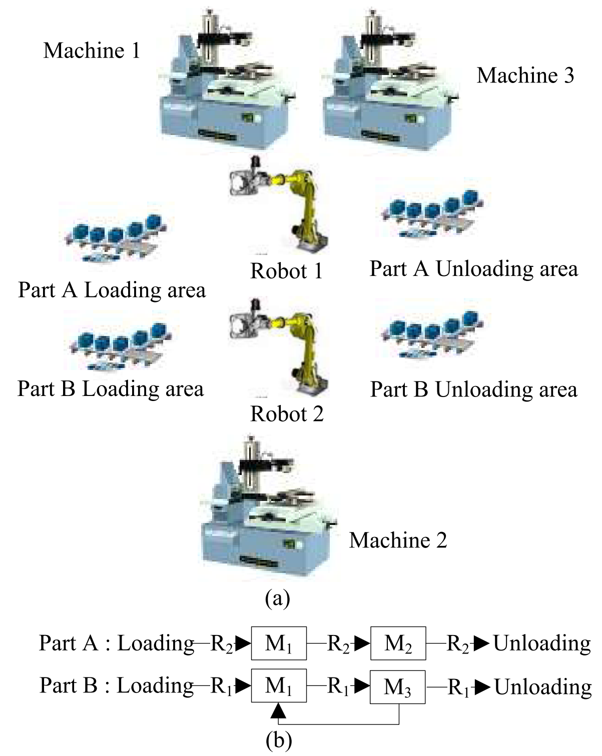

FIGURE 6. (a) Automated manufacturing system in [66], [67] and (b) the operation path.

respectively connected to two places p_{down} and p_{up} $\{(p_{up}, t_{fi}), (t_{fi}, p_{down}), (p_{down}, t_{ri}), (t_{ri}, p_{up})\}$, and initial marking $M_{RNio}(p_{up}) = 1$ and $M_{RNio}(p_{down}) = 0$, which describe the down and up state of the system, as shown in Figure 5. If a failure transition t_{fi} is fired, one token is withdrawn from the on condition place p_{up} and one token is placed in the off condition place p_{down}. For the repair event, the PN behaviour is symmetric.

The system's on time and downtime may therefore easily be calculated by inserting a time counter represented by a test arc, two transitions $t_{downtime}$ and t_{uptime} with the deterministic time delay, and two places $p_{downtime}$ and p_{uptime} with arcs $\{(p_{up}, t_{uptime}), (t_{uptime}, p_{uptime}), (p_{down}, t_{downtime}), (t_{downtime}, p_{downtime})\}$, and initial marking $M_{RNio}(p_{uptime}) = 0$ and $M_{RNio}(p_{downtime}) = 0$, which gather tokens from transitions $t_{downtime}$ and t_{uptime} to represent the time units, as illustrated in Figure 5.

Furthermore, a place $p_{failures}$ is added to the net to estimate the number of occurred failures $N_{failures}$. Each failure transition t_{fi} is connected to a place $p_{failures}(t_{fi}, p_{failures})$ where a token is sent in case of failure, as illustrated in Figure 5. Finally, the reliability parameters can be obtained as follows:

$$T_{MTTF} = T_{uptime}/N_{failures} \quad (19)$$

$$T_{MTTR} = T_{downtime}/N_{failures} \quad (20)$$

$$As = T_{uptime}/(T_{uptime}+T_{downtime}) \quad (21)$$

V. EXPERIMENTAL RESULTS

This section considers an AMS example from the literature [66], [67] to show the experimental results of the developed approach. The system and its operation path are respectively shown in Figure 6 (a) and (b). After implementing Algorithm 1, Figure 7 illustrates the developed CROTPN and we have places p_1, p_2, and p_3 representing respectively machines 1, 2, and 3, one transportation place p_r representing robot 1 (R1) and robot 1 (R1) that used for loading/unloading parts A and B to/from machines, and the operation paths are

1. For part A: a raw part A is assigned to the p_o, the robot 1 loads a part to the machine 1 p_1 by a transition t_{01A}, then the robot 1 unloads the part from machine 1 and loads it to the machine 2 p_2 by a transition by t_{12A}, then the robot 1 unloads the finished part by t_{20A} to the unload station p_0.
2. For part B: a raw part B is assigned to the p_o, the robot 2 loads a part to the machine 1 p_1 by a transition t_{01B}, then the robot 2 unloads the part from machine 1 p_1 and loads it to the inspection machine 3 p_3 by a transition by t_{13B}, if the part has no defects, then the robot 2 places the finished part by t_{30B} to the unload station p_0, otherwise the part returns to the machine 1 p_1 by t_{31B}, then to t_{13B}, p_3, t_{30B}, and p_0.

