International Journal of Mechanical, Industrial and Aerospace Sciences
ISSN: 2517-9950
Vol:14, No:10, 2020

Supervisor Controller-Based Colored Petri Nets for Deadlock Control and Machine Failures in Automated Manufacturing Systems

Husam Kaid, Abdulrahman Al-Ahmari, Zhiwu Li

Abstract—This paper develops a robust deadlock control technique for shared and unreliable resources in automated manufacturing systems (AMSs) based on structural analysis and colored Petri nets, which consists of three steps. The first step involves using strict minimal siphon control to create a live (deadlock-free) system that does not consider resource failure. The second step uses an approach based on colored Petri net, in which all monitors designed in the first step are merged into a single monitor. The third step addresses the deadlock control problems caused by resource failures. For all resource failures in the Petri net model a common recovery subnet based on colored petri net is proposed. The common recovery subnet is added to the obtained system at the second step to make the system reliable. The proposed approach is evaluated using an AMS from the literature. The results show that the proposed approach can be applied to an unreliable complex Petri net model, has a simpler structure and less computational complexity, and can obtain one common recovery subnet to model all resource failures.

Keywords**—**Automated manufacturing system, colored Petri net, deadlock, siphon.

I. Introduction

An automated manufacturing system (AMS) is a collection of buffers, machines, robots, fixtures, and automated tools. There are different products types entering AMS at separate points in time; the system has the ability to handle these products according to the specific sequence of processes and resources sharing. The resource sharing causes deadlocks, in which the local or global system is disabled. Therefore, in order to prevent deadlock in AMS, an effective deadlock control algorithm is needed.

Petri nets are graphical and mathematical tools that are convenient for modeling, analysis, and control of deadlocks in AMSs [1]. It is utilized to represent the characteristics and behaviors of AMS, such as confliction, synchronization, and sequencing. Moreover, Petri nets can be applied to provide the behavioral characteristics such as boundedness and liveness [2]. In order to solve the deadlock issue in AMSs, several approaches based on Petri nets are proposed in the literature. These methods have been classified into three strategies: deadlock detection and recovery, deadlock avoidance, and deadlock prevention [2]. In addition, three criteria were proposed for evaluating and designing a supervisor for AMS control, namely, computational complexity, structural complexity, and behavioral permissiveness [2]. Therefore, deadlock prevention policies are the objectives of many researchers and can provide liveness-enforcing supervisors with the mentioned criteria [2]. The deadlock control techniques available in the literature were developed for AMSs with reliable and unreliable resources.

For an AMS with reliable resources, there are two techniques used to prevent deadlock involving the use of Petri nets: reachability graph analysis [3]-[5] and structural analysis [1], [6]. The reachability graph analysis requires listing all or part of the reachable markings; hence, it suffers from a state explosion problem. The reachability graph can be classified into two parts: the live zone (LZ) and the deadlock zone (DZ). First-met bad markings (FBMs) are defined and extracted from the DZ. In this case, deadlocks are eliminated by designing and adding monitors to prevent FBMs from being reached. This process requires iterations to identify all FBMs [7].

Various approaches have been introduced to prohibit deadlock situations, which are the siphon control and theory of region based approaches [6]-[13]. Several deadlock control methods and fault detection have been developed for different classes of Petri nets under unreliable resources [14]-[20].

From the literature, we observed that deadlock control strategies based on siphons may be applied by adding the monitors and the related arcs to the initial net, such that its siphons are permanently marked. The major drawback of the existing policies is that several monitors and related arcs are placed into the initial Petri net model, leading to increase the Petri net model supervisor's complexity compared with the initial Petri net model. In addition, for unreliable resources, recovery subnets are proposed to model resource failures and recoveries and applied for each unreliable resource, resulting in high structural complexity of the initial model. However, there is a need to propose a supervisor to address all the unreliable resources in a system. This supervisor does not require introducing inhibitor arcs or enumerating of reachability graphs and leads to low computational overheads. Therefore, an integrated strategy is required to minimize the structural complexity of the Petri net supervisors for AMS. The purpose of this paper is to develop an integrated deadlock control strategy. The first step is to use the strict minimal

H. K. and A.A. A are with the Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia (e-mail: yemenhussam@yahoo.com, alahmari@ksu.edu.sa).

Z. L. is with the Institute of Systems Engineering, Macau University of Science and Technology, Taipa, Macau 999078, China. In addition, he is with the School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, P. R. China (e-mail: systemscontrol@gmail.com).

siphon control to derive a controlled system that does not consider the failure of resources [21]. The second step is a colored Petri net based on approach, in which all monitors designed in the first step are merged into a single monitor. In the third step, global recovery subnet is designed to model all resource failures based on colored Petri nets, and recovery is applied to the obtained system by the second step to ensure the reliability of the system.

The rest of the paper is constructed as follows. Petri nets basics, deadlock prevention approach based on the SMS and the proposed robust control based on colored Petri nets are introduced in Section II. Section III shows an example from the literature. Finally, Section IV presents conclusions and future research.

II. Methodology

A. Petri Nets Basics

Let N be a Petri net with $N = (P, T, F, W)$, where P is finite non-empty set of places and T is finite non-empty set of and transitions. Elements in $P \cup T$ are graphically named by nodes. Where $P \cup T \neq \emptyset$ and $P \cap T = \emptyset$; places are depicted by circles and transitions are depicted by bars. $F \subseteq (P \times T) \cup (T \times P)$ is said to be a set of directed arcs of N that join the places to transitions or transitions to places. $W: (P \times T) \cup (T \times P) \rightarrow \boldsymbol{IN}$ is a mapping that assigns an arc's weight, where $\boldsymbol{IN} = \{0, 1, 2, ...\}$ [1].

$N = (P, T, F, W)$ is called an ordinary net if $\forall(p, t) \in F$, $W(p, t) = 1$. Let $x, z \in P \cup T$ be nodes in $N = (P, T, F, W)$, N is said to be a weighted net if there is an arc between x and z such that $W(x, z) > 1$. Assume that a and b are nodes in $N = (P, T, F, W)$, i.e., $a, b \in P \cup T$. Then, ${}^{\bullet}a = \{b \in P \cup T \mid (b, a) \in F\}$ is called the input (preset) of node a, and $a^{\bullet} = \{b \in P \cup T \mid (a, b) \in F\}$ is called the output (postset) of node a [1].

