IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received October 28, 2021, accepted November 11, 2021, date of publication November 15, 2021, date of current version December 3, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3128206

Trust-Based Petri Net Model for Fault Detection and Treatment in Automated Manufacturing Systems

EBRAHIM ALI ALZALAB[1], AHMED M. EL-SHERBEENY[2], MOHAMMED A. EL-MELIGY[2], AND HAFIZ TAYYAB RAUF[3]

[1]School of Electro-Mechanical Engineering, Xidian University, Xi'an 710071, China
[2]Industrial Engineering Department, College of Engineering, King Saud University, Riyadh 11421, Saudi Arabia
[3]Department of Computer Science, Faculty of Engineering and Informatics, University of Bradford, Bradford BD7 1DP, U.K.

Corresponding author: Ahmed M. El-Sherbeeny (aelsherbeeny@ksu.edu.sa)

This work was supported by the King Saud University, Riyadh, Saudi Arabia, under Project RSP-2021/133.

ABSTRACT Automated manufacturing systems (AMSs) are vulnerable to failures. Trust evaluation is becoming a novel technique for detecting faults in AMSs. In this article, a two-step robust deadlock control strategy for systems with unreliable resources is proposed. Recovery subnets are added for all failure-prone resources of the system model in the first step based on Petri nets. The recovery subnets are applied to a system to ensure the system works in a reliable model. In the second step, a novel hybrid technique combining a trust-model and colored Petri nets is proposed to the detection and handling of faults obtained in the first step. The proposed approach includes the descriptive features of a modular Petri net integration with the trust method. It provides the combination of three kinds of procedures: management of all resource failures, detection, and treatment of faults in a system. Consequently, the proposed model considers not only resolving resource failures in AMSs, but also their treatment once they are detected. Finally, an example from the literature is used to test the proposed approach.

INDEX TERMS Fault detection, fault treatment, deadlock, colored Petri net, automated manufacturing system, trust model.

I. INTRODUCTION

Automated manufacturing systems (AMSs) are one of the typical real discrete event systems [1]. The distribution of resources like automated tools, robots, buffers, machines, and automatic guided vehicles, allows several types of products to enter into a system at discrete times and there are asynchronous operations in the system. In an AMS, each element of the set of resources can be treated according to a specified process concatenation. Nevertheless, this distribution of resources can result in deadlocks; consequently, some processes may not be completed as expected. Therefore, deadlock control is necessary and sometimes mandatory, particularly in highly automated production processes. The inability of a system's element to perform its function is referred to as a fault [2], and it is synonymous with disturbances, mistakes, failures, or errors that lead to undesirable equipment behavior [3]. In an actual production system, resource failures cannot be ignored. However, most previous studies take into consideration the definition of the process and optimization under ideal conditions. In order to guarantee effective, safe, and reliable processes in AMSs and to avoid performance degradation, early detection and handling of faults are of utmost importance. Therefore, a robust deadlock prevention strategy must be developed that can perform failure detection, diagnosis, and recovery to guarantee a deadlock-free operation of AMSs.

In [4], [5], Petri nets are considered a commonly used tool for the design, analysis and control of deadlocks of AMSs. They are thought of as outstanding tools for modelling the features and behaviour of AMSs, such as synchronisation, conflict, sequencing, and causal dependency. Petri nets can also be used to model behavioural characteristics, such as liveness and boundedness [6]–[8]. Most of the deadlock prevention policies using Petri nets are based on the reachability graph [9]–[11] or structural analyses [5], [13]. These policies

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

suffer from the problems of structural complexity, computational complexity, and behavioral permissiveness [7], [11]. There have also been studies in the literature that deal with the evolution of deadlock control policies [62]. Most studies in the literature assume that the resources in an AMS are reliable [12], [14]–[21], while others assume they are unreliable [6], [22]–[32].

The study in [27] proposes two policies to tackle deadlocks arising as a result of failures of failure-prone resources in single resource allocation systems. The first approach considers only one unreliable resource, while the second considers multiple unreliable resources. In [26], two policies are developed to deal with deadlocks in an AMS with unreliable resources by utilizing central buffers. The idea of using recovery subnets for unreliable resources is introduced in [25]. Control places are designed to prevent deadlocks in the presence or absence of a resource failure. In some cases, a waiting period is necessary, which leads to a significant decrease in the use of resources. Two robust deadlock control methods are presented in [6] based on a reachability graph analysis technique for GS^3PRs (Generalised Systems of Simple Sequential Processes with Resources) with unreliable resources. The markings are classified into prohibited and legitimate ones. Consequently, by avoiding the prohibited markings, the control of deadlocks in a robust manner is achieved. A two-step deadlock control strategy and robust supervisory control are proposed in [33].

An elementary siphon policy, consisting two steps, is reported in [34] to configure control places to ensure that a system is deadlock-free without considering resource failures in the first step. The second step deals with the deadlock problems produced by resource failures. Recovery subnets are used to model failures. In order to suggest a deadlock control approach to an AMS with several unreliable resources, Yue *et al.* [24] used a modified Banker algorithm and propose a set of resource capacity restrictions. The authors of [35] suggest a deadlock control strategy together with unreliable resources for S^3PRs using the elementary siphons theory [36], [37]. A robust deadlock strategy for feedback control consisting of a two-stage system of shared resources with unreliable resources is presented [6]. A controlled system (deadlock-free) is achieved in the first stage by controlling strict minimal siphons without taking resource failures into account. The second stage involves addressing the deadlock caused by resource failures. To ensure the reliability of the system, a recovery subnet is employed in the system earned in the first stage.

Within the last few years, various methods have been suggested to tackle AMSs faults. The work in [38] presents an approach for examining the behavior of faults on resources in failure-prone discrete event systems, addresses the fault-tolerant problem, and offers a solution to keep the system fulfilling its obligations while the resource failures are repaired and recovered.

The authors of [39] propose a fault detection and diagnostic method in industrial processes. It is a multi-layered feed-forward neural network approach that needs concurrent monitoring of multi-data. Their results demonstrate that neural network-based methods are effective against detecting and diagnosing related defects. Moreover, for real industrial processes, this approach may also be sufficiently generalised. In [40] a distributed Petri net approach is used to construct a supervisory structure for manufacturing systems to detect and treat the operation failures of a system. The proposed technique is effective, and several diagrams are created to depict the operational units for AMSs. It could use networks that have already been analyzed without taking the conditions that are anomalous into consideration for modelled systems.

A novel three-step deadlock control technique is proposed in [43] for failure detection and repair of a system with unreliable resource. An approach for modeling and evaluating fault-tolerant industrial systems is suggested in [42], which enhances regular manufacturing processes in the detection and handling of faults, which integrates networks and Petri nets hierarchically. The distributed Petri net is presented in [44] to develop a control mechanism for identifying and remediating defects in manufacturing system operations. The proposed technique is efficient and several graphs depicting operating units are produced effectively. In [41] Bayesian and Petri nets are incorporated to tackle fault detection and repair in AMSs. Automated recovery processes are also studied in machining processes and electric autonomous driven vehicles.

In [45], fault detection methods are divided into three classes, namely model-based, knowledge-based, and signal-based ones, to reduce faults in manufacturing industry and to establish methods for understanding and diagnosing faults, which can achieve satisfied results in cost reduction and improve quality control with several other advantages. On the basis of mathematical models of the electrical machines, observers for residual generation can be created in order to isolate faults, and online parametric identification is proposed as a diagnosis tool, since some faults affect primarily self and mutual inductance and friction coefficients [46]. In [47], the authors propose a fault diagnostic technique for AMSs. This technique guarantees proper maintenance and operation of a system and consists of three parts: identification, diagnosis, and decision-making parts. Kaid *et al.* [5] presented a two-step deadlock control strategy established on Petri nets for AMSs in which the structural complexity of the Petri net supervisor is greatly reduced. In [48], the authors propose a system for fault detection modelled on partially-observable timed hybrid Petri nets. Liu *et al.* [35] studied robust deadlock control problems of AMSs with numerous unreliable resources by employing a deadlock detection technique. In [6], conventional Petri nets are utilised to model, control, simulate, and analyze a multi-unit resource system.

Zhang *et al.* [49] suggested an adaptive deadlock controller based on elementary siphons that can switch system modes between regular and robust modes based on resource failures and normal operating modes. Feng *et al.* [50] establish a deadlock prohibition controller of AMSs by considering resource failures. If one of the unreliable resources fails, the suggested controller guarantees that the system is able to process parts continuously. Du and Hu [51] propose a recursive deadlock control technique for AMSs with multiple unreliable resources. The controlled system is able to continue to work smoothly even if certain unreliable resources fail. Based on structural analyses using three-step colored Petri nets, the work in [52] develops a robust deadlock control strategy, aiming at shared resources and unreliable resources within AMSs.

In [53] a trust-based formal model (TFM) is designed for fault detection of wireless sensor networks and, based on TFM, the detection of a fault process is established. Formal models for public-key infrastructures for security in distributed systems (trust models) donate immensely to a deeper comprehension of the desirable design essentials based on the modelling technique of coloured Petri nets, as seen in [54].

Trust-model networks are of considerable importance as a problem-resolving technique in detection and handling of faults. The feature of trust-model networks is that it can handle noisy or partial inputs generally. Networks of trust-models may also deal with continuous input data. The study must be made easier to resolve the problem of failure diagnosis and detection. In [56], a problem of finding the maximal number of forbidden first-met bad marking, namely MFFBMP1, is addressed in order to prevent as many first-met bad markings (FBMs) as possible by construct a place invariant (PI). To present a three-step controller synthesis technique to detect fault and handle an unreliable system resource is the main objective of this research. The controlled system is achieved in the first step using a linear programming problem, namely (MFFBMP1), without considering resource failures. The deadlock control is affected by resource failures in the second step. If there is a resource failure in the Petri net model, a recovery subnet is added. Adding the recovery subnet ensures that the system is reliable in the first step. To detect and treat faults in the second step, the third step is a novel hybrid methodology that combines a trust-based model with colored Petri nets. The main contribution of this strategy is as follows: (1) It is more robust compared with the previous works in the literature; (2) the proposed supervisor has an uncomplicated structure; (3) it combines three kinds of procedures: a deadlock-free system without considering a resource failure, fault detection, and fault treatment; (4) it takes into consideration not only the solution of AMSs deadlock problems but also failure detection and treatment.