The initial marking of the CROTPN model is designed as $M_o(p_o, p_r, p_1, p_2, p_3) = M_o(\{c_{p1}, c_{p2}\},\{c_{t1}, c_{t2}\}, 0, 0, 0)$,

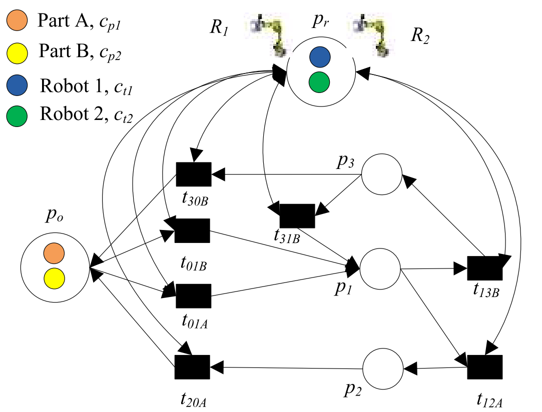

FIGURE 7. Th CROTPN (N, M_O) of an AMS illustrated in Figure 6(a).

where c_{p1}, c_{p2}, c_{t1}, and c_{t2} represent part A, part B, R1 and R2 in the system.

Based on Algorithm 2, the CROTPN shown in Figure 2 has one PPC: $e_1 = \{p_1, t_{13B}, p_3, t_{31B}\}$. Figure 8 illustrates the CROTPN reachability graph. The available spaces in e_1 is computed as stated in Table 1 according to Theorem 2. Table 1 shows that the condition in Theorem 2 has been met. Thus, the deadlock can be avoided and the CROTPN is live.

TABLE 1. The current spaces in production process circuit e_1 of the CROTPN presented in Figure 7.

Marking (M_w)	$K(e_1)$	$\bar{M}(e_1)$	$S'(e_1)$
0	2	0	2
1	2	1	1
2	2	1	1
3	2	0	2
4	2	1	1

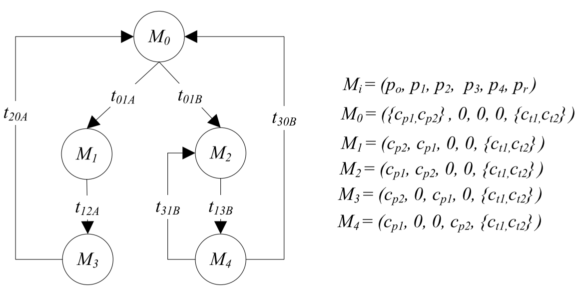

FIGURE 8. The R_L (N, M_o) of the CROTPN presented in Figure 7.

Reconsider the developed CROTPN presented in Figure 7 to show the steps of the Algorithm 3 showing the (N_{NU}, M_{NUo}) construction. First, the unreliable net (N_U, M_{Uo}) is developed as illustrated in Figure 9. In Figure 9, we have

1. three machines, which are p_1, p_2, and p_3, $\mathbf{NP} = \{1, 2, 3\}$;
2. the failure transitions t_{f1}, t_{f2}, and t_{f3};
3. the recovery transitions t_{r1}, t_{r2}, and t_{r3};
4. the failure colors c_{rn1}, c_{rn2}, and c_{rn3}.

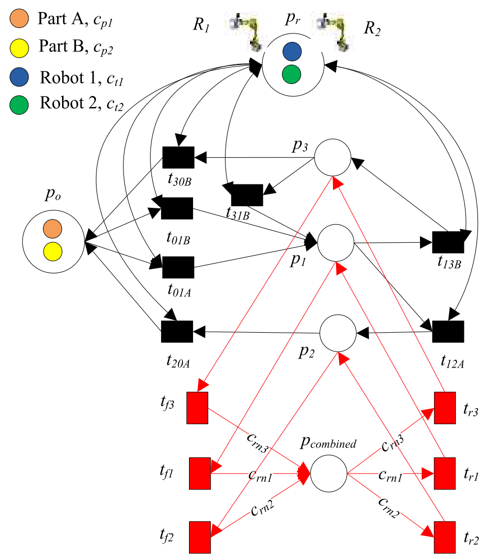

FIGURE 9. The unreliable CROTPN (N_U, M_{Uo}) of an AMS illustrated in Figure 6.