A marking M of N is a mapping $M: P \rightarrow \boldsymbol{IN}$ and represents the status of the system. (N, M_o) is a marked Petri net, expressed as $N = (P, T, F, W, M_o)$, where M_o is the initial marking of N, $M_o: P \rightarrow \boldsymbol{IN}$. Suppose that $M(p)$ is the number of tokens in place p, a transition $t \in T$ is enabled at any marking M if for all $p \in {}^{\bullet}t$, $M(p) \geq W(p, t)$, which is expressed as $M[t\rangle$. When the enable transition t fires, it takes $W(p, t)$ token(s) from each place $p \in {}^{\bullet}t$, and deposits $W(t, p)$ token(s) in each place $p \in t^{\bullet}$. Thus, it leads the system to reach a new marking M', expressed as $M[t\rangle M'$ in short, the new reachable marking can be computed as $M'(p) = M(p) - W(p, t) + W(t, p)$.

Let (N, M_o) be a marked Petri net with $N = (P, T, F, W, M_o)$. N has a self-loop if for all $p, t \in P_C \cup T$; $W(p, t) > 0$ implies $W(t, p) > 0$. Let (N, M_o) be an ICPN with $N = (P, T, F, W, M_o)$. $[N]$ is called the incidence matrix of net N, where $[N]$ is a $|P|\times|T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$. The incidence vector of a place p is represented by $[N](p, .)$ and the incidence vector of a transition t is represented by $[N](., t)$.

$R(N, M)$ is a set of markings that are reachable from M in a Petri net model (N, M). Let (N, M_o) be a marked Petri net with $N = (P, T, F, W, M_o)$. A transition $t \in T$ is live (deadlock free) if for all $M \in R(N, M)$, there exists $M' \in R(N, M)$ such that firing sequence $M'[t\rangle$ holds. A transition is dead at M_o if there does not exist $t \in T$ such that $M_o[t\rangle$ holds. A marking M' is called reachable from M if there exists a sequence of transitions $\delta = t_1\ t_2\ t_3\ ...\ t_n$ that can be fired, and markings M_1, M_2, M_3, ..., and M_{n-1} are such that $M[t_o\rangle M_1[t_1\rangle M_2[t_2\rangle M_3 ... M_n [t_n\rangle M'$ holds, denoted as $M[\delta\rangle M'$, fulfills the state equation $M' = M + [N]\,\vec{\delta}$ [1].

Let (N, M_o) be a marked Petri net with $N = (P, T, F, W, M_o)$. N is called bounded if there exists $q \in \mathbf{IN}$, $\forall M \in R(N, M_0)$, $\forall p \in P_C$, $M(p) \leq q$. (N, M_0) is structurally bounded if it is bounded for any M_o. N is called safe if $\forall M \in R(N, M_0)$, $\forall p \in P_C$, $M(p) \leq 1$. (N, M_0) is q -safe if it is q-bounded [1].

A place vector I: $P \rightarrow \mathbf{Z}$ indexed by P is said to be a place invariant (P-invariant) if $I^T. [N] = \mathbf{0}^T$ and $I \neq \mathbf{0}$, a transition vector J: $T \rightarrow \mathbf{Z}$ indexed by T is said to be a transition invariant (T-invariant) if $[N]. J = \mathbf{0}$ and $J \neq \mathbf{0}$, where $\mathbf{Z} = \{\ldots, -2, -1, 0, 1, 2, ...\}$. A place invariant I is called a place semiflow or P-semiflow, if each element of I is nonnegative. Assume I is a place invariant of $N = (P, T, F, W, M_o)$ and M is a marking reachable from the initial marking M_o. Then, $I^T M = I^T M_o$. $||I|| = \{p \mid I(p) \neq 0\}$ is called support of place invariant I. $||J|| = \{t \mid J(t) \neq 0\}$ is called support of transition invariant J. The $||I||$ are divided into three parts: (1) $||I||^+ = \{p|I(p) > 0\}$, which called positive support of place invariant I. (2) $||I||^- = \{p \mid I(p) < 0\}$, which called negative support of P-invariant I. (3) I is a minimal place invariant if $||I||$ is not a superset of the support of any other one and its components are mutually prime [1]. The $||J||$ are divided into three parts: (1) $||J||^+ = \{t|J(t) > 0\}$, which called positive support of transition invariant J. (2) $||J||^- = \{t \mid J(t) < 0\}$, which called negative support of transition invariant J. (3) J is a minimal transition invariant if $||J||$ is not a superset of the support of any other one and its components are mutually prime [1].

Let (N, M_o) be a colored finite-capacity PN with $N = (P_C, T, F, W, M_o, K, SC, C_f, N_f, A_f, G_f, I_f)$ [1], where P, T, and F are already defined in previous Definitions. SC is a set of colors involving c colors and the color operations. C_f: $P_C \rightarrow c$ is the function of the color, which maps $p \in P_C$ into colors $c \in SC$. N_f is the function of the node, which maps F into $(P_C \times T) \cup (T \times P_C)$. A_f is the function of the arc, which maps each flow (arc) $f \in F$ into the term e. G_f is the function of the guard, which maps each transition $t \in T$ to a guard expression g that has a Boolean value. I_f is the function of the initialization, which maps each place $p \in P_C$ into an initialization expression that has initial states of places.

B. Strict Minimal Siphons

Definition1 [22]. A simple sequential process (S^2P) is a Petri net model with $N = (\{p^0\} \cup P_A, T, F)$, satisfying: (1) N is a strongly connected state machine and (2) each circuit N contains place p^0, where p^0 is a process idle place and P_A is a set of operation places, $P_A \neq \emptyset$.

Definition2 [22]. A simple sequential process with resources (S^2PR) is a Petri net model with $N = (\{p^0\} \cup P_A \cup P_R, T, F)$, satisfying:

1. The subnet created by $Y = P_A \cup \{p^0\} \cup T$ is an S^2P.
2. $P_R \neq \emptyset$ and $(P_A \cup \{p^0\}) \cap P_R = \emptyset$, where P_R is called a set of resource places.