The structure of this article is organized as follows. Section II elaborates upon the programming problem MFFBMP1 and trust-based colored Petri nets and presents basic notions of Petri nets and the deadlock avoidance policy for fault detection and treatment. Section III illustrates a real-world AMS case study. The future research and conclusions are provided in Section IV.

II. PRELIMINARIES

A. BASICS OF PETRI NETS

This section presents the fundamentals of Petri nets, a deadlock avoidance policy based on colored Petri nets and an iterative method, as well as a trust-based colored Petri net for fault detection and handling.

Definition 1 [55]: A Petri net is a four-tuple $N = (P, T, F, W)$, where P and T are nonempty, finite, and disjoint sets. P and T are the sets of places and transitions respectively, with $P \cap T = \emptyset$ and $P \cup T \neq \emptyset$. $F \subseteq (P \times T) \cup (T \times P)$ is called the flow relation of the net, represented by arcs with arrows from places to transitions or from transitions to places. $W : F \rightarrow \mathbb{N}$ is a mapping that assigns a weight to an arc: $W(f) > 0$ if $f \in F$, $W(f) = 0$ otherwise, where $\mathbb{N} = \{0, 1, 2, \ldots\}$ is the set of non-negative integers. $N = (P, T, F, W)$ is said to be an ordinary net, denoted as $N = (P, T, F)$, if for all $f \in F$, $W(f) = 1$. A marking M of N is a mapping from P to $\mathbb{N}$. Given a marking M, $M(p)$ is the marking of place p. (N, M_0) is said to be a marked net or net system. For economy of space, we use $\sum_{p\in P} M(p)p$ to indicate marking M. For mathematical convenience, a marking M can be written as a vector.

Definition 2: Let $h \in P \cup T$ be a node of net $N = (P, T, F, W)$. The preset of h is defined as ${}^\bullet h = \{q \in P \cup T \mid (q, h) \in F\}$; the postset of h is defined as $h^\bullet = \{q \in P \cup T \mid (h, q) \in F\}$. This notation can be extended to a set of nodes in the following way: given $H \subseteq P \cup T$, ${}^\bullet H = \cup_{h\in H} {}^\bullet h$, and $H^\bullet = \cup_{h\in H} h^\bullet$. Given place p, we denote $\max\{W(p, t) \mid t \in p^\bullet\}$ by $\max_{p^\bullet}$.

Definition 3: A transition $t \in T$ is said to be enabled at a marking M if for all $p \in {}^\bullet t$, $M(p) \geq W(p, t)$; this fact is denoted as $M[t\rangle$. Firing t produces a new marking M' such that for all $p \in P$, $M'(p) = M(p) - W(p, t) + W(t, p)$, denoted as $M[t\rangle M'$. Marking M' is said to be reachable from M if there exist a sequence of transitions $\sigma = t_0 t_1 \cdots t_n$ and markings $M_1, M_2, \ldots, M_n$ such that $M\,[t_0\rangle\, M_1\,[t_1\rangle\, M_2 \cdots M_n\,[t_n\rangle\, M'$ holds. $R(N, M)$ indicates the set of markings reachable from M in N, i.e., $R(N, M) = \{M' \in \mathbb{N}^{|P|} \mid (\exists \sigma \in T^*) M[\sigma\rangle M'\}$.

Definition 4: A net is pure (self-loop free) if there do not exist $h, q \in P \cup T$ such that $(h, q) \in F \wedge (q, h) \in F$. A pure net $N = (P, T, F, W)$ can be alternatively described by its incidence matrix $[N]$, where $[N]$ is a $|P| \times |T|$ integer matrix with $[N](p, t) = W(t, p) - W(p, t)$.

Definition 5: A transition $t \in T$ is live at M_0 if for all $M \in R(N, M_0)$, there exists $M' \in R(N, M)$, $M'[t\rangle$ holds. N is dead at M_0 if there does not exist $t \in T$ such that $M_0[t\rangle$ holds. (N, M_0) is deadlock-free if for all $M \in R(N, M_0)$, there exists $t \in T$, $M[t\rangle$. (N, M_0) is live if for all $t \in T$, t is live at M_0. (N, M_0) is bounded if there exists $k \in \mathbb{N}$, for all $M \in R(N, M_0)$, for all $p \in P$, $M(p) \leq k$ holds.

Definition 6: A column vector $I : P \rightarrow \mathbb{Z}$ indexed by P is a P-vector and a column vector $J : T \rightarrow \mathbb{Z}$ indexed by T is a T-vector, where $\mathbb{Z}$ is the set of integers. A P(T)-vector $I(J)$ can be represented by $\sum_{p\in P} I(p)p$ $\left(\sum_{t\in T} J(t)t\right)$ for economy of space. A column vector in which each entry equals 0(1) is represented by $\mathbf{0}(\mathbf{1})$. $[N]^T$ and I^T are the transposed versions of matrix $[N]$ and vector I, respectively. P-vector I is a P-invariant (place invariant) if $I \neq \mathbf{0}$ and $I^T[N] = \mathbf{0}^T$. P-invariant I is called a P-semiflow if no element of I is negative. $\|I\| = \{p \in P \mid I(p) \neq 0\}$ is said to be the support of I. $\|I\|^- = \{p \mid I(p) < 0\}$ indicates the negative support of I, while $\|I\|^+ = \{p \mid I(p) > 0\}$ indicates the positive support of P-invariant I. When the support of an invariant is not a strict superset of any other invariant support and the greatest common divisor of its entries is one, it is said to be minimal. If I is a P-invariant of (N, M_0), then for all $M \in R(N, M_0)$, $I^T M = I^T M_0$.

Definition 7: Let (N, M_0) be a Petri net system. Marking M_0 is reversible if M_0 is reachable from M' for all markings $M' \in R(N, M_0)$.

A class of Petri nets, namely System of Simple Sequential Processes with Resources (S^{3}PR), that can be used to model automated manufacturing systems is proposed in [59]. The net class has been received much attention from many researchers and even practitioners. This paper proposes a new class of Petri nets based on S^{3}PR nets, called colored S^{3}PR nets. Due to the limited space, the related definitions regarding S^{3}PR are not presented here. For details, see [59].

Definition 8: Net system $N = (P_A \cup P_R \cup P^0, T, W, F, C_f, S_C, A_f, N_f, G_f, I_f, M_0)$ is a colored S^{3}PR net if the following statements are satisfied:

(1) P_A, P_R, and P^0 denote the sets of activity places, resource places, and process idle places, respectively; T, W, F, and M_0 are defined as above.

(2) C_f is the function of colors, which assigns a color c_i to a place $p \in P_C$, where $c_i \in S_C$ and S_C is a set of colors.

(3) A_f is the arc function that tracks the term e to all flow (arc) $f \in F$.

(4) N_f is the function of the nodes which tracks F into $(P_C \times T) \cup (T \times P_C)$.

(5) G_f is the guard function that traces a Boolean value to a guard expression g for all transitions $t \in T$.

(6) I_f is the initialization function that in an initialization expression tracks each place $p \in P_C$.

B. DEADLOCK AVOIDANCE POLICY PREMISED ON COLORED PETRI NETS AND AN ITERATIVE METHOD

MFFBMP1 is developed in this section by using place invariants (PI) that can prevent the largest possible number of first-met bad markings (FBMs). Subsequently, the vector covering method is used to define the minimum coverage range for legal labeling and FBMs [3]. Invariant place coefficients and monitors can then be derived by the resolution of the integer linear programming problem (ILPP). The iterative monitor design process carried out is as follows. A PI aims to inhibit the largest possible amount of FBMs at each iteration. Every FBM that the PI forbids is omitted from the minimum covered set of FBM. This approach terminates if the prohibition of all the minimum covering set of FBMs are achieved.

Definition 9 [57]: Let (N, M_0) be an S^{3}PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$, and let M and M' be two markings in $\mathcal{M}_{\text{FBM}}$. M covers M' (or M' is covered by M) if for all $p \in P$, $M(p) \geq M'(p)$, which is represented as $M \geq M'$ (or $M' \leq M$). Marking M A-covers M' (or M' is A-covered by M) if for all $p \in P_A$, $M(p) \geq M'(p)$, which is represented as $M \geq_A M'$ (or $M' \leq_A M$).

Definition 10 [57]: Let (N, M_0) be an S^{3}PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$, and let $\mathcal{M}_{\text{FBM}}$ be the set of FBMs in (N, M_0). For all $M \in \mathcal{M}_{\text{FBM}}$, a subset of $\mathcal{M}_{\text{FBM}}$ that A-covers M is defined as $F_M = \{M' \in \mathcal{M}_{\text{FBM}} \mid M' \geq_A M\}$.

Definition 11 [57]: Let (N, M_0) be an S^{3}PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$ and M_L be the set of legal markings. For all $M \in M_L$, the subset of M_L that is A-covered by M is represented as $R_M = \{M' \in M_L \mid M' \leq_A M\}$.

Definition 12 [57]: Let (N, M_0) be an S^{3}PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$. A subset of FBMs in N is M^*_{FBM}, referred to as the minimal covered set of FBMs, fulfilling:

1) for all $M \in M_{\text{FBM}}$, there exists $M' \in M^*_{\text{FBM}}$, subject to $M \geq_A M'$; and
2) for all $M \in M^*_{\text{FBM}}$, there does not exist $M'' \in M^*_{\text{FBM}}$ such that $M \geq {}_A M''$ and $M \neq M''$ hold.