Second, we design the neural model for fault detection and treatment. Two phases are involved in the use of neural networks: training and testing. The relationship between inputs and output of the model is determined during the training phase. Then, by using the test dataset, the neural networks are tested. Finally, the neural networks are capable of diagnosing faults under a variety of operational settings.

Using the unreliable net (N_U, M_{Uo}) presented in Figure 9, the data we need are calculated. The data contains five input continuous factors [46], [67]: x_1(the accelerometer) that measures mechanical vibrations, x_2(the current sensor) that measures variations in the current consumption of the electric motor, x_3(the strain gages), which measures tool torsion, x_4(the coolant sensor), which measures the coolant level, and x_5 (the acoustic emission sensor), which measures acoustic stress wave effects for the diagnosis of a tool break. The single output is defined as: y_1(the tool wearing failure), y_2(the tool breaking failure), y_3(the coolant failure), y_4(the programming errors). The output variable of this model is defined as [1 0 0 0] for the tool wearing failure, [0 1 0 0] for the tool breaking failure, [0 0 1 0] for the coolant failure, and [0 0 0 1] for the programming errors. The collected datasets have 2250 patterns, which include respectively 1575, 450, and 225 training, test, and validation patterns. The GPenSIM tool [1, 68] is applied to construct the developed model presented in Figure 10. In addition, the training, test, and validation datasets respectively provide the feature vector, the fault estimation, and break iteration when the maximum generalization capacity is achieved.

Signals from these sensors are obtained by acquisition systems, identifying that machine tools are abnormal or normal. The machine tool produces random and uniform peaks. Signals showing peaks of the same wavelength mean that tool wear or machining parameters are incorrectly

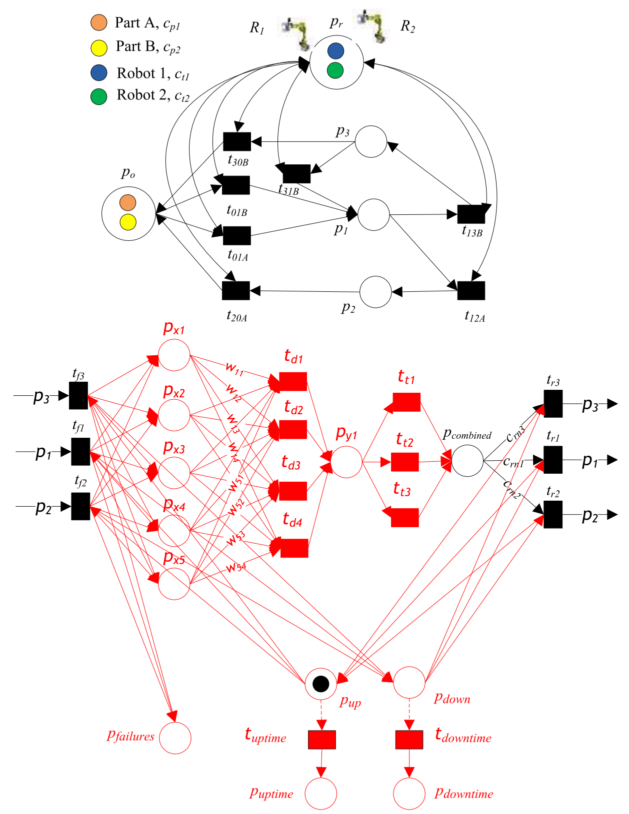

FIGURE 10. The neural unreliable CROTPN (N_{NU}, M_{NUo}) and reliability model of an AMS presented in Figure 9.

TABLE 2. Maintenance data (min) of machines for the system presented in Figure 10.