3. $\forall p \in P_A, \forall t \in {}^\bullet p, \forall t' \in p^\bullet, \exists r_p \in P_R, {}^\bullet t \cap P_R = t'^\bullet \cap P_R = \{r_p\}$.
4. $\forall r \in P_R, {}^{\bullet\bullet}r \cap P_A = r^{\bullet\bullet} \cap P_A \neq \emptyset$ and ${}^\bullet r \cap r^\bullet \neq \emptyset$.
5. ${}^{\bullet\bullet}(p^0) \cap P_R = (p^0)^{\bullet\bullet} \cap P_R \neq \emptyset$.

Definition3 [22]. A simple sequential process with resources (S^2PR) with $N = (\{p^0\} \cup P_A \cup P_R, T, F)$, and M_o is called an initial marking of net N. An S^2PR is called acceptably marked, satisfying: (1) $M_o(p^0) \geq 1$, (2) $M_o(p) = 0$, $\forall p \in P_A$, and (3) $M_o(r) \geq 1, \forall r \in P_R$.

Definition4 [22]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ be an S^2PR net and called S^3PR for abbreviation, is repetitively expressed as follows:

1. An S^2PR is as well an S^3PR
2. Let i=1,2 and N_i be two S^3PRs, where $N_i = (\{p^0{}_i\} \cup PA_i \cup PR_i, T_i, F_i)$, such that $(\{p^0{}_1\} \cap \{p^0{}_2\}) = \emptyset$, $(PA_1 \cap PA_2) = \emptyset$, $PR_1 \cap PR_2 = P_C$ and $T_1 \cap T_2 \neq \emptyset$. Thus, the net S^3PR with $N = (\{p^0\} \cup P_A \cup P_R, T, F)$ resulting from the composition of N_1 and N_2 by the set of common P_C and denoted as: (1) $p^0 = \{p^0{}_1\} \cup \{p^0{}_2\}$, $P_R = PR_1 \cup PR_2$, $P_A = PA_1 \cup PA_2$, $T = T_1 \cup T_2$, $F = F_1 \cup F_2$.

Definition5 [22]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net. N is called acceptably marked if the following conditions are satisfied: (1) M_o is an initial marking of N; (2) N is an acceptably marked S^2PR. (3) For all $i \in \{1,2\}$, (N_i, M_{oi}) is called an acceptably marked S^3PR. Thus, the net S^3PR with $N = N_1 \circ N_2$ is acceptably marked, and

1. For all $i \in \{1,2\}$, for all $r \in P_{Ri} \setminus P_C$, $M_o(r) = M_{oi}(r)$.
2. For all $i \in \{1,2\}$, for all $r \in P_C$, $M_o(r) = \max\{M_{o1}(r), M_{o2}(r)\}$.
3. For all $i \in \{1,2\}$, for all $p \in P_{Ai} \cup \{p^0{}_i\}$, $M_o(p) = M_{oi}(p)$.

Definition6 [22]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net. A non-empty set $S \subseteq P$ is named a siphon in N if ${}^\bullet S \subseteq S^\bullet$. If a siphon contains no other siphons, it is considered a minimal siphon.

Definition7 [22]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net. S is named a minimal siphon in N. A minimal siphon S is named a strict minimal siphon if $S^\bullet \subsetneq {}^\bullet S$. Let $\Pi = \{S_1, S_2, S_3, \ldots, S_k\}$ be a set of strict minimal siphons of N. We have $S = S_A \cup S_R$, $S_R = S \cap P_R$, and $S_A = S \setminus S_R$, where S_A and S_R are sets of operations and resources places, respectively.

Definition8 [22]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net, $r \in P_R$ be a reliable resource place in N. The operation places that use r are recognized as the set of holders of r, expressed as $H(r) = \{p | p \in P_A, p \in {}^{\bullet\bullet}r \cap P_A \neq \emptyset\}$. $[S]$ is called the complementary set or stealing places of S if $[S] = (\cup_{r \in SR} H(r)) \setminus S_A$, where stealing places are operation places that require resources places of siphon S, but are not in a siphon S.

C. Deadlock Prevention Policy Based on Strict Minimal Siphons

This section introduces a deadlock-prevention approach based on SMSs to build a controlled Petri net model. This approach is derived from Ezpeleta et al. [21].

Definition9 [22]. Let (N_1,M_1) and (N_2,M_2) be two Petri nets with $N_i = (P_i, T_i, F_i, W_i)$, where i = 1, 2. (N, M) with $N = (P, T, F, W)$ is named a synchronous net resulting from the integration of (N_1,M_1) and (N_2,M_2), defined as $(N_1,M_1) \parallel (N_2,M_2)$, satisfying: (1) $P = P_1 \cup P_2$ and $P_1 \cap P_2 = \emptyset$. (2) $T = T_1 \cup T_2$, (3) $F = F_1 \cup F_2$, (4) $W(e) = W_i(e)$, where $e \in F_i$, i = 1, 2, and (5) $M(p) = M_i(p)$, $p \in P_i$, i= 1, 2.

Definition10 [22]: Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net. The deadlock controller for (N, M_o) proposed in [21] is defined as $(V, M_{Vo}) = (P_V, T_V, F_V, M_{Vo})$, where (1) $P_V = \{V_S | S \in \Pi\}$ is set of control places. (2) $T_V = \{t | t \in {}^\bullet V_S \cup V_S^\bullet\}$. (3) $F_V \subseteq (P_V \times T_V) \cup (T_V \times P_V)$ is named a flow relation of N, denoted by an arc with an arrow from control places to transitions or transitions to control places. (4) For all $V_S \in P_V$, $M_{Vo}(V_S) = M_{Vo}(S) - 1$, where $M_{Vo}(V_S)$ is an initial marking of a control place.