Definition 13 [57]: Let (N, M_0) be an S^{3}PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$ and M^*_L be a subset of legal markings in N. It is said to be the minimal covered set of legal markings if

1) for all $M \in M_L$, there exists $M' \in M^*_L$ such that $M' \geq_A M$, and
2) for all $M \in M_L$, there does not exist $M'' \in M^*_L$ such that $M'' \geq_A M$ and $M \neq M''$.

The techniques of computing the sets of legal markings M_L, the set of FBMs M_{FBM}, the minimal set of FBMs M^*_{FBM} and the minimal set of legal markings M^*_L are presented in [57]. The model developed in [56] is illustrated by the mathematical model below.

MFFBMP1:

$$\max f = \sum\nolimits_{k \in N^*{}_{FBM}} f_k$$

$$\text{Subject to} \sum_{i \in N_A} l_i \cdot M_i(p_i) \leq \beta, \forall M_i \in M^*_L \tag{1}$$

$$\sum_{i \in N_A} l_i \cdot M_k(p_i) \geq \beta + 1 - (1 - f_k) \cdot Q, \forall M_k \in M^*_{\text{FBM}}$$

$$l_i \in \{0, 1, 2, 3, \ldots\}$$

$$\beta \in \{1, 2, 3, \ldots\}$$

$$f_k \in \{0, 1\}, \quad \forall k \in N^*_{FBM}. \tag{2}$$

To maximize the number of FBMs that are prevented by a place PI, the objective function f is utilized. Indicate by f^* its optimum value. In Definitions 12 and 13, M^*_L and M^*_{FBM} are defined, respectively. N^*_{FBM} is utilised to describe $\{i \mid M_i \in M^*_{\text{FBM}}\}$. If $f^* = 0$, we have $f_k = 0$, for all $k \in N^*_{\text{FBM}}$, implying that no FBMs in M^*_{FBM} can be prevented via the place invariant. In (2), let I be a place invariant, while β is a positive integer variable, and l_i's ($\{i \mid p_i \in P_A\}$) are the coefficients of I, for all $M_l \in M^*_L$. $M_l(p_i)$ is the number of tokens in the legal markings minimum covering set. All legal markings must be retained after adding a monitor, which means that each marking $M_l \in M_L^*$ cannot be prohibited. It must fulfill the coefficient l_i's in (2), which is referred to as the reachability condition. In (3), the positive integer Q is a constant and must be large enough; in a minimal covering set of FBMs the number of tokens is denoted by $M_k(p_i)$; f_k is a set of M^*_{FBM} variables and for all $M_k \in M^*_{\text{FBM}}$, $f_k \in \{0, 1\}$, $k = 1, 2, \ldots$. Furthermore, in (3), $f_k = 0$ indicates that this restriction is redundant; $f_k = 1$ denotes that M_k is prevented by I and M_k cannot be prevented by I.

Definition 14: Let (N_h, M_h) and $\left(N_q, M_q\right)$ be two S^3PRs with $N_h = (P_{Ch}, T_h, F_h, W_h)$ and $N_q = (P_{Cq}, T_q, F_q, W_q)$. A concurrent Petri net (N_s, M_s) is generated from the integration (composition) of (N_h, M_h) and $\left(N_q, M_q\right)$, denoted as $(N_h, M_h) \parallel \left(N_q, M_q\right)$, satisfying: (i) $P_s = P_h \cup P_q$ and $P_h \cap P_q = \emptyset$. (ii) $T_s = T_h \cup T_q$, (iii) $F_s = F_h \cup F_b$, (iv) $M(p) = M_i(p), p \in P_{i,} i = h, q$, and (v) $W(b) = W_i(b)$, where $b \in F_i$, $i = h, q$.

Let (N, M_0) be an S^3PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$. We use (DC, M_{DCo}) to represent a deadlock controller for (N, M_0) via the methods in [56], where $(DC, M_{DCo}) = (P_{DC}, T_{DC}, F_{DC}, M_{DCo})$. It is possible to simplify (DC, M_{DCo}) and replace it with a colored subnet of common deadlock control. A Petri net $N_{CC} = (\{p_{\text{combined 1}}\}, F_{CC}, C_{vsi}, \{T_{CCi} \cup T_{CCo}\}, M_{CCo})$, where $p_{\text{combined 1}}$ is referred to as the combined monitor of P_{DC}. $T_{CCo} = \{t \mid t \in V_S^{\bullet}\}$ is the set of output transitions of V_S and $T_{CCi} = \{t \mid t \in {}^{\bullet}V_S\}$ is the set of input transitions of V_S. $F_{CC} \subseteq (\{p_{\text{combined 1}}\} \times \{T_{CCi} \cup T_{CCo}\}) \cup (\{T_{CCi} \cup T_{CCo}\} \times \{p_{\text{combined 1}}\})$ is referred to as a flow relation of N_{CC}. C_{cri} is the color that maps $\{p_{\text{combined 1}}\}$ into colors $C_{vsi} \in C_R$, in which $C_R = \cup_{i \in V_s}\{C_{vsi}\}$. (N_{CC}, M_{CCo}) is referred to as a colored subnet of common deadlock control. $M_{CCo}(p_{\text{combined 1}})$ is an initial marking including the combined monitor's color markings, represented as $M_{CCo}(p_{\text{combined 1}}) = \sum M_{DCo}(V_S)$, $V_S \in P_{DC}$.

Let (N, M_0) be an S^3PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$. As shown before, (DC, M_{DCo}) is a deadlock controller for N due to [56], represented as $(DC, M_{DCo}) = (P_{DC}, T_{DC}, F_{DC}, M_{DCo})$. Now we explain the related nations in (DC, M_{DCo}):

(i) $P_{DC} = \left\{V_S \backslash V_S \in V_p\right\}$ is the control places set, and $V_p = \{V_{S1}, V_{S2}, V_{S3}, \ldots, V_{Si}\}$ is the control places set to be revealed. (ii) $T_{DC} = \left\{t \mid t \in {}^{\bullet}V_S \cup V_S^{\bullet}\right\}$. (iii) $F_{DC} \subseteq (P_{DC} \times T_{DC}) \cup (T_{DC} \times P_{DC})$ is said to be a flow relation of (M_{DCo}, DC) (iv) The initial marking of a control place is $M_{DCo}(\text{V}_\text{S})$, defined as $M_{DCo}(V_S) = \beta$, where $V_S \in P_{DC}$. (N_{DC}, M_{DCo}) is said to be a controlled Petri net model resulting from the integration of (N, M_0) and (DC, M_{DCo}), represented as $(N, M_0) \parallel (DC, M_{DCo})$

Definition 15 [4]: Let (N, M_0) be an S^3PR with $N = (P_A \cup P_R \cup P^0, T, F, W)$. A common colored deadlock control subnet is a common colored subnet (N_{CC}, M_{CCo}). A controlled colored S^3PR Petri net is called (N_{CN}, M_{CNo}). Furthermore, $(N_{CN}, M_{CNo}) = (N, M_0)||(N_{CC}, M_{CCo})$ is the integration of (N_{CC}, M_{CCo}) and (N, M_0), in which $N_{CN} = (P_A \cup \{p_{\text{combined 1}}\} \cup \{p^0\} \cup P_R, T \cup T_{CCi} \cup T_{CCo}, F \cup F_{CC}, C_R, M_{Co})$, and let its reachable graph be $R(N_{CN}, M_{CNo})$.

Theorem 1: The controlled colored S^3PR Petri net (N_{CN}, M_{CNo}) is live.

Proof: It requires to demonstrate that each transition T, T_{CCi}, T_{CCo} is live in (N_{CN}, M_{CNo}).

There is no M^*_{FBM}, since all $t_\text{h} \in T$ are live. For every $t_\text{q} \in T_{CCi}$, if for all $p_i \in {}^{\bullet}t_q$, $M_{CN}(p_i) > 0$, then, in any instance, t_q can fire in any situation because it is uncontrollable. Hence, $M_{CN}(p_{\text{combined 1}}) > 0$; for all $t_s \in T_{CCo}$, if $M_{CN}(p_{\text{combined 1}}) > 0$, then t_s can fire. Consequently, the controlled colored S^3PR Petri net (N_{CN}, M_{CNo}) is live. □

The deadlock-prohibition algorithm is developed according to the colored Petri net [5] and the method of MFFP1 [56], as shown in Algorithm 1.

Algorithm 1 Deadlock-Prohibition Procedure Based on Colored Petri Nets and an Iterative Method

Input: *An S^3PR (N, M_0)*
Output: *An S^3PR Petri net with controlled colored (N_{CN}, M_{CNo})*
Initialisation :$V_p = \emptyset$.
Step 1: *Compute M^*_{FBM}, M_{FBM}, M^*_L, M_L.*
Step 2: ***while** $M^*_{FBM} \neq \emptyset$ **do***
1. Build MFFBMP1.
2. Resolve MFFBMP1. Let β and li's be the resolution
*if $f^* \neq 0$ **then***
Let β and li's be a solution
else
Exit, if the solution does not exist.
end if
3. Design a monitor Vs and PI.
*4. $M^*_{FBM} = M^*_{FBM}$ - FI. /*FI is wrapped M^*_{FBM} V_p: $= V_p \cup \{V_S\}$ and*
end while
Step 3: *Combine monitors into a common monitor ($p_{combined1}$) all V_p and follow the following procedures:*
1. Insert output arcs of $p_{combined1}$ T_{CCo}. / Via Definition 14.*/*
2. Insert input arcs of $p_{combined1}$ T_{CCi}. / Via Definition 14.*/*
3. Assign colors C_{vsi} for all monitors P_{DC} / Via Definition 14.*/*
4. Compute with the colors marking an initial token of an incorporated monitor M_{CCo} ($p_{combined1}$) $= \sum M_{DCo}(V_S)$/ Via Definition 14.*/*
Step 4: *The combined monitor adds to the net (N, M_o)*
Step 5: *Output a controlled colored S^3PR Petri net (N_{CN}, M_{CNo})*
Step 6: ***End***

C. ROBUST CONTROL ESTABLISHED ON COLORED PETRI NETS FOR OVER UNRELIABLE RESOURCES

In Algorithm 1, it is generally assumed that the resources of a system are reliable. One common monitor is integrated after using Algorithm 1 for a system. In fact, the system resources may probably fail. Hence, a common recovery subnet is developed to model all system resource failures. The correlation between Algorithm 1 and resource failures of the controlled system is the topic of this section.