Machine	MTTF	MTTR t_{t1}	MTTR t_{t2}	MTTR t_{t3}	System state during failure
Machine 1	300	20	30	15	Off
Machine 2	400	22	32	15	Off
Machine 3	450	25	36	15	Off

programmed. Random peaks show that the tool is highly broken [46], [67]. Finally, insufficient coolant causes coolant failure. The suggested treatments contain changing the parameter (t_{t1}) to solve a tool wearing and programming error failures, changing the tool (t_{t2}) to solve a tool break failure, or changing the coolant (t_{t3}) to solve a coolant failure. Many training trials are conducted to determine the best network parameters, which will lead to minimum errors in training. Furthermore, several training tests are carried out using various hidden neurons. Therefore, 12 layers of hidden neural networks are employed. The mean square error of a neural unreliable CROTPN model with a learning rate of 0.00001 at 61 iterations is 0.273 and results in an accurate 95% model.

Finally, verification and validation of Algorithms 3 are performed using the GPenSIM tool [1], [2], [38], [66], [68].

TABLE 3. Comparison of algorithm 3 with current approaches for the system presented in Figure 10.

Performance		Ref. [38]	Ref. [46]	Algorithm 3
Throughput (parts)		75	76	81
Throughput time (min/part)		8.00	7.89	7.40
Utilization	Machine 1 (%)	53.82	55.21	55.22
	Machine 2 (%)	23.63	23.75	24.81
	Machine 3 (%)	22.33	24.55	24.89
	Robot 1 (%)	48.86	48.33	50.57
	Robot 2 (%)	39.15	39.58	43.76

TABLE 4. Reliability parameters for the system presented in Figure 10.

Parameter	Value
$N_{failures}$	4
T_{uptime} (min)	533
$T_{downtime}$ (min)	67
A_s (%)	88.83

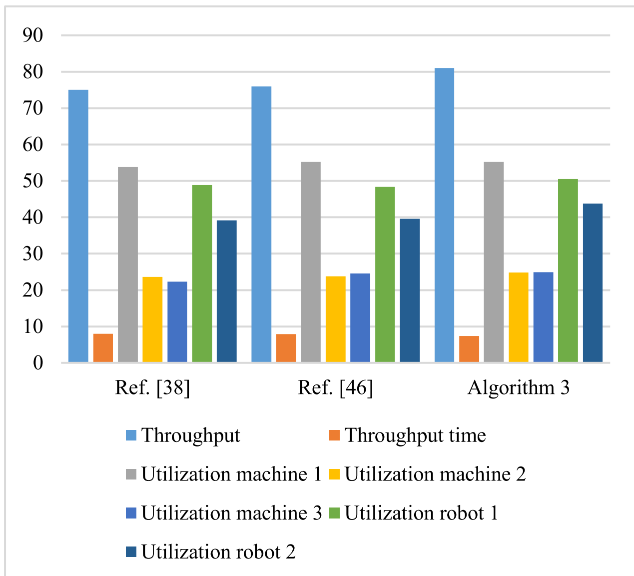

FIGURE 11. Performance of algorithm 3 compared with the current approaches.

R. Davidrajuh [68] developed the GPenSIM tool, which runs in MATLAB software. The GPenSIM application has been created to model, control, simulate, and analyze discrete event systems. GPenSIM integrates Petri net models with other MATLAB toolboxes such as neural networks, control systems and Fuzzy logic. The proposed code of the Algorithms 3 is implemented on MATLAB R2015a. The simulation is performed for 600 min. Maintenance data of machines for the system presented in Figure 10 are presented in Table 2. In addition, Table 3 displays the output of the

GPenSIM tool, including the total throughput (the number of units goes through the simulation time), throughput time (simulation time/total throughput), and utilization of resources ((total occupation time of resource/simulation time)*100). The results of these experiments are compared with those in [38], [46], as shown in Figure 11. The results indicate that Algorithm 3 outperforms the other techniques in terms of throughput, throughput time, and utilization. In addition, the reliability parameters of the system: up time, down time, total number of occurred failures, and availability are presented in Table 4. It observed that the availability of the system is 88.83%.

VI. CONCLUSION

This paper develops a method for deadlock control and reliability modeling based on CROTPN and neural networks to obtain important reliability measures in AMSs, including MTTF, MTTR, and availability. First, an unreliable CROTPN is proposed to obtain "sufficient and necessary conditions" for the CROTPN liveness. Secondly, a fault diagnosis and treatment approach is proposed that integrates the obtained unreliable net with neural networks to ensure that the system is reliable. Furthermore, a neural unreliable CROTPN is used to analyze the system reliability. A simulation is carried out to show the developed approach and the simulation results are compared with the current methods.

