DEVELOPING A SUPERVISORY TECHNIQUE FOR ONLINE FAULT ISOLATION AND COORDINATION TASKS

تطوير تقنية إشراف لكشف الأعطال ولمهام التنسيق أثناء التشغيل

Hamdi A. Awad[1], Mostafa M. Gomaa[2], Ahmed R. Anwar[3]

[1] Assoc. Prof., dept. of industrial Electronics and Control Eng., Fac. of Electronic Eng., Minufiya Univ., Minuf. (awadhaa@yahoo.co.uk)
[2] Assoc. Prof., dept. of Computer and Systems Engineering, Faculty of Engineering, Ain Shams University, Cairo. (m_gomaa_eg@yahoo.com)
[3] Ass. Lecturer, dept. of industrial Electronics and Control Eng.,Fac. of Electronic Eng., Minufiya Univ., Minuf.(rag_gwi@yahoo.com)

ملخص البحث:- التحكم فى النظم الآلية المعقدة يتطلب نظام إشرافى والذى يحتوى على إمكانات تنسيق وأيضاً كشف وتشخيص الأعطال. الهدف من نظام التنسيق هو إجبار النظام الآلى لكى يتصرف بشكل أمن ومسموح به، بينما الهدف من نظام كشف وتشخيص الأعطال هو كشف وتحديد الأعطال بناء على ملاحظة الحالة الموجود عليها النظام الآلى. هذا البحث يقترح تقنية تحكم إشرافى جديدة لغرض كشف وتحديد الأعطال وأيضاً التنسيق أثناء التشغيل إعتماداً على نماذج شبكات بترى. هذه التقنية المقترحة تحتوى على مُشرف ومشخّص. الأخير يقوم بتقييم الحالات التى بها أعطال بدون الحاجة لفحص الخصائص التى تَحد من أداء المشخّص التقليدى إعتماداً على أنواع محددة من نماذج شبكات بترى. هذا البحث يقدم أيضاً نظام مراقبة بإستخدام زوج من الروبوت، كمثال تطبيقى، لإختبار التقنية المقترحة.

Abstract:- The control of complex automated systems requires a supervision system that includes coordination as well as fault detection and diagnosis capabilities. The objective of coordination system is to enforce the plant to legal and safe behavior, while the objective of the fault detection and isolation is to detect and isolate the failures according to the observation traces generated by the system. This paper proposes a new supervisory control technique for online fault detection, isolation and coordination based on Petri net models. This technique has two components, a supervisor, and a diagnoser. The latter evaluates the faulty states without checking the properties that restrict the performance of the conventional dignosers to certain types of Petri net models. This paper also presents two-robot surveillance system, as an application example, to test the proposed algorithm.

Keywords:- Discrete event systems; Fault detection and isolation; Supervisory control theory; Petri nets; Hybrid systems, Robotics

1-INTRODUCTION

Safety plays a crucial role for the reliability of complex automated systems. In such systems this feature is incorporated mainly to eliminate unnecessary risks. The fault detection is very important for the safety of both systems and humans and also maintains the production rate.

Such automated systems are composed of different elements (such as machines, feeders, controllers, etc.); the interaction among these elements can be characterized as discrete, asynchronous, and sequential. Therefore, the process synchronization, deadlock avoidance, etc. should be considered as the main problems in both

Accepted June 30, 2009.

normal and abnormal cases [1]. These characteristics allow the system to be considered as Discrete Event System (DES) and allow the researchers to perform the analysis and control of such systems using uniform approaches. Discrete Event System (DES) is a dynamic system with state evolution produced by the occurrence of physical events. An event may correspond to the movement of a work piece in a transport system, the occurrence of a disturbance or the change in the set point in the control system, etc. DES can be found in domains such as manufacturing, robotic, traffic control, logistics, and communication systems, etc.

Important contributions in supervisory control of DES based on the Finite Automata (FA) and Petri Nets (PNs) are found in [2-4]. Petri net models are normally more compact than similar automata based models and are better suited for the representation of discrete event systems. They also have a good representational power [3]. Some of the common successful applications of Petri nets in discrete event modeling and control are flexible manufacturing [5-6], industrial automation [7], robotics [8-9] and batch processes [10-11]. Numerous approaches to the systematic construction of Petri net models have been proposed in [12]. From this point of view, PNs are employed as systematic techniques for the modeling of DES.

Automated systems are subjected to faults due to the physical characteristics of their components and the complex interaction among their parts. The components of automated systems such as sensors and actuators are subjected to unexpected faults, which produce unacceptable deviations from nominal conditions [13]. Once a fault has been detected, the control law can be modified in order to safely continue the operations [14]. Researchers have been focused on the fault diagnosis and detection in the field of discrete event systems [15-19]. Fault Detection and Isolation (FDI) are important issues for discrete event systems and for Hybrid Dynamical Systems (HDS) [20] which have received a lot of attention in the last years. It has been motivated by the practical need