Definition 16 [6]: Let P_{UR} be an unreliable resource set, and let (N_{CN}, M_{CNo}) be a controlled colored S^3PR Petri net with $N_{CN} = (P_A \cup \{p_{\text{combined }1}\} \cup P_R \cup P^0, F \cup F_{CC}, T \cup T_{CCi} \cup T_{CCo}, M_{Co}, C_R)$. For all $r_u \in P_{UR}$, insert a common subnet for recovery to all $p_i \in P_H$, ending in a colored controlled unreliable S^3PR Petri net represented as $(N_{\text{CU}}, M_{\text{CU}o}) = (N_{\text{CN}}\ M_{CNo}) \parallel (N_{ccr}, M_{ccro})$, which is the integration of (N_{ccr}, M_{ccro}) and (N_{CN}, M_{CNo}).

Definition 17 [6]: Let $r_u \in P_R$ be an unreliable resource in N_{CN} and (N_{CN}, M_{CNo}) be a controlled S^3PR colored Petri net. A common colored recovery subnet for r_u is a Petri net $N_{ccr} = (\{p_i, p_{\text{combined }2}\}, \{t_{fj}, t_{ri}\}, F_{ccr}, C_F)$, where t_{fi} is a failure transition, $p_{\text{combined }2}$ is said to be the recovery place for all p_i, and t_{ri} is the recovery transition. $F_{ccr} = \{(p_i, t_{fi}), (t_{fi}, p_{\text{combined }2}), (p_{\text{combined }2}, t_{ri}), (t_{ri}, p_i)\}$, and when the resource is busy (its holders) or r_u is idle, an unreliable resource can fail.

Here we denote by $P_H = \{r_u\} \cup H(r_u)$ a set of places, while $H(r_u)$ is a set of holders of r_u, defined by $H(r_u) = \{p \in P_A \mid p \in {}^{\bullet\bullet}r_u \cap P_A \neq \emptyset\}$; $p_i \in P_H$. C_F is a set of colors that mapps $p_i \in P_H$ into colors $C_{ccri} \in C_F$ in which $C_F = \cup_{i \in p_i}\{C_{ccri}\}$. (N_{ccr}, M_{ccro}) is termed a colored common recovery subnet, where $M_{\text{ccro}}(p_i) \geq 0$ and $M_{\text{ccrio}}\ (p_{\text{combined }2}) = 0$.

Definition 18 [6]: Let $N_{CU} = (P_A \cup \{p_{combined1}, p_{combined2}\} \cup P_R \cup P^0, T \cup T_{CCi} \cup T_{CCo} \cup T_F \cup T_R, F \cup F_{CC} \cup F_{ccr}, \{C_R, C_F\}, M_{CUo})$ be a colored unreliable controlled S^3PR Petri net, T_F be the set of transitions of failures, and T_R be the set of recovery transitions, where $T_F = \cup_{i \in \text{N}_\text{A}} \{t_{fi}\}$ and $T_R = \cup_{i \in \text{NA}} \{t_{ri}\}$, $\text{N}_\text{A} = \{i \mid p_i \in P_H\}$, and M_{CUo} is an initial marking of N_{CU}.

Theorem 2: The net (N_{CU}, M_{CUo}) is alive.

Proof: We need to prove that all transitions in T, T_{CCi}, T_{CCo}, T_F, and T_R are live in (N_{CU}, M_{CUo}). If the failure in $r_u \in P_{UR}$ for all $t_\text{h} \in T$ does not occur with $M_{\text{CU}}\ (p_i) > 0$ for all $p_i \in {}^\bullet t_h$, then t_h can fire. For all $t_\text{q} \in T_{CCi}$, if $M_{CU}\ (p_i) > 0$ for all $p_i \in {}^\bullet t_q$, then t_q can fire. For all $t_\text{d} \in T_F$, if for all $p_i \in {}^\bullet t_d, M_{CU}\ (p_{\text{combined }2}) > 0$, then t_d can fire, leading to $M_{CU}\ (p_{\text{combined }2}) > 0$. For all $t_\text{s} \in T_{\text{CCo}}$, if $M_{CU}\ (p_{\text{combined }1}) > 0$, then t_s can fire. For all $t_\text{z} \in T_R$, if $M_{\text{CU}}\ (p_{\text{combined }2}) > 0$, then t_z can fire. Consequently, it can be said that the net (N_{CN}, M_{CNo}) is live. □

Based on Theorem 2 and Definitions 16–18 as well as the algorithm of colored Petri net, the unreliable resources are identified by the following algorithm.

Algorithm 2 Unreliable Resources Handling Based on Colored Petri Nets

---

Input: *A net* (N_{CN}, M_{CNo}) due to Algorithm 1
Output: *A net system* (N_{CU}, M_{CUo})
Initialisation : Design a $p_{combined2}$.
Step 1: ***do for*** *all* $r_u \in P_{UR}$
1. *Add* t_{fi} *a failure transition. /* Via Definition 17.*/*
2. *Assign color* C_{ccri} *for* t_{fi} *failure transition. /* Via Definition 17.*/*
3. *Add* t_{ri} *a recovery transition. /* Via Definition 17.*/*
4. *Add an arc to* t_{fi} *from* p_i. */* Via Definition 17.*/*
5. *Add an arc to* $p_{combined2}$ *from* t_{fi}. */* Via Definition 17.*/*
6. *Add an arc to* t_{ri} *from* $p_{combined2}$. */* Via Definition 17.*/*
7. *Add an arc to* p_i *from* t_{ri}. */Via Definition 17.*/*

end for
Step 2: *: Output a net* (N_{CU}, M_{CUo})
Step 3: ***End***

D. TRUST-BASED COLORED PETRI NET FOR FAULT DETECTION AND TREATMENT

Definition 19: A net system (N_{TM}, M_{TMo}) is a trust-based colored unreliable controlled S^3PR Petri net with $N_{TM} = (P_A \cup \{p_{\text{combined }1}, p_{\text{combined }2}\} \cup P_R \cup P^0, T \cup T_{CCi} \cup T_{CCo} \cup T_F \cup T_R, F \cup F_{CC} \cup F_{ccr}, \{C_R, C_F\}, M_{TMo}, \psi, \eta)$, and $R(N_{TM}, M_{TMo})$ being its reachability graph if

1) η is the weight of an arc, which indicates the significance of input arcs into a transition or probability. If there is an arc(p, t), then $\eta(p, t) = c$ denotes that there is a probability of $\eta(p, t)$ supporting the token from p entry t. A new capacity will be h^*c if it has a capacity h.
2) ψ is the threshold of tokens in a place p, $\psi : p \to R$, where R is an actual kind of data. $\Psi(p) = r_1$ denotes the fact that if the number of tokens in p is equal to or greater than r_1, p is able to reach a new position.

In an unreliable controlled S^3PR Petri net, a variety of factors can influence its trust. Therefore, to model a trust-based colored unreliable S^3PR Petri net for fault detection and treatment, the value of each factor type can be represented by a non-negative value of a real number. In order to assemble a novel trust value, the evaluation process will consume the factors. E^{in} is used to describe the input factors ingested, and E^{ou} is used to describe the aggregation trust value. $E^{in}(x_i)$ and $E^{ou}(y_i)$ correspond to the input place and the output place respectively in a fault detection assessment process. In a trust-based colored unreliable controlled S^3PR Petri net, the laws for firing transitions are presented as follows.

Definition 20: Let (N_{TM}, M_{TMo}) be a trust-based colored unreliable controlled S^3PR Petri net with $N_{TM} = (P_A \cup \{p_{combined1}, p_{combined2}\} \cup P_R \cup P^0, T \cup T_{CCi} \cup T_{CCo} \cup T_F \cup T_R,$

$F \cup F_{CC} \cup F_{ccr}, \{C_R, C_F\}, \eta, \psi, M_{TMo})$. A fault detection transition t_{di} can fire at a marking if:

1) $t_{di} \in T_k$ is a fault detection transition,
2) $E^{in}(x_i) > 0$,
3) $E^{\text{out}}(y_i) \geq \Psi(y_i)$, $E^{\text{out}}(y_i) = \sum_{i=1}^{nf} \sum_{j=1}^{|T_k|} \eta_i * E^{in}{}_i(x_i)$, where nf is the number of input factors,
4) for all $p \in {}^\bullet t_{di}, M'(p) = M(p) - W(p, t_{di}) + W(t_{di}, p)$.

where $E^{in}(x_i)$ is the flow value of the token in the input place of the fault detection transition. In the output place of the fault detection transition, $E^{\text{out}}(y_i)$ is the current value of the token, and $\Psi(y_i)$ is the threshold for entering y_i.

Algorithm 3 A Trust-Based Colored Unreliable Controlled S^3PR Petri Net Construction

Input: An unreliable controlled S^3PR with $N_{CU} = (P_A \cup \{p_{combined1}, p_{combined2}\} \cup P_R \cup P^0, T \cup T_{CCi} \cup T_{CCo} \cup T_F \cup T_R, F \cup F_{CC} \cup F_{ccr}, \{C_R, C_F\}, M_{CUo})$, and $\eta \rightarrow [0, 1]$.