The advantages of the developed methodology are (1) the neural unreliable net has been proved to be simpler in structure and efficient of overcoming the deadlock issue and modeling AMS reliability as compared with the studies in [38], [46], [67], (2) it can integrate the neural networks and reliability model with CROTPN, (3) an integrated approach is utilized in an AMS to insure that no deadlock happens, faults are diagnosed and treated, and the reliability of the system is evaluated.

The limitation of this study includes that the model is proposed using discrete types of data. Thus, future research will focus on developing the proposed approach for designing CROTPN using continuous types of data.

ACKNOWLEDGMENT

This Project was funded by the National Plan for Science, Technology and Innovation (MAARIFAH), King Abdulaziz City for Science and Technology, Kingdom of Saudi Arabia, Award Number (14-ELE69-02).

REFERENCES

[1] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, "Single controller-based colored Petri nets for deadlock control in automated manufacturing systems," *Processes*, vol. 8, no. 1, p. 21, Dec. 2019.

[2] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, "Intelligent colored token Petri nets for modeling, control, and validation of dynamic changes in reconfigurable manufacturing systems," *Processes*, vol. 8, no. 3, p. 358, Mar. 2020.

[3] M. Bertolini, M. Bevilacqua, and G. Mason, "Reliability design of industrial plants using Petri nets," *J. Qual. Maintenance Eng.*, vol. 12, no. 4, pp. 397–411, Oct. 2006.

[4] W. P. Wang, S. Y. Bao, and Z. L. Gao, "The development of reliability modeling and analysis tool based on stochastic Petri nets," in *Proc. Adv. Mater. Res.*, 2010, pp. 566–570.

[5] S. Y. Bao, W. P. Wang, J. X. Zhu, and Z. L. Gao, "SPN based reliability analysis in the process industry," in *Proc. Adv. Mater. Res.*, 2010, pp. 561–565.

[6] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 374–393, Apr. 2011.

[7] H. Kaid, A. Al-Ahmari, A. M. El-Tamimi, E. A. Nasr, and Z. Li, "Design and implementation of deadlock control for automated manufacturing systems," *South Afr. J. Ind. Eng.*, vol. 30, no. 1, pp. 1–23, 2019.

[8] D. Y. Chao, "Improvement of suboptimal siphon-and FBM-based control model of a well-known S^3PR," *IEEE Trans. Autom. Sci. Eng.*, vol. 8, no. 2, pp. 404–411, Nov. 2011.

[9] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 34, no. 1, pp. 38–51, Jan. 2004.

[10] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *Int. J. Adv. Manuf. Technol.*, vol. 19, no. 3, pp. 192–208, Feb. 2002.

[11] M. Uzam and M. Zhou, "Iterative synthesis of Petri net based deadlock prevention policy for flexible manufacturing systems," in *Proc. IEEE Int. Conf. Syst., Man Cybern.*, Oct. 2004, pp. 4260–4265.

[12] Y.-L. Pan, C.-Y. Tseng, and T.-C. Row, "Design of improved optimal and suboptimal deadlock prevention for flexible manufacturing systems based on place invariant and reachability graph analysis methods," *J. Algorithms Comput. Technol.*, vol. 11, no. 3, pp. 261–270, Sep. 2017.

[13] M. Zhao and M. Uzam, "A suboptimal deadlock control policy for designing non-blocking supervisors in flexible manufacturing systems," *Inf. Sci.*, vols. 388–389, pp. 135–153, May 2017.

[14] A. M. El-Tamimi, E. A. Nasr, A. Al-Ahmari, H. Kaid, and Z. Li, "Evaluation of deadlock control designs in automated manufacturing systems," in *Proc. Int. Conf. Ind. Eng. Oper. Manage. (IEOM)*, New York, NY, USA, Mar. 2015, pp. 1–10.

[15] S. Wang, D. You, and M. Zhou, "A necessary and sufficient condition for a resource subset to generate a strict minimal siphon in S^4PR," *IEEE Trans. Autom. Control*, vol. 62, no. 8, pp. 4173–4179, Aug. 2017.