(N_V, M_{Vo}) is named a controlled Petri net model resulting from the composition of (V, M_{Vo}) and (N, M_o), defined as $(V, M_{Vo}) \parallel (N, M_o)$. Based on the strict minimal siphon concept, the proposed deadlock prevention algorithm developed by [21] is shown as follows:

Algorithm 1: *(Deadlock-prevention algorithm based on SMSs)*
Input: *An S^3PR (N, M_o).*
Output: *A net (N_V, M_{Vo}).*
Step 1: *Calculate the Π for N.*
Step 2: for *each $S \in \Pi$* ***do***
1. *Design the V_S;*
2. *Insert the Vs output arcs;*
3. *Insert the Vs input arcs;*
4. *Define $M_{Vo}(V_S)$.*

end for
Step 3*: Output the net (N_V, M_{Vo}).*
Step 4: End

D. Deadlock Prevention Policy Based on SMSs and Colored Petri Nets

Definition11 [1]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net. The controller subnet for deadlocks of (N, M_o) proposed in [21] is represented as $(V, M_{Vo}) = (P_V, T_V, F_V, M_{Vo})$. Here, (V, M_{Vo}) can be changed by a colored common controller subnet N_{DC} with $N_{DC} = (\{p_{combined1}\}, \{T_{DCi} \cup T_{DCo}\}, F_{DC}, C_{vsi})$, where $p_{combined1}$ is said to be the global place of all control places $P_V = \{V_S \setminus S \in \Pi\}$. $T_{DCi} = \{t | t \in {}^\bullet V_S\}$ is a set of the input (preset) of V_S. $T_{DCo} = \{t | t \in V_S^\bullet\}$ is a set of the output (postset) of V_S. $F_{DC} \subseteq (\{p_{combined1}\} \times \{T_{DCi} \cup T_{DCo}\}) \cup (\{T_{DCi} \cup T_{DCo}\} \times \{p_{combined1}\})$ is said to be a set of directed arcs of N_{DC} that join the $p_{combined1}$ to $(T_{DCi} \cup T_{DCo})$ or $(T_{DCi} \cup T_{DCo})$ to $p_{combined1}$. C_{vsi}, $C_{vsi} \in SC$, $SC = \cup_{i \in Vs}\{C_{vsi}\}$ is the color, which maps $p_{combined1}$ into colors. (N_{DC}, M_{DCo}) represents a colored common controller subnet, and $M_{DCo}(p_{combined1})$ represents the initial tokens with color of the global place and expressed as $\forall V_S \in P_V$, $M_{DCo}(p_{combined1}) = \sum M_{Vo}(V_S)$.

Definition12 [1]. Let $N = (\{p^0\} \cup P_A \cup P_R, T, F, M_o)$ be an S^3PR net and $N_{DC} = \{p_{combined1}\}, \{T_{DCi} \cup T_{DCo}\}, F_{DC}, C_{vsi}, M_{DCo})$ be a deadlock controller for (N, M_o) designed by Definition 11. The composition of (N, M_o) and (N_{DC}, M_{DCo}) is called a a colored controlled S^3PR net, expressed as $(N_C, M_{Co}) = (N, M_o) \parallel (N_{DC}, M_{DCo}$, and $N_C = (P_A \cup \{p^0\} \cup P_R \cup \{p_{combined1}\}, T \cup T_{DCi} \cup T_{DCo}, F \cup F_{DC}, C_R, M_{Co})$.

Theorem1 [1]. Let $N_C = (P_A \cup \{p^0\} \cup P_R \cup \{p_{combined1}\}, T \cup$

$T_{DCi} \cup T_{DCo}$, $F \cup F_{DC}$, C_R, M_{Co}) be colored controlled S^3PR net. Then, the net (N_C, M_{Co}) is live.

Proof. First, all transitions T, T_{DCi}, T_{DCo} in (N_C, M_{Co}) have to be proved to be live. Moreover, no new SMS is occurred, since for all $t_1 \in T$ are live. When all $t_2 \in T_{DCi}$, $p \in {}^{\bullet}t_2$, $M_C(p_i) > 0$, then t_2 can fire. Thus, $M_C(p_{combined1}) > 0$, for all $t_3 \in T_{DCo}$, if $M_C(p_{combined1}) > 0$, then t_3 can fire. Thus, we can say that the net (N_C, M_{Co}) is live.

According to the concepts of colored Petri nets [1] and SMSs, the proposed method is constructed in Algorithm 2.

Algorithm 2: *Strict Minimal Siphon based on Colored Petri Nets Algorithm*

Input: *The nets* (N, M_o) *and* (V, M_{Vo}).

Output: *A net* (N_C, M_{Co}).

Step 1: Combine *all monitors* P_V *into* $p_{combined1}$, ***then*** *do the following steps:*

1. *Insert* $p_{combined1}$ *output arcs* T_{DCo};
2. *Insert* $p_{combined1}$ *input arcs* T_{DCi};
3. *Assign colors* C_{vsi} *for all monitors* P_V;
4. *Compute an initial token with the colors marking of a merged monitor* $M_{DCo}(p_{combined1}) = \sum M_{Vo}(V_S)$.

Step 2: *Add the merged monitor into the net* (N, M_o).

Step 3: *Output a net* (N_C, M_{Co}).

Step 4: End

E. Robust Control for Unreliable Resources Based on Colored Petri Nets

This section describes a robust two-step control policy. In the first step, the system's resources are generally assumed to be reliable. Several monitors are added after applying a strict minimal siphon-based control strategy to such a system. In the second step, by considering that resources may fail, a common recovery subnet is added to model all resource failures in a system. As a result, a robust controlled system is developed. Section II presents the method used in the first step. This section concentrates on the relevance between resource failures and the controlled system in the first step.

Definition13 [17]. Let $r_u \in P_R$ be an unreliable resource. A recovery subnet of r_u is a PN $N_{ri} = (\{p_i, p_{ri}\}, \{t_{fi}, t_{ri}\}, F_{ri})$, where $F_{ri} = \{(p_i, t_{fi}), (t_{fi}, p_{ri}), (p_{ri}, t_{ri}), (t_{ri}, p_i)\}$, and an unreliable resource may fail when it is idle r_u or in a busy state (its holders), $p_i \in \{\{r_u\} \cup H(r_u)\}$. ($N_{ri}$, M_{rio}) is called a marked recovery subnet, where $M_{rio}(p_i) >= 0$ and $M_{rio}(p_{ri}) = 0$.