Output: A trust-based colored unreliable controlled S^3PR Petri net $N_{TM} = (P_A \cup \{p_{combined1}, p_{combined2}\} \cup P_R \cup P^0, T \cup T_{CCi} \cup T_{CCo} \cup T_F \cup T_R, F \cup F_{CC} \cup F_{ccr}, \{C_R, C_F\}, \eta, \psi, M_{TMo})$ and fault type;

1. *Design all places of input factors*
2. *Design all places of a trust output*
3. *Design all fault detection transitions*
4. *Design all fault treatment transitions*
5. ***for*** all $1 \leq i \leq |T_F|$ ***do***
6. ***if*** t_{fi} fires ***then***
7. ***for*** all $1 \leq j \leq nf$ ***do***
8. ***for*** all $1 \leq l \leq |T_k|$ ***do***
9. calculate the $E^{out}(y_l)$;
10. ***end for***
11. Calculate the winner y_l;
12. ***end for***
13. ***else if***
14. *Break;*
15. ***end if***
16. ***end for***

If the $E^{\text{out}}(y_i)$ value meets the threshold for entering y_i, then it is called the winner and its output value y_i is asserted as 1; otherwise, another output value is asserted as 0, expressed by

$$y_i = \begin{bmatrix} 1, & E^{\text{out}}(y_{i+1}) < E^{\text{out}}(y_i) \\ 0, & \text{otherwise} \end{bmatrix} \tag{3}$$

The trust-based colored unreliable controlled S^3PR Petri net is stated in Algorithm 3.

III. NUMERICAL EXAMPLE

Consider the AMS outlined in Fig. 1(a) to illustrate the proposed methodology. In Kaid *et al.* [5] and Chen *et al.* [56], the S^3PR Petri net model is provided. The system consists of two M1 and M2 machines. All machines handle a single piece at a time and the robot R1 carries the single piece at a time. There are two unloading/loading buffers. Furthermore, two types of pieces A and B in the system are processed. The processing routes for two types of products are depicted in Fig. 1(b). Fig. 2 illustrates the Petri net model of this AMS example. It is composed of 8 transitions and 11 places. The descriptions of places as configured partitions are as follows: $P_R = \{p_9, p_{10}, p_{11}\}$, $P_A = \{p_2, p_3, \ldots, p_7\}$, and $P^0 = \{p_1, p_8\}$. There are twenty reachable markings in the Petri net model. We have $M_L^* = \{p_5 + p_6 + p_7, p_2 + p_3 + p_4\}$ and $M_{\text{FBM}}^* = \{p_3 + p_5, p_2 + p_5, p_2 + p_6\}$.

By considering the implementation of steps 1 and 2 of Algorithm 1, suppose that I_l is the invariant place to be computed and Equation (2) for the two legal markings in M_L^* is satisfied, i.e., $l_2 \cdot 1 + l_3 .1 + l_4 .1 \leq \beta$ and $l_5 .1 + l_6 .1 + l_7 .1 \leq \beta$. Therefore, we have two constraints:

$$\begin{aligned} &l_2 + l_3 + l_4 \leq \beta \text{ and} \\ &l_5 + l_6 + l_7 \leq \beta \end{aligned}$$

To represent that I_1 prohibits FBM1s $\text{FBM}_1 = p_3 + p_5$, $\text{FBM}_2 = p_2 + p_5$, and $\text{FBM}_3 = p_2 + p_6$, respectively, we need three variables f_1, f_2, and f_3. Then, three constraints are derived:

$$\begin{aligned} &l_3 + l_5 \geq \beta - Q \cdot (1 - f_1) + 1 \\ &l_2 + l_5 \geq \beta - Q \cdot (1 - f_2) + 1, \text{ and} \\ &l_2 + l_6 \geq \beta - Q \cdot (1 - f_3) + 1 \end{aligned}$$

Consequently, MFFP1 is described as MFFBMP1:

$$\begin{aligned} &\max f = f_1 + f_2 + f_3 \\ &\text{subject to } l_5 + l_6 + l_7 \leq \beta \\ &\quad l_2 + l_3 + l_4 \leq \beta \\ &\quad l_2 + l_5 \geq \beta - Q \cdot (1 - f_1) + 1 \\ &\quad l_3 + l_5 \geq \beta - Q \cdot (1 - f_2) + 1 \\ &\quad l_2 + l_6 \geq \beta - Q \cdot (1 - f_3) + 1 \\ &\quad \beta \in \{1, 2, 3, \ldots\} \\ &\quad l_i \in \{0, 1, 2, 3, \ldots\}, \quad \forall i \in \{2, 3, 4, 5, 6, 7\} \\ &\quad f_1, f_2, f_3 \in \{0, 1\} \end{aligned}$$

The MFFBMP1 is solved and an optimal solution is obtained: $l_6 = 1, l_5 = 1, l_2 = 2, \beta = 2, f_2 = 1, f_3 = 1$, and all the other variables are zero. A control place V_{S1} for $I_I : 2\mu_2 + \mu_5 + \mu_6 + \mu_{V_{S1}} = 2$ is therefore developed. As a consequence, I_1 forbids FBM2 and FBM3, and we have ${}^\bullet V_{S1} = \{2t_2, t_7\}$, $V_{S1}{}^\bullet = \{2t_1, t_5\}$, and $M_{CNo}(V_{S1}) = \beta = 2$. Hence, we have $F_{I1} = \{\text{FBM3}, \text{FBM2}\}$, $M_{\text{FBM}}^* := M_{\text{FBM}}^* - F_{I1}$; thus, $M_{\text{FBM}}^* = \text{FBM1} = \{p_3 + p_5\}$.

Let the place invariant I_2 be calculated and fulfill (2) for the two legal markings in M_L^* at the second reiteration, i.e., $l_5 .1 + l_6 .1 + l_7 .1 \leq \beta$ and $l_2 .1 + l_3 .1 + l_4 .1 \leq \beta$. Hence, we have two restrictions:

$$\begin{aligned} &l_5 + l_6 + l_7 \leq \beta \text{ and} \\ &l_2 + l_3 + l_4 \leq \beta \end{aligned}$$

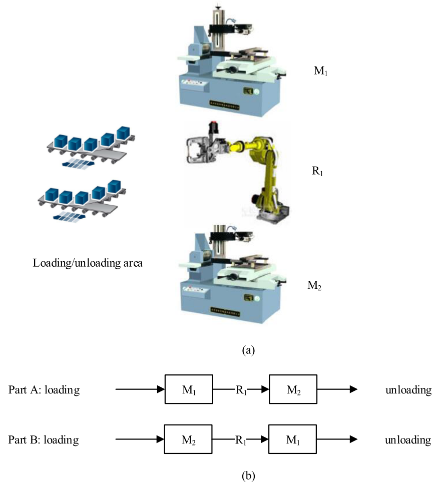


FIGURE 1. **(a) Example of AMS and (b) Sequence of operation.**

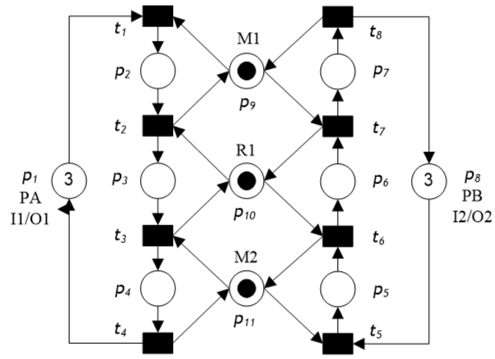


FIGURE 2. **Petri net S^3PR model of the AMS.**

We have one variable f_1 to show whether FBM1 $FBM_1 = p_3 + p_5$ is prohibited by I_2. Thus, we have one constraint:

$$l_3 + l_5 \geq \beta - Q \cdot (1 - f_1) + 1$$

Consequently, a novel MFFP1 is depicted below.
MFFBMP1:

$$\begin{aligned} &\max f = f_1 \\ &\text{subject to } l_5 + l_6 + l_7 \leq \beta \\ &\qquad l_2 + l_3 + l_4 \leq \beta \\ &\qquad l_3 + l_5 \geq \beta - Q \cdot (1 - f_1) + 1 \\ &\qquad \beta \in \{1, 2, 3, \ldots\} \\ &\qquad l_i \in \{0, 1, 2, 3, \ldots\}, \quad \forall i \in \{2, 3, 4, 5, 6, 7\} \\ &\qquad f_1 \in \{0, 1\}. \end{aligned}$$

The new MFFBMP1 is now solved, with $l_3 = 1$, $l_5 = 1$, $\beta = 1$, $f_1 = 1$, being the optimal solution, and all the other variables being zero. After that, a control place V_{S2} for I_2: $\mu_3 + \mu_5 + \mu_{VS2} = 1$ is developed. Consequently, I_2 forbids FBM1; we have ${}^\bullet V_{S2} = \{t_3, t_6\}$, $V_{S2}^\bullet = \{t_2, t_5\}$, and $M_{CNo}(V_{S2}) = \beta = 1$. Therefore, we have $F_{I2} = \{\text{FBM1}\}$, $M^*_{\text{FBM}} := M^*_{\text{FBM}} - \text{F}_{\text{I2}}$. Thus, $M^*_{\text{FBM}} = \emptyset$, and Steps 1 and 2 are finished. Therefore, for this net, two control places are computed in total.