[16] M. A. Lawley and W. Sulistyono, "Robust supervisory control policies for manufacturing systems with unreliable resources," *IEEE Trans. Robot. Autom.*, vol. 18, no. 3, pp. 346–359, Jun. 2002.

[17] F.-S. Hsieh, "Robustness analysis of Petri nets for assembly/disassembly processes with unreliable resources," *Automatica*, vol. 42, no. 7, pp. 1159–1166, 2006.

[18] S. Y. Wang, S. F. Chew, and M. A. Lawley, "Using shared-resource capacity for robust control of failure-prone manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst. Humans*, vol. 38, no. 3, pp. 605–627, Apr. 2008.

[19] S. F. Chew, S. Wang, and M. A. Lawley, "Robust supervisory control for product routings with multiple unreliable resources," *IEEE Trans. Autom. Sci. Eng.*, vol. 6, no. 1, pp. 195–200, Jan. 2009.

[20] G. Y. Liu, Z. W. Li, K. Barkaoui, and A. M. Al-Ahmari, "Robustness of deadlock control for a class of Petri nets with unreliable resources," *Inf. Sci.*, vol. 235, pp. 259–279, Jun. 2013.

[21] H. Yue, K. Xing, and Z. Hu, "Robust supervisory control policy for avoiding deadlock in automated manufacturing systems with unreliable resources," *Int. J. Prod. Res.*, vol. 52, no. 6, pp. 1573–1591, 2014.

[22] H. Yue, K. Y. Xing, H. S. Hu, W. M. Wu, and H. Y. Su, "Robust supervision using shared-buffers in automated manufacturing systems with unreliable resources," *Comput. Ind. Eng.*, vol. 83, pp. 139–150, May 2015.

[23] F. Wang, K.-Y. Xing, M.-C. Zhou, X.-P. Xu, and L.-B. Han, "A robust deadlock prevention control for automated manufacturing systems with unreliable resources," *Inf. Sci.*, vol. 345, pp. 243–256, Jun. 2016.

[24] Y. Feng, K. Xing, Z. Gao, and Y. Wu, "Transition cover-based robust Petri net controllers for automated manufacturing systems with a type of unreliable resources," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 47, no. 11, pp. 3019–3029, Nov. 2017.

[25] G. Y. Liu, P. Li, Z. W. Li, and N. Q. Wu, "Robust deadlock control for automated manufacturing systems with unreliable resources based on Petri net reachability graphs," *IEEE Trans. Syst., Man Cybern., Syst.*, vol. 49, no. 7, pp. 1371–1385, Jul. 2018.

[26] N. Ran, J. Hao, S. Wang, Z. Dong, Z. He, Z. Liu, and Y. Ruan, ''K-codiagnosability verification of labeled Petri nets,'' *IEEE Access*, vol. 7, pp. 185055–185062, 2019.

[27] E. A. Alzalab, Z. Yu, N. Wu, and H. Kaid, ''Fault-recovery and repair modeling of discrete event systems using Petri nets,'' *IEEE Access*, vol. 8, pp. 170237–170247, 2020.

[28] A. Ghaffari, N. Rezg, and X. Xie, ''Design of a live and maximally permissive Petri net controller using the theory of regions,'' *IEEE Trans. Robot. Autom.*, vol. 19, no. 1, pp. 137–141, Feb. 2003.

[29] M. Uzam, ''The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems,'' *Int. J. Adv. Manuf. Technol.*, vol. 23, nos. 3–4, pp. 204–219, Feb. 2004.

[30] D. J. Sun, Y. F. Chen, M. A. El-Meligy, M. A. F. Sharaf, N. Q. Wu, and Z. W. Li, ''On algebraic identification of critical states for deadlock control in automated manufacturing systems modeled with Petri nets,'' *IEEE Access*, vol. 7, pp. 121332–121349, 2019.

[31] H. Kaid, A. Al-Ahmari, Z. Li, and R. Davidrajuh, ''Automatic supervisory controller for deadlock control in reconfigurable manufacturing systems with dynamic changes,'' *Appl. Sci.*, vol. 10, no. 15, p. 5270, Jul. 2020.

[32] Y. Chen and Z. Li, ''On structural minimality of optimal supervisors for flexible manufacturing systems,'' *Automatica*, vol. 48, no. 10, pp. 2647–2656, Oct. 2012.