Definition14. Let $r_u \in P_R$ be an unreliable resource. A colored common recovery subnet of r_u is a PN $N_{cri} = (\{p_i, p_{combined2}\}, \{t_{fi}, t_{ri}\}, F_{cri}, C_{cri})$, where $F_{cri} = \{(p_i, t_{fi}), (t_{fi}, p_{combined2}), (p_{combined2}, t_{ri}), (t_{ri}, p_i)\}$, and an unreliable resource may fail when it is idle r_u or in a busy state (its holders). Thus, we define $P_{RH} = \{r_u\} \cup H(r_u)$ as a set of places, where $H(r_u)$ is a set of holders of r_u, indicated by $H(r_u) = \{p | p \in P_A, p \in {}^{\bullet\bullet}r_u \cap P_A \neq \emptyset\}$, $p_i \in P_{RH}$. C_{cri} is the color that maps $p_i \in P_{RH}$ into colors $C_{cri} \in SC$. (N_{cri}, M_{crio}) is called a colored common marked recovery subnet, where $M_{crio}(p_i) >= 0$ and $M_{crio}(p_{combined2}) = 0$.

In Definition 14, $p_{combined2}$ is called the recovery place of all p_i. Transitions t_{fi} and t_{ri} indicate that an unreliable resource r_u fails in p_i and recovers through $p_{combined2}$, respectively. If an unreliable resource fails in p_i, then the token in p_i flows into $p_{combined2}$ by firing t_{fi}. When transition t_{fi} fires, it adds a color C_{cri} to the tokens from p_i and deposits them into the common place $p_{combined2}$. After the failed resource is repaired, the colored token in $p_{combined2}$ flows into p_i by firing t_{ri}. When transition t_{ri} fires, it selects only the tokens with color C_{cri} from $p_{combined2}$ and deposits them into p_i, indicating that a resource recovery is finished. Note that by default, colors are inherited: when a transition t_{ri} fires, it collects all colors from the consumed ($p_{combined2}$) tokens and passes them to the deposited (p_i) tokens. However, color inheritance can be prevented by overriding.

Definition15. Let (N_C, M_{Co}) be a colored controlled S^3PR, and P_{Ru} be the set of unreliable resources. For all $r_u \in P_{Ru}$, adding one common recovery subnet for all $p_i \in P_{RH}$ results in a colored controlled unreliable Petri net defined as (N_{CU}, M_{CUo}) = (N_C, M_{Co}) ‖ (N_{cri}, M_{crio}) that is the composition of (N_C, M_{Co}) and (N_{cri}, M_{crio}).

Theorem2. The colored controlled unreliable Petri net (N_{CU}, M_{CUo}) is live.

Proof. There is a need to prove that all transitions T, T_F, and T_R in (N_{CU}, M_{CUo}) are live. It does not matter how a controlled system develops, and there are no strict minimal siphons being emptied. In addition, there is no new strict minimal siphon being created, which indicates that the net (N_{CU}, M_{CUo}) is live when there is no failure in an unreliable resource $r_u \in P_{Ru}$ because all $t_1 \in T$ are live. The net (N_{CU}, M_{CUo}) is live when at least one unreliable resource fails but at least one part type can still be processed by the colored controlled unreliable Petri net (N_{CU}, M_{CUo}). Moreover, if a failure in the unreliable resource $r_u \in P_{Ru}$ occurs, then the failed resource r_u will be repaired successfully, the system may be returned to operate without causing a deadlock, and the colored controlled unreliable Petri net (N_{CU}, M_{CUo}) will remain live for all $t_2 \in T_F$ if for all $p_i \in {}^{\bullet}t_2$, $M_{CU}(p_i) > 0$. Then, t_{fi} can fire in any case because it is uncontrollable, leading to $M_{CU}(p_{combined2}) > 0$ for all $t_3 \in T_R$. If $M_{CU}(p_{combined2}) > 0$, then t_3 can fire. It can therefore be said that the colored controlled unreliable Petri net (N_{CU}, M_{CUo}) is live.

Based on strict minimal siphons, unreliable resources, and colored Petri nets, the developed policy is shown as follows:

Algorithm 3: *(Robust control for unreliable resources based on colored Petri net algorithm)*

Input: *A colored controlled net* (N_C, M_{Co}) *by Algorithm 2*

Output: *A colored controlled unreliable marked Petri net* (N_{CU}, M_{CUo})

Step 1: for *each* $r_u \in P_{Ru}$ ***do***

1. *Add a transition to represent breakdown resource* t_{fi}.
2. *Define colors* C_{cri} *for failure transition* t_{fi}.
3. *Add a recovery place* $p_{combined2}$.
4. *Add a recovery transitions to represent that the failed resource is repaired* t_{ri}.
5. *The* p_i *output arcs are connected to the* t_{fi} *transition, and all arc weights are unitary*
6. *The* $p_{combined2}$ *input arcs are connected from the* t_{fi} *transitions, and all arc weights are unitary.*
7. *The* $p_{combined2}$ *output arcs are connected to the* t_{ri} *transitions, and all arc weights are unitary.*
8. *The* p_i *input arcs are connected from the* t_{ri} *transitions, and all arc weights are unitary.*

end for
Step 2*: Output a net (N_C, M_{Co}).*
Step 3: End

III. Numerical Example

To illustrate the proposed methodology, consider an AMS example shown in Fig. 1 [1]. It comprises 11 places and 8 transitions. The places can be defined as the following set partition: $P_A = \{p_2, p_3, \dots, p_7\}$, $P_R = \{p_9, p_{10}, p_{11}\}$, and $P^0 = \{p_1, p_8\}$. The model has 20 reachable markings. In addition, the model has three SMSs, which are

1. $S_1 = p_4 + p_7 + p_9 + p_{10} + p_{11}$,
2. $S_2 = p_4 + p_6 + p_{10} + p_{11}$, and
3. $S_3 = p_3 + p_7 + p_9 + p_{10}$.