Consider the implementation of Step 3 in Algorithm 1. The two control places that are obtained are combined into $p_{\text{combined 1}}$. For $p_{\text{combined 1}}$, the output arcs T_{CCo} are represented as $T_{CCo} = \{2t_1, t_2, 2\,t_5\}$. For $p_{\text{combined 1}}$, the input arcs T_{CCi} are represented as $T_{CCi} = \{2t_2, t_3, t_6, t_7\}$. In addition, $M_{CNo}\,(p_{\text{combined 1}}) = \sum M_{DCo}\,(V_S) = M_{DCo}\,(V_{S1}) + M_{DCo}\,(V_{S2}) = 2 + 1 = 3$. Therefore, $C_R = \{C_{vs1}, C_{vs2}\}$ are two types of colors. Hence three colored tokens are the total: one and two token for C_{vs2}, C_{vs1} colors, respectively. Fig. 3 shows the developed single controller of the controlled colored S^3PR Petri net model displayed in Fig. 2, which is obtained via Algorithm 1. If transition t_1 fires, as shown in Fig. 3, it selects from the common place $p_{\text{combined 1}}$ two tokens with color C_{vs1}, from the input place p_1 one token, from p_9 one token, and drops them into p_2. If transition t_2 fires, it selects one token from the common place $p_{\text{combined 1}}$ with the color C_{vs2}, one token of p_{10}, from p_2 one token, and drops it into p_3. Moreover, if transition t_5 fires, it selects from the common place $p_{\text{combined 1}}$ one token with the color C_{vs1}, and also one token with the color C_{vs2} from the same place, from the place of input p_8 one token, from the p_{11} one token, and deposits them into p_5. It creates on the tokens from p_{10} and p_2 two colors C_{vs1} if transition t_2 fires, and drops them in the common place $p_{\text{combined 1}}$. Moreover, if transition t_3 fires, from p_3 and p_{11} on the tokens it creates one color C_{vs2} and drops it into the common place $p_{\text{combined 1}}$. It creates from p_5 and p_{10} one color C_{vs2} on the tokens if transition t_6 fires, and places them inside the common place $p_{\text{combined 1}}$. Finally, if transition t_7 fires, it creates one C_{vs1} color on the p_9 and p_6 tokens and deposits them in the common place $p_{\text{combined 1}}$.

Consider an S^3PR net shown in Fig. 3 to demonstrate the efficacy of Algorithm 2. We have $P_{UR} = \{p_9, p_{10}, p_{11}\}$, $H(p_{11}) = \{p_4, p_5\}$, $H(p_{10}) = \{p_3, p_6\}$, and $H(p_9) = \{p_2, p_7\}$. A common recovery subnet for p_{11}, p_{10}, and p_9 is introduced in an unreliable S^3PR net model, as shown in Fig. 4. Hence, $N_A = \{2, 3, 4, 5, 6, 7, 9, 10, 11\}$ is the index set, $T_F = \{t_{f2}, t_{f3}, t_{f4}, t_{f5}, t_{f6}, t_{f7}, t_{f9}, t_{f10}, t_{f11}\}$, $T_R = \{t_{r2}, t_{r3}, t_{r4}, t_{r5}, t_{r6}, t_{r7}, t_{r9}, t_{r10}, t_{r11}\}$, and $C_R = \{C_{ccr2}, C_{ccr3}, C_{ccr4}, C_{ccr5}, C_{ccr6}, C_{ccr7}, C_{ccr9}, C_{ccr10}, C_{ccr11}\}$.

When machine 1 fails in busy state p_7 or p_2 has a token, i.e., t_{f7} or t_{f2} fires, to release a token of color C_{ccr7} or C_{ccr2} from p_7 or p_2 respectively and deposits them into $p_{combined2}$. When machine 1 fails and there is a token in p_9, i.e., t_{f9} fires, it creates from p_9 a token of the color C_{ccr9} and drops it into $p_{combined2}$. Machine 1, which has a token in p_9, is then repaired by moving the color token to p_9, p_7 or p_2 from $p_{combined2}$ by firing t_{r9}, t_{r7} or t_{r2}. If the transition t_{r9}, t_{r7} or t_{r2} fires, it selects from $p_{combined2}$ the token with color C_{ccr9}, C_{ccr7}, or C_{ccr2} and drops it into p_9, p_7 or p_2, indicating the accomplishment of the recovery process of machine 1 p_9.

If robot 1 fails in the busy state p_6 or p_3 has a token, i.e., t_{f6} or t_{f3} fires, it creates from p_6 a token C_{ccr6} or from p_3 a token C_{ccr3}, and drops them into $p_{combined2}$. When robot 1 fails in

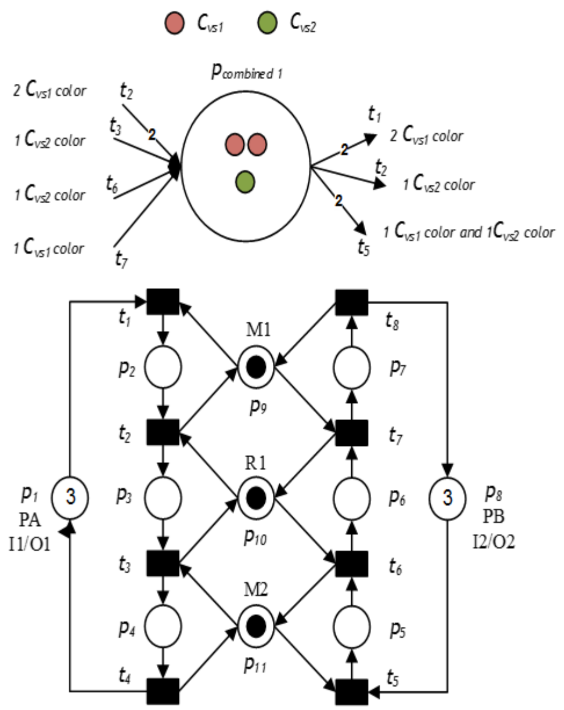


FIGURE 3. Controlled model of colored S^3PR Petri net by Algorithm 1.

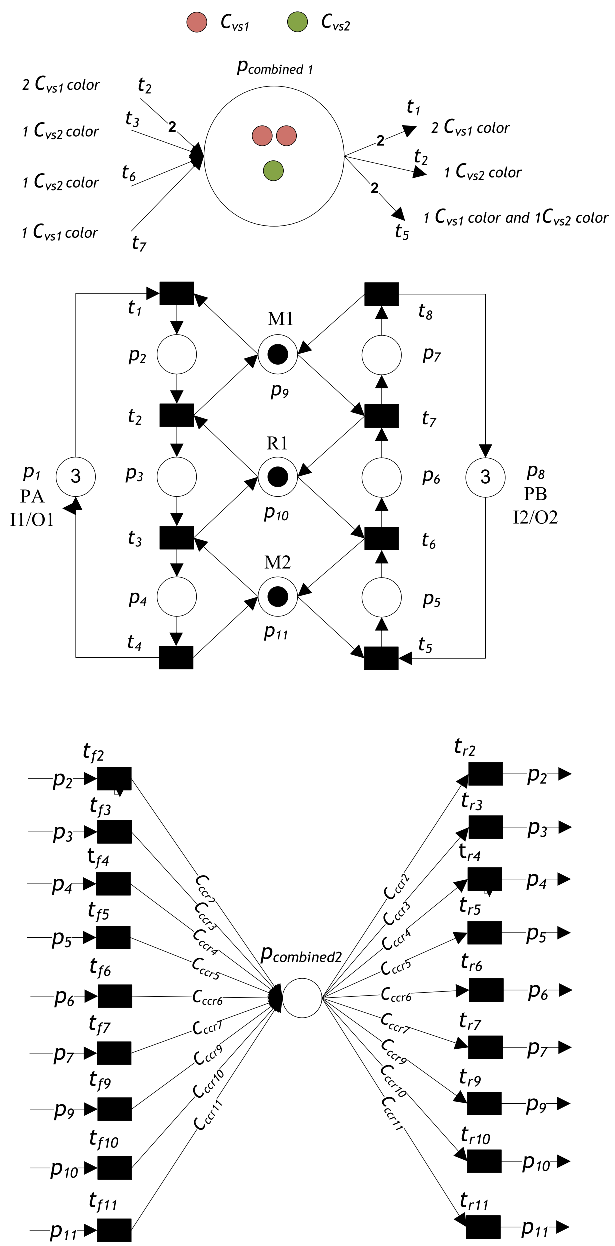


FIGURE 4. Using Algorithm 2 colored, unreliable, controlled Petri net model.

idle status or p_{10} has a token, i.e., t_{f10} fires, it creates from p_{10} a token C_{ccr10}, and drops it into $p_{combined2}$. Robot 1 p_{10} is then repaired, and by firing t_{r10}, t_{r6}, or t_{r3}, the token with color in $p_{combined2}$ moves into p_{10}, p_6, or p_3. If the transition t_{r10}, t_{r6}, or t_{r3} fires, it selects the token C_{ccr10}, C_{ccr6}, or C_{ccr3} from $p_{combined2}$ and drops the same into p_{10}, p_6, or p_3, indicating that the recovery process of robot p_{10} is accomplished.

Ultimately, if machine 2 fails in the busy state p_5 or p_4 has a token, i.e., t_{f5} or t_{f4} fires, it creates from p_5 or p_4 a token C_{ccr5} or a token C_{ccr4}, correspondingly, and drops them into the $p_{combined2}$. When machine 2 fails in idle position or p_{11} has a token, i.e., t_{f11} fires, it creates from p_{11} a token C_{ccr11}, and drops it into $p_{combined2}$. Machine 2 is then repaired, and by firing t_{r5}, t_{r4}, or t_{r11}, the token with color in $p_{combined2}$ moves into p_5 or p_4, or p_{11}. If the transition t_{r11}, t_{r4}, or t_{r5} fires, it selects from $p_{combined2}$ the token C_{ccr11}, C_{ccr4}, or C_{ccr5} and drops it into p_{11}, p_4, or p_5, implying the accomplishment of the recovery process of machine 2 p_{11}.

Finally, let us consider the unreliable S^3PR net model presented in Fig. 6 to demonstrate the applicability of Algorithm 3. In order to detect the failure that results in different faults, Algorithm 3 is used. The necessary data are obtained, utilizing the unreliable S^3PR net model presented in Fig. 6. The six input variables are presented as follows:

x_1: the mechanical vibration accelerator-meter caused by the force-oscillation cutting;

x_2: the current sensor that detects alterations in the current drained via the electrical motor;

x_3: tool torsion that is disclosed via the strain gauges;

x_4: coolant sensor that observes the coolant level;

x_5: sensor in the grip of the robot, which detects collapse or collision effects from sensor signals differences with kinematics/ dynamic effects; and

x_6: acoustic emission sensor detected for tool break-detection acoustic effects of stress waves.