[33] E. A. Nasr, A. M. El-Tamimi, A. Al-Ahmari, and H. Kaid, ''Comparison and evaluation of deadlock prevention methods for different size automated manufacturing systems,'' *Math. Problems Eng.*, vol. 2015, pp. 1–19, Sep. 2015.

[34] G. Y. Liu, L. C. Zhang, L. Chang, A. Al-Ahmari, and N. Q. Wu, ''Robust deadlock control for automated manufacturing systems based on elementary siphon theory,'' *Inf. Sci.*, vol. 510, pp. 165–182, Feb. 2020.

[35] X. Y. Li, G. Y. Liu, Z. W. Li, N. Q. Wu, and K. Barkaoui, ''Elementary siphon-based robust control for automated manufacturing systems with multiple unreliable resources,'' *IEEE Access*, vol. 7, no. 1, pp. 21006–21019, 2019.

[36] Z. Li and M. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London, U.K.: Springer, 2009.

[37] H. Kaid, A. Al-Ahmari, and Z. Li, ''Supervisor controller-based colored Petri nets for deadlock control and machine failures in automated manufacturing systems,'' *Int. J. Ind. Manuf. Eng.*, vol. 14, no. 10, pp. 426–432, 2020.

[38] A. Al-Ahmari, H. Kaid, Z. Li, and R. Davidrajuh, ''Strict minimal siphon-based colored Petri net supervisor synthesis for automated manufacturing systems with unreliable resources,'' *IEEE Access*, vol. 8, pp. 22411–22424, 2020.

[39] Y. Maki and K. A. Loparo, ''A neural-network approach to fault detection and diagnosis in industrial processes,'' *IEEE Trans. Control Syst. Technol.*, vol. 5, no. 6, pp. 529–541, Nov. 1997.

[40] X.-Q. Liu, H.-Y. Zhang, J. Liu, and J. Yang, ''Fault detection and diagnosis of permanent-magnet DC motor based on parameter estimation and neural network,'' *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 1021–1030, Oct. 2000.

[41] L. A. M. Riascos and P. E. Miyagi, ''Supervisor system for detection and treatment of failures in manufacturing systems using distributed Petri nets,'' in *Proc. Manuf., Modeling, Manage. Control IFAC Workshop (MIM)*, Prague, Czech Republic, Aug. 2001, p. 83.

[42] L. A. M. Riascos, F. G. Cozman, and P. E. Miyagi, ''Detection and treatment of faults in automated machines based on Petri nets and Bayesian networks,'' in *Proc. IEEE Int. Symp. Ind. Electron.*, Jun. 2003, pp. 729–734.

[43] E. P. Miyagi and L. A. M. Riascos, ''Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets,'' *Control Eng. Pract.*, vol. 14, no. 4, pp. 397–408, Apr. 2006.

[44] S. Rajakarunakaran, P. Venkumar, D. Devaraj, and K. S. P. Rao, ''Artificial neural network approach for fault detection in rotary system,'' *Appl. Soft Comput.*, vol. 8, no. 1, pp. 740–748, Jan. 2008.

[45] H. Honggui, L. Ying, and Q. Junfei, ''A fuzzy neural network approach for online fault detection in waste water treatment process,'' *Comput. Electr. Eng.*, vol. 40, no. 7, pp. 2216–2226, Oct. 2014.

[46] H. Kaid, A. Al-Ahmari, E. A. Nasr, A. Al-Shayea, A. K. Kamrani, M. A. Noman, and H. A. Mahmoud, ''Petri net model based on neural network for deadlock control and fault detection and treatment in automated manufacturing systems,'' *IEEE Access*, vol. 8, pp. 103219–103235, 2020.

[47] N. Wu and M. Zhou, ''Shortest routing of bidirectional automated guided vehicles avoiding deadlock and blocking,'' *IEEE/ASME Trans. Mechatronics*, vol. 12, no. 1, pp. 63–72, Feb. 2007.

[48] N. Wu and M. Zhou, ''Modeling and deadlock avoidance of automated manufacturing systems with multiple automated guided vehicles,'' *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 35, no. 6, pp. 1193–1202, Dec. 2005.

[49] N. Wu and M. Zhou, ''Modeling and deadlock control of automated guided vehicle systems,'' *IEEE/ASME Trans. Mechatronics*, vol. 9, no. 1, pp. 50–57, Mar. 2004.