Based on the Algorithm 1 the control places are designed as follows:

1. For S_1: ${}^\bullet V_{S1} = \{t_3, t_7\}$, $V_{S1}{}^\bullet = \{t_1, t_5\}$, and $M_{Vo}(V_{S1}) = 2$;
2. For S_2: ${}^\bullet V_{S2} = \{t_3, t_6\}$, $V_{S2}{}^\bullet = \{t_2, t_5\}$, and $M_{Vo}(V_{S2}) = 1$;
3. For S_3: ${}^\bullet V_{S3} = \{t_2, t_7\}$, $V_{S3}{}^\bullet = \{t_1, t_5\}$, and $M_{Vo}(V_{S3}) = 1$.

The controlled Petri net model by using Algorithm 1 is displayed in Fig. 2.

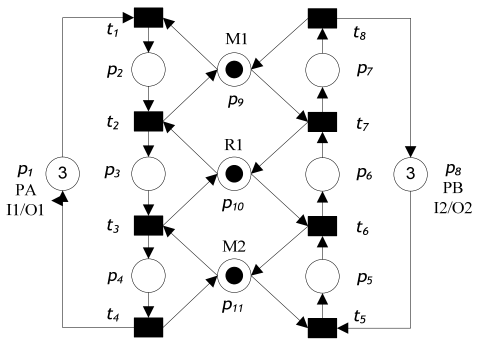

Fig. 1 S^{3}PR Petri net model of an AMS

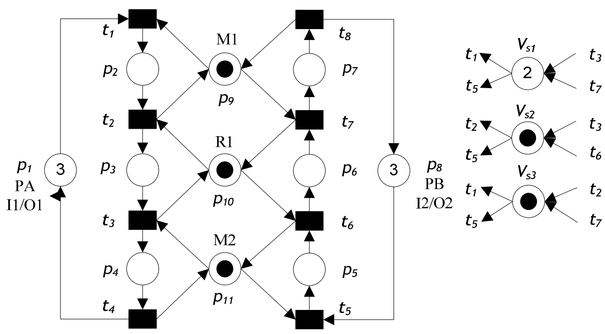

Fig. 2 Controlled S^{3}PR Petri net model by Algorithm 1

By Algorithm 2, a single controller for the controlled Petri net model from Fig. 2 is displayed in Fig. 3. In the net shown in Fig. 3, if transition t_1 fires, it chooses one token of color C_{vs1} from $p_{combined1}$, one token of color C_{vs3} from $p_{combined1}$, one token from resource place p_9, and one token from input place p_1 it deposits them into p_2. In addition, if transition t_2 fires, it chooses one token of color C_{vs2} from $p_{combined1}$, one token from resource place p_{10}, and one token from operation place p_2 and deposits them into p_3. If transition t_5 fires, chooses one token of color C_{vs2} from $p_{combined1}$, one token of color C_{vs2} from $p_{combined1}$, one token of color C_{vs3} from $p_{combined1}$, one token from resource place p_{11}, and one token from input place p_8 and deposits them into p_5. If transition t_2 fires, it generates a color C_{vs3} on the tokens from p_2 and p_{10} and deposits them into $p_{combined1}$. Moreover, if the transition t_3 fires, it generates two colors C_{vs1} and C_{vs2} on the tokens from p_3 and p_{11} and deposits them into $p_{combined1}$. In addition, if transition t_6 fires, it generates a color C_{vs2} on the tokens from p_5 and p_{10} and deposits them into $p_{combined1}$. Finally, if transition t_7 fires, it generates two colors C_{vs1} and C_{vs3} on the tokens from p_6 and p_9 and deposits them into $p_{combined1}$.

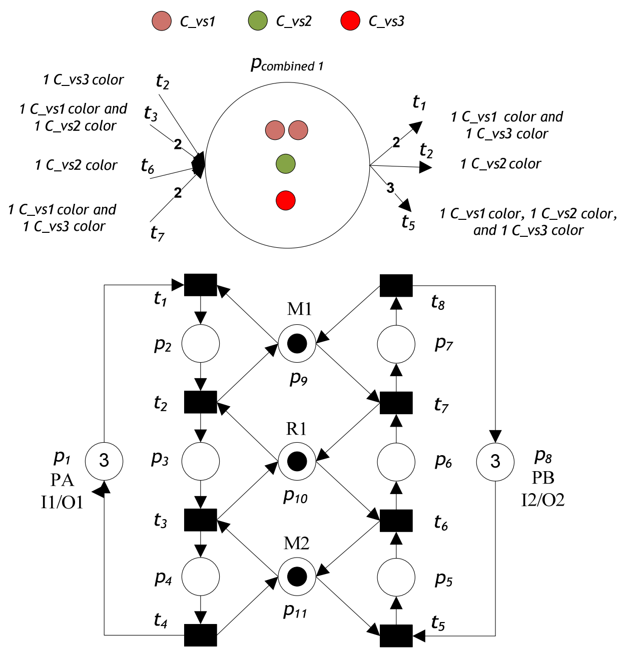

Fig. 3 Colored controlled S3PR Petri net model by Algorithm 2

Considering an unreliable marked S^{3}PR net shown in Fig. 3, we assume that the index set that may be used is NA = $\{i|p_i \in P_{RH}\}$. $T_F = \cup_{i\in NA}\{t_{fi}\}$, $T_R = \cup_{i\in NA}\{t_{ri}\}$, and $C_F = \cup_{i\in NA}\{C_{cri}\}$, where t_{fi}, t_{ri}, and C_{cri} are defined by using Definitions 13 and 14. In the net shown in Fig. 3, we have $P_{Ru} = \{p_9, p_{10}, p_{11}\}$, $H(p_9) = \{p_2, p_7\}$, $H(p_{10}) = \{p_3, p_6\}$, and $H(p_{11}) = \{p_4, p_5\}$. Adding a common recovery subnet for p_9, p_{10}, and p_{11} by Definition 14 results in an unreliable S^{3}PR net model, as depicted in Fig. 4. NA = {2, 3, 4, 5, 6, 7}, $T_F = \{t_{f2}, t_{f3}, t_{f4}, t_{f5}, t_{f6}, t_{f7}\}$, $T_R = \{t_{r2}, t_{r3}, t_{r4}, t_{r5}, t_{r6}, t_{r7}\}$, and $C_R = \{C_{cr2}, C_{cr3}, C_{cr4}, C_{cr5}, C_{cr6}, C_{cr7}\}$.