One output is expressed as a fault at a time, such as tool wear failure y_1 represented as [1 0 0 0 0], tool break failure y_2 represented as [0 1 0 0 0], coolant failure y_3 represented as [0 0 1 0 0], programming errors failure y_4 represented as [0 0 0 1 0], and robot's wrist or grip failure y_5 represented as [0 0 0 0 1].

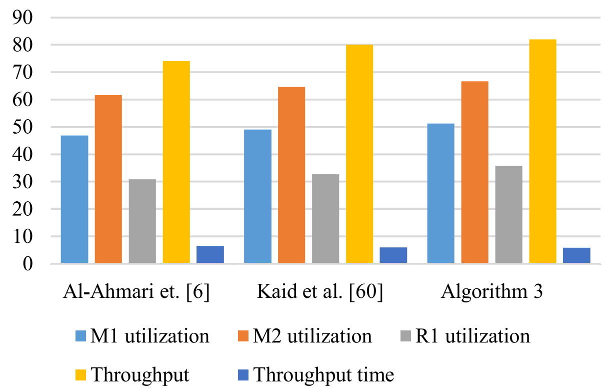


FIGURE 5. Comparison of Kaid *et al.* [58] and Al-Ahmari *et al.* [6] with Algorithm 3.

MATLAB R2015a is used to implement the proposed trust colored controlled unreliable S^3PR Petri net model. A PC with 16 GB RAM, Intel(R) Core (TM) i7-4702MQ CPU @ 2.20 GHz, 64-bit operational system was employed and running on Windows 10.

The sensors in the input factors of the trust model shown in Fig. 6 are acquired by a collection signal system for machine tool states (abnormal or normal). The "uniform" or "random" signal peaks may be created via the machine tool. Signals with random peaks indicate tool-break. The machining parameters programmed in the wrong way or wear small tools refer to signals with uniform peaks. Sensors in the robot's wrist or grip can monitor the loading or unloading process. The collision or crashing influences can be detected from the disparity between the kinematics/dynamics effects and the sensor signals. In the output of the trust model shown in Fig. 6, programming errors that cause failures resulting from tool-break produce random peaks, and generate

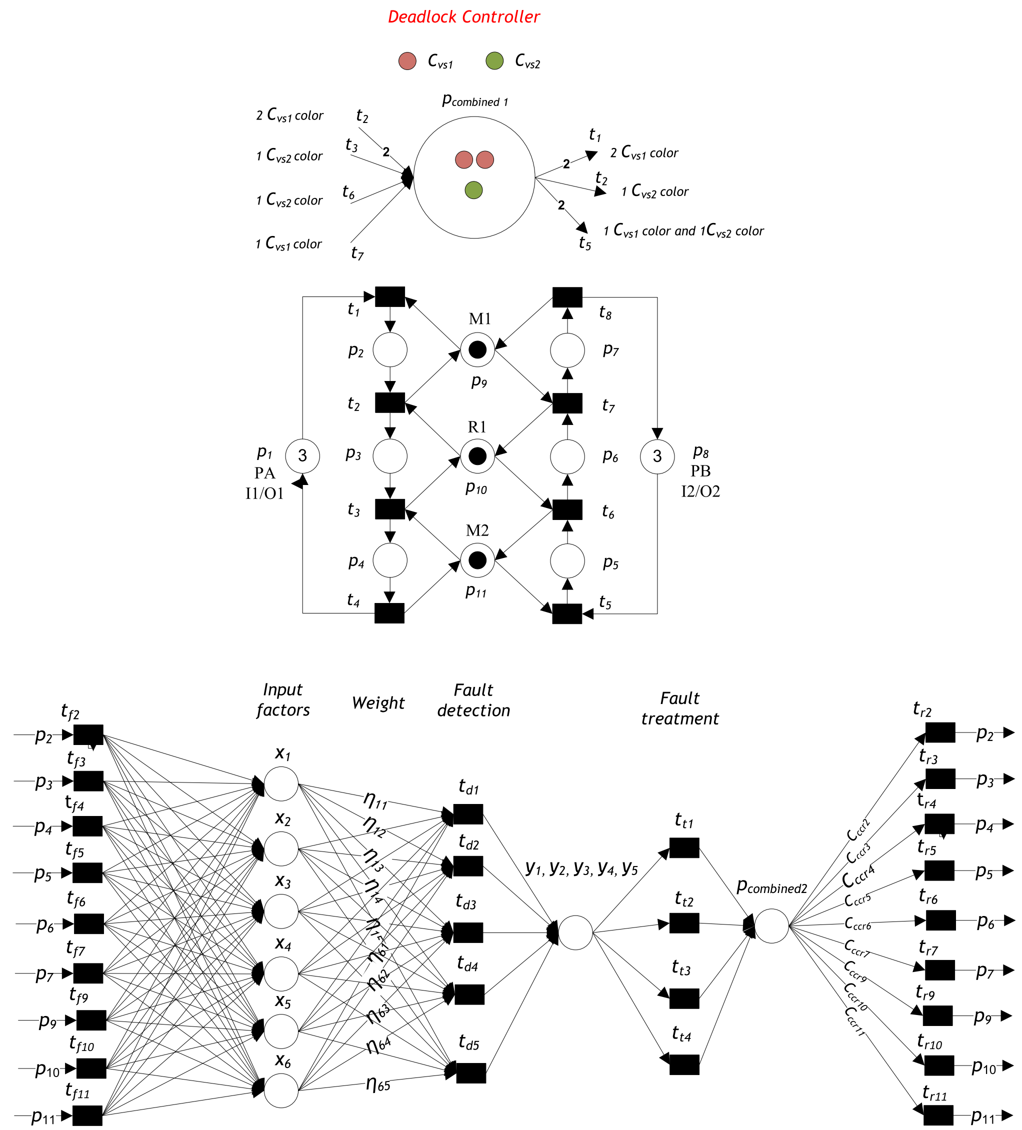


FIGURE 6. Trust colored controlled unreliable S^3PR Petri net model utilising Algorithm 3.

TABLE 1. Time performance comparison with [6], [58] and Algorithm 3.

Parameter	Al-Ahmari et al. [6]	Kaid et al. [58]	Algorithm. 3
Utilization,(%)in M1	46.875	49.0909	51.2354
Utilization,(%) in M2	61.6666	64.6465	66.6335
Utilization,(%) in R1	30.8334	32.7273	35.8235
Throughput (parts)	74	80	82
Throughput time (min/ part)	6.4865	6.0000	5.8536

uniform peaks from non-appropriate tools or wrong machine parameters. Variations among sensor signals and dynamics/kinematics effects are recognized because of failures due to the robot's wrist or grip. Defaults caused via the lack of coolant lead to coolant failures, and the failure of the tool-wear is recognized through motor oscillations and uniform peaks.

The proposed treatments utilised to recover a failed robot or machine are parameter change, tool change, the intervention of a human operator, and coolant change, which are represented by t_{t1}, t_{t2}, t_{t3}, and t_{t4}, respectively. If a wear failure is the output trust model, the change in the parameter is the proposed treatment. If a tool break failure is the output trust model, the proposed handling is a tool alteration. When coolant failure is the output trust model, the suggested remedy is a coolant change. Moreover, if a machining parameter failure or a programming error is the output trust model, the suggested remedy is a parameter change. Finally, when a robot's wrist or grip failure is the output trust model, the proposed treatment will require intervention of a human operator.

Algorithm 3 is finally compared with Kaid *et al.* [58] and Al-Ahmari *et al.* [6]. The performance time criteria are shown in Fig. 5 and Table 1. Altogether, the performance due to Algorithm 3 is better than that in Kaid *et al.* [58] and Al-Ahmari *et al.* [6] in relation to resource utilization. Furthermore, the suggested model could provide high throughput with less throughput time than the latter two.

IV. CONCLUSION

This study offers a robust control strategy for failure detection and handling of failures of unreliable resource in an AMS, falling into the domain of discrete event systems [**?**], [63]–[66]. The proposed strategy consists of three steps. The controlled system is established in the first step using MFFBMP1 that is an integer linear programming problem, without considering resource failures. A treatment of deadlock control problems caused by resource failures in an AMS is provided in the second step. Using a Petri net model of AMSs, recovery subnets for all resource failures are established in colored Petri nets. The recovery subnets are added to the system to guarantee the reliability of the system in the first step. The third step is a new hybrid approach, which combines a trust-based model together with colored Petri nets to detect and treat faults in the second step. In comparison with the existing approaches, the strategy presented in this paper is effective, carrying the advantages of high resource utilization, which is validated with the GPenSIM tool.

The main benefits of the reported strategy are as follows: (1) The proposed supervisor has an uncomplicated and straightforward structure, and the strategy is considered to be more robust compared with the one given by Kaid *et al.* [5], Ezpeleta *et al.* [59], Al-Ahmari *et al.* [6], and Chen *et al.* [57]; (2) it combines three kinds of procedures: a deadlock-free system without considering a resource failure, fault detection, and fault treatment; (3) it takes into consideration not only the solution of AMSs deadlock problems, but also failure detection and treatment. The method proposed in this paper will be applied to multi-agent manufacturing systems [60].

REFERENCES

[1] L. Yang, Z. Yu, M. A. El-Meligy, A. M. El-Sherbeeny, and N. Wu, "On multiplexity-aware influence spread in social networks," *IEEE Access*, vol. 8, pp. 106705–106713, 2020, doi: 10.1109/ACCESS.2020.2999312.

[2] Z. Gao, C. Cecati, and S. X. Ding, "A survey of fault diagnosis and fault-tolerant techniques—Part I: Fault diagnosis with model-based and signal-based approaches," *IEEE Trans. Ind. Electron*, vol. 62, no. 6, pp. 3757–3767, Jun. 2015.