[50] H. Chen, N. Wu, and M. Zhou, ''A novel method for deadlock prevention of AMS by using resource-oriented Petri nets,'' *Inf. Sci.*, vol. 363, pp. 178–189, Oct. 2016.

[51] N. Wu, M. Zhou, and Z. Li, ''Resource-oriented Petri net for deadlock avoidance in flexible assembly systems,'' *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 56–69, Jan. 2008.

[52] N. Wu and M. Zhou, *System Modeling and Control With Resource-Oriented Petri Nets*. New York, NY, USA: CRC Press, 2009.

[53] N. Wu and M. Zhou, ''Avoiding deadlock and reducing starvation and blocking in automated manufacturing systems,'' *IEEE Trans. Robot. Autom.*, vol. 17, no. 5, pp. 658–669, Oct. 2001.

[54] N. Wu, ''Avoiding deadlocks in automated manufacturing systems with shared material handling system,'' in *Proc. Int. Conf. Robot. Autom.*, 1997, pp. 2427–2432.

[55] N. Wu and M. Zhou, ''Deadlock avoidance in semiconductor track systems,'' in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2002, pp. 193–198.

[56] Z. Xiang, ''Deadlock avoidance of flexible manufacturing systems by colored resource-oriented Petri nets with novel colored capacity,'' in *Proc. EasyChair*, 2020, pp. 2516–2314.

[57] H. F. Chen, N. Q. Wu, Z. W. Li, and T. Qu, ''On a maximally permissive deadlock prevention policy for automated manufacturing systems by using resource-oriented Petri nets,'' *ISA Trans.*, vol. 89, pp. 67–76, Jun. 2019.

[58] N. Wu and M. Zhou, ''Process vs resource-oriented Petri net modeling of automated manufacturing systems,'' *Asian J. Control*, vol. 12, no. 3, pp. 267–280, Apr. 2010.

[59] N. Wu, M. Zhou, and G. Hu, ''On Petri net modeling of automated manufacturing systems,'' in *Proc. IEEE Int. Conf. Netw., Sens. Control*, Apr. 2007, pp. 228–233.

[60] H. Chen, N. Wu, and M. Zhou, ''Resource-oriented Petri net-based approach to deadlock prevention of AMSs,'' in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 515–520.

[61] N. Wu and M. Zhou, ''Resource-oriented Petri nets in deadlock avoidance of AGV systems,'' in *Proc. ICRA. IEEE Int. Conf. Robot. Autom.*, May 2001, pp. 64–69.

[62] H. Chen, N. Wu, Z. Li, T. Qu, and H. Xiao, ''Liveness of disjunctive and strict single-type automated manufacturing system: An ROPN approach,'' *IEEE Access*, vol. 7, pp. 17760–17771, 2019.

[63] N. Wu and M. Zhou, ''Deadlock resolution in automated manufacturing systems with robots,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 4, no. 3, pp. 474–480, Jul. 2007.

[64] N. Wu, M. Zhou, and G. Hu, ''One-step look-ahead maximally permissive deadlock control of AMS by using Petri nets,'' *Acm Trans. Embedded Comput. Syst.*, vol. 12, no. 1, pp. 1–23, Jan. 2013.

[65] M. N. Q. Wu and M. C. Zhou, ''Intelligent token Petri nets for modelling and control of reconfigurable automated manufacturing systems with dynamical changes,'' *Trans. Inst. Measur. Control*, vol. 33, no. 1, pp. 9–29, May 2011.

[66] H. Kaid, A. Al-Ahmari, and Z. Li, ''Colored resource-oriented Petri net based ladder diagrams for PLC implementation in reconfigurable manufacturing systems,'' *IEEE Access*, vol. 8, pp. 217573–217591, 2020.

[67] H. Kaid, A. Al-Ahmari, Z. Li, and W. Ameen, ''Deadlock control and fault detection and treatment in reconfigurable manufacturing systems using colored resource-oriented Petri nets based on neural network,'' *IEEE Access*, vol. 9, pp. 84932–84947, 2021.

[68] R. Davidrajuh, *Modeling Discrete-Event Systems With GPenSIM: An Introduction*. Cham, Switzerland: Springer, 2018.