If resource p_9 fails in busy state p_2 or p_7, i.e., t_{f2} or t_{f7} fires, it adds a color C_{cr2} or C_{cr7} to the token from p_2 or p_7 and deposits them into the common place $p_{combined2}$. After the failed resource p_9 is repaired, the colored token in $p_{combined2}$ flows into p_2 or p_7 by firing t_{r2} or t_{r7} when the transition t_{r2} or t_{r7} fires, and selects only the tokens with colors C_{cr2} or C_{cr7} from $p_{combined2}$ and deposits them into p_2 or p_7, indicating that the recovery of resource p_9 is finished. When resource p_{10} fails in busy state p_3 or p_6, i.e., t_{f3} or t_{f6} fires, it adds a color C_{cr3} or C_{cr6} to the token from p_3 or p_6 and deposits them into the common place $p_{combined2}$. After the failed resource p_{10} is repaired, the colored token in $p_{combined2}$ flows into p_3 or p_6 by firing t_{r3} or t_{r6} when the transition t_{r3} or t_{r6} fires, and selects only the tokens with colors C_{cr3} or C_{cr6} from $p_{combined2}$ and deposits them into p_3 or p_6, indicating that the recovery of resource p_{10} is finished. Finally, if resource p_{11} fails in busy state p_4 or p_5, i.e., t_{f4} or t_{f5} fires, it adds a color C_{cr4} or C_{cr5} to the token from p_4 or p_5 and deposits them into the common place $p_{combined2}$. After the failed resource p_{11} is repaired, the colored token in $p_{combined2}$ flows into p_4 or p_5 by firing t_{r4} or t_{r5} when the transition t_{r4} or t_{r5} fires, and selects only the tokens with colors C_{cr4} or C_{cr5} from $p_{combined2}$ and deposits them into p_4 or p_5, indicating that the recovery of resource p_{11} is finished.

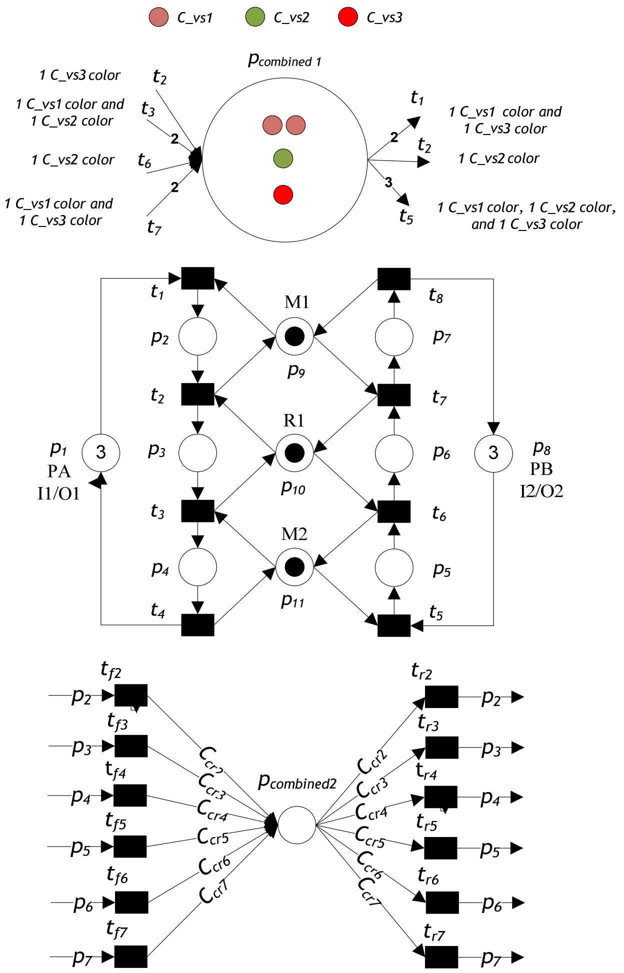

Fig. 4 Colored controlled unreliable Petri net model by Algorithm 3

IV. Conclusion

Various studies on the prevention of deadlocks have been devoted to enforcing liveness on the premise that all resources in a system work correctly [1]-[11], [24]-[26]. However, in practice, AMSs may face unexpected resource failures. Therefore, this research suggested a robust deadlock-prevention controller for an AMS with all unreliable resources. In the absence or existence of resource failure, the proposed robust controller can ensure the required properties, i.e., the liveliness of the controlled system. A three-step policy was developed in this paper to address the robust deadlock control of AMSs. The strict minimal siphon-based policy in [21] was utilized in the first step for a specific plant, resulting in a controlled net without considering resource failures. The second step uses an approach based on the colored Petri net to merge all monitors designed in the first step into a global monitor to make all SMSs marked. Then, resource failures were considered by adding a common recovery subnet in a unified modeling method using a Petri net. The controlled net can work smoothly even if unreliable resources fail as long as there is no failure of one unit of unreliable resource. The main advantages of the proposed method are: (1) It can be applied to an unreliable complex Petri net model for AMSs; (2) It has less computational complexity for the computation of the common recovery subnet; (3) It can obtain one common recovery subnet to model all resource failures; (4) It has a simpler structure; (5) It does not need to compute reachability graphs, which means that it has small computational overhead; (6) It can easily handle robust deadlock control problems with all unreliable resources without using inhibitor arcs.

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, 2020.

[2] Y. Chen, Z. Li, M. Khalgui, and O. Mosbahi, "Design of a maximally permissive liveness-enforcing Petri net supervisor for flexible manufacturing systems," *Automation Science and Engineering, IEEE Transactions on,* vol. 8, no. 2, pp. 374-393, 2011.

[3] A. Ghaffari, N. Rezg, and X. Xie, "Design of a live and maximally permissive Petri net controller using the theory of regions," *Robotics and Automation, IEEE Transactions on,* vol. 19, no. 1, pp. 137-141, 2003.