[3] L. A. M. Riascos, L. A. Moscato, and P. E. Miyagi, "Detection and treatment of faults in manufacturing systems based on Petri nets," *J. Brazilian Soc. Mech. Sci. Eng.*, vol. 26, no. 3, pp. 280–289, Sep. 2004.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] S. Zhou, Z. Yu, E. S. A. Nasr, H. A. Mahmoud, E. M. Awwad, and N. Wu, "Homomorphic encryption of supervisory control systems using automata," *IEEE Access*, vol. 8, pp. 147185–147198, 2020, doi: 10.1109/ACCESS.2020.3014217.

[9] 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.

[10] 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.

[11] 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. 137–142, Feb. 2003.

[12] 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, Apr. 2011.

[13] 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.

[14] 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.

[15] 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.
[16] 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.
[17] 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.
[18] 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.
[19] 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)*, Mar. 2015, pp. 1–10.
[20] 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.
[21] M. Uzam, Z. W. Li, and U. S. Abubakar, "Think globally a ctlocally approach for the synthesis of a liveness-enforcing supervisor of FMSs based on Petri nets," *Int. J. Prod. Res.*, vol. 54, no. 15, pp. 4634–4657, 2016.
[22] 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.
[23] 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.
[24] 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.
[25] 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.
[26] 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.
[27] S. 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, May 2008.
[28] F.-S. Hsieh, "Robustness analysis of Petri nets for assembly/disassembly processes with unreliable resources," *Automatica*, vol. 42, no. 7, pp. 1159–1166, Jul. 2006.
[29] 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.
[30] 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.
[31] 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.
[32] U. S. Abubakar, G. Liu, and M. Uzam, "Petri net-based robust supervisory control of automated manufacturing systems with multiple unreliable resources," *IEEE Access*, vol. 9, pp. 100264–100278, 2021.
[33] X. Li, G. Liu, Z. Li, N. Wu, and K. Barkaoui, "Elementary siphon-based robust control for automated manufacturing systems with multiple unreliable resources," *IEEE Access*, vol. 7, pp. 21006–21019, 2019.
[34] Z. Li and M. Zhou, *Deadlock Resolution in Automated Manufacturing Systems: A Novel Petri Net Approach*. London, U.K.: Springer, 2009.
[35] 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, Jan. 2020.
[36] Z. Li and M. Zhao, "On controllability of dependent siphons for deadlock prevention in generalized Petri nets," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 2, pp. 369–384, Mar. 2008.
[37] Z. W. Li and M. C. Zhou, "Control of elementary and dependent siphons in Petri nets and their application," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 38, no. 1, pp. 133–148, Jan. 2008.
[38] 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.
[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] 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., Model., Manage. Control, IFAC Workshop*, Prague, Czech Republic, Aug. 2001, p. 83.
[41] 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.
[42] 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.
[43] H. Asada and J.-J. Slotine, *Robot Analysis and Control*. Hoboken, NJ, USA: Wiley, 1986.
[44] L. A. M. Riascos and P. E. Miyagi, "Supervisor system for detection and treatment of failures in balanced automation systems using Petri nets," in *Proc. IEEE Int. Syst, Man, Cybern. Tucson*, Oct. 2001, pp. 2528–2533.
[45] M. Hanna, "Modeling product quality in a machining center using fuzzy Petri nets with neural networks," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 1999, pp. 1502–1507.
[46] 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," in *IEEE Trans. Ind. Electron.*, vol. 47, no. 5, pp. 1021–1030, Oct. 2000.
[47] I. Djelloul, Z. Sari, and I. dit Bouran Sidibe, "Fault diagnosis of manufacturing systems using data mining techniques," in *Proc. 5th Int. Conf. Control, Decis. Inf. Technol. (CoDIT)*, Apr. 2018, pp. 198–203.
[48] D. Lefebvre and E. Aguayo-Lara, "A discussion on fault detection for a class of hybrid Petri nets," *IFAC-PapersOnLine*, vol. 50, no. 1, pp. 6837–6842, Jul. 2017.
[49] Z. Zhang, G. Liu, K. Barkaoui, and Z. Li, "Adaptive deadlock control for a class of Petri nets with unreliable resources," *IEEE Trans. Syst., Man, Cybern. Syst.*, early access, Mar. 12, 2021, doi: 10.1109/TSMC.2021.3062469.
[50] Y. Feng, K. Xing, M. Zhou, H. Chen, and F. Tian, "Polynomial-complexity robust deadlock controllers for a class of automated manufacturing systems with unreliable resources using Petri nets," *Inf. Sci.*, vol. 533, pp. 181–199, Sep. 2020.
[51] N. Du and H. Hu, "Robust deadlock detection and control of automated manufacturing systems with multiple unreliable resources using Petri nets," *IEEE Trans. Autom. Sci. Eng.*, vol. 18, no. 4, pp. 1790–1802, Oct. 2021, doi: 10.1109/TASE.2020.3019684.
[52] 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.
[53] N. Wang, J. Wang, and X. Chen, "A trust-based formal model for fault detection in wireless sensor networks," *Sensors*, vol. 19, no. 8, p. 1916, Apr. 2019.
[54] P. Lory, "A coloured Petri net trust model," in *Proc. 14th Int. Workshop Database Expert Syst. Appl.*, Prague, Czech Republic, 2003, pp. 415–419.
[55] T. Murata, "Petri nets: Properties, analysis and applications," *Proc. IEEE*, vol. 77, no. 4, pp. 541–580, Apr. 1989.
[56] Y. Chen, Z. Li, and M. Zhou, "Behaviorally optimal and structurally simple liveness-enforcing supervisors of flexible manufacturing systems," *IEEE Trans. Syst., Man, Cybern. A, Syst., Humans*, vol. 42, no. 3, pp. 615–629, May 2012.
[57] Y. Chen and Z. Li, *Optimal Supervisory Control of Automated Manufacturing Systems*. Boca Raton, FL, USA: CRC Press, 2013.
[58] 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.
[59] J. Ezpeleta, J. M. Colom, and J. Martínez, "A Petri net based deadlock prevention policy for flexible manufacturing systems," *IEEE Trans. Robot. Autom.*, vol. 11, no. 2, pp. 173–184, Apr. 1995.
[60] X. Li, Z. Yu, Z. Li, and N. Wu, "Group consensus via pinning control for a class of heterogeneous multi-agent systems with input constraints," *Inf. Sci.*, vol. 542, pp. 247–262, Jan. 2021, doi: 10.1016/j.ins.2020.05.085.

[61] M. Abdul-Hussin, ''Design of a Petri net based deadlock prevention policy supervisor for S3PR,'' in *Proc. 6th Int. Conf. Intell. Syst., Model. Simul.*, Feb. 2015, pp. 46–52, doi: 10.1109/ISMS.2015.54.

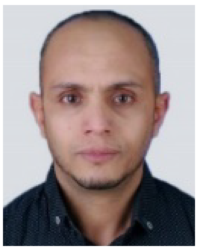
[62] Z. Yu, S. Lu, D. Wang, and Z. Li, ''Modeling and analysis of rumor propagation in social networks,'' *Inf. Sci.*, vol. 580, pp. 857–873, Nov. 2021, doi: 10.1016/j.ins.2021.09.012.

[63] Z. Ma, Z. Li, and A. Giua, ''Characterization of admissible marking sets in Petri nets with conflicts and synchronizations,'' *IEEE Trans. Autom. Control*, vol. 62, no. 3, pp. 1329–1341, Mar. 2016.

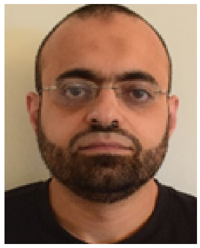
[64] H. Grichi, O. Mosbahi, M. Khalgui, and Z. Li, ''RWiN: New methodology for the development of reconfigurable WSN,'' *IEEE Trans. Autom. Sci. Eng.*, vol. 14, no. 1, pp. 109–125, Jan. 2016.

[65] M. Khalgui, O. Mosbahi, Z. Li, and H.-M. Hanisch, ''Reconfigurable multiagent embedded control systems: From modeling to implementation,'' *IEEE Trans. Comput.*, vol. 60, no. 4, pp. 538–551, Apr. 2010.

[66] G. H. Zhu, Z. W. Li, and N. Q. Wu, ''Model-based fault identification of discrete event systems using partially observed Petri nets,'' *Automatica*, vol. 96, pp. 201–212, Oct. 2018.



EBRAHIM ALI ALZALAB received the B.S. degree in electric and computer engineering and the M.S. degree in electrical engineering and control from Karary University, Khartoum, Sudan, in 2008 and 2016, respectively. He is currently pursuing the Ph.D. degree in control theory and control engineering with the School of Electro-Mechanical Engineering, Xidian University, Xi'an, China. His research interests include Petri net theory and applications and robust supervisory control of automated manufacturing systems.



AHMED M. EL-SHERBEENY received the master's and Ph.D. degrees in mechanical engineering from West Virginia University (WVU), in 2001 and 2006, respectively. He was a Graduate Teaching Assistant and a Research Assistant with WVU. Since 2010, he has been an Assistant Professor with the Industrial Engineering Department and the former Head of the Alumni and Employment Unit, College of Engineering, King Saud University, from 2013 to 2018. His research interests include human factors engineering, manufacturing engineering, and engineering education.

MOHAMMED A. EL-MELIGY received the B.Sc. degree in information technology from Menoufia University, Egypt, in 2005. He has been a Software Engineer with King Saud University, Riyadh, Saudi Arabia, since 2009. His research interests include Petri nets, supervisory control of discrete event systems, database software, and network administration.

HAFIZ TAYYAB RAUF received the bachelor's and master's degrees in computer science from the University of Gujrat, Pakistan. He is currently working as a Research Assistant with the Department of Computer Science, Faculty of Engineering and Informatics, University of Bradford, U.K. He is also working as Visiting Researcher at the Centre for Smart Systems, AI and Cybersecurity, Staffordshire University, Stoke-on-Trent, U.K. He is the author of more than 40 research articles published in reputed journals. His research interests include evolutionary computing, swarm intelligence, neural networks, image processing, computer vision, and machine learning.