[4] M. Uzam, "The use of the Petri net reduction approach for an optimal deadlock prevention policy for flexible manufacturing systems," *The International Journal of Advanced Manufacturing Technology,* vol. 23, no. 3-4, pp. 204-219, 2004.

[5] D. Sun, Y. Chen, M. A. El-Meligy, M. A. F. Sharaf, N. Wu, and Z. 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.

[6] D. Y. Chao, "Improvement of suboptimal siphon-and FBM-based control model of a well-known S^{3}PR," *Automation Science and Engineering, IEEE Transactions on,* vol. 8, no. 2, pp. 404-411, 2011.

[7] M. Uzam, "An optimal deadlock prevention policy for flexible manufacturing systems using Petri net models with resources and the theory of regions," *The International Journal of Advanced Manufacturing Technology,* vol. 19, no. 3, pp. 192-208, 2002.

[8] Z. Li and M. Zhou, "Elementary siphons of Petri nets and their application to deadlock prevention in flexible manufacturing systems," *Systems, Man and Cybernetics, Part A: Systems and Humans, IEEE Transactions on,* vol. 34, no. 1, pp. 38-51, 2004.

[9] M. Uzam and M. Zhou, "Iterative synthesis of Petri net based deadlock prevention policy for flexible manufacturing systems," in *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, 2004, pp. 4260-4265.

[10] 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," *Journal of Algorithms & Computational Technology,* vol. 11, no. 3, pp. 261-270, 2017.

[11] M. Zhao and M. Uzam, "A suboptimal deadlock control policy for

designing non-blocking supervisors in flexible manufacturing systems," *Information Sciences,* vol. 388, pp. 135-153, 2017.

[12] M. Abdulaziz, E. A. Nasr, A. Al-Ahmari, H. Kaid, and Z. Li, "Evaluation of Deadlock Control Designs in Automated Manufacturing Systems," in *International Conference on Industrial Engineering and Operations Management*, USA, 2015.

[13] 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 Transactions on Automatic Control,* vol. 62, no. 8, pp. 4173-4179, 2017.

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

[15] S. Wang, S. F. Chew, and M. A. Lawley, "Using shared-resource capacity for robust control of failure-prone manufacturing systems," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans,* vol. 38, no. 3, pp. 605-627, 2008.

[16] S. F. Chew, S. Wang, and M. A. Lawley, "Robust supervisory control for product routings with multiple unreliable resources," *IEEE Transactions on Automation Science and Engineering,* vol. 6, no. 1, pp. 195-200, 2009.

[17] G. Liu, Z. Li, K. Barkaoui, and A. M. Al-Ahmari, "Robustness of deadlock control for a class of Petri nets with unreliable resources," *Information Sciences,* vol. 235, pp. 259-279, 2013.

[18] 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 Transactions on Systems, Man, and Cybernetics: Systems,* vol. 47, no. 11, pp. 3019-3029, 2016.

[19] G. Liu, P. Li, Z. Li, and N. Wu, "Robust Deadlock Control for Automated Manufacturing Systems With Unreliable Resources Based on Petri Net Reachability Graphs," *IEEE Transactions on Systems, Man, and Cybernetics: Systems,* vol. 49, no. 7, pp. 1371-1385, 2018.

[20] N. Ran, J. Hao, S. Wang, Z. Dong, Z. He, Z. Liu, and Y. Ruan, "K-codiagnosability verification of labeled Petri nets," *IEEE Access,* pp. 1-8, 2019.

[21] J. Ezpeleta, J. M. Colom, and J. Martinez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *Robotics and Automation, IEEE Transactions on,* vol. 11, no. 2, pp. 173-184, 1995.

[22] Z. Li and M. Zhou, *Deadlock resolution in automated manufacturing systems: a novel Petri net approach*: Springer Science & Business Media, 2009.

Husam Kaid is a researcher and PhD student in the Industrial Engineering Department, College of Engineering, King Saud University, Saudi Arabia. He received his BS in Industrial Engineering from the University of Taiz, Taiz, Yemen, in 2010. He received his MS in Industrial Engineering from the King Saud University, Saudi Arabia, in 2015. His research areas and specialties are design and analysis of manufacturing systems, deadlock control in manufacturing systems, supply chain, simulation, operations research, optimization techniques, and bibliometric network analysis.

Abdulrahman Al-Ahmari a Professor of industrial engineering with King Saud University, Riyadh, Saudi Arabia. He received the Ph.D. degree in manufacturing systems engineering from the University of Sheffield, Sheffield, U.K., in 1998. He worked as Dean of the Advanced Manufacturing Institute, Chairman of Industrial Engineering Department and He led a number of funded projects from different organizations in Saudi Arabia. He has published papers in leading Journal of Industrial and Manufacturing Engineering. His current research interests include advanced manufacturing technologies, Petri nets, analysis and design of manufacturing systems, computer integrated manufacturing, optimization of manufacturing operations, flexible manufacturing systems and cellular manufacturing systems, and applications of decision support systems in manufacturing.

Zhiwu Li (M'06-SM'07-F'16) received the B.S., M.S., and Ph.D. degrees in mechanical engineering, automatic control, and manufacturing engineering, respectively, all from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He joined Xidian University in 1992. Over the past decade, he was a Visiting Professor at the University of Toronto, Technion (Israel Institute of Technology), Martin-Luther University, Conservatoire National des Arts et Métiers (Cnam), Meliksah Universitesi, University of Cagliari, University of Alberta, and King Saud University. Currently, He is also with Macau University of Science and Technology. His current research interests include Petri net theory and application, supervisory control of discrete event systems, workflow modeling and analysis, system reconfiguration, game theory, and data and process mining. He serves as a reviewer for 90+ international journals. He is listed in Marquis Who's Who in the world, 27th Edition, 2010. Dr. Li is a recipient of an Alexander von Humboldt Research Grant, Alexander von Humboldt Foundation, Germany, and Research in Paris, France. He is the founding chair of Xi'an Chapter of IEEE Systems, Man, and Cybernetics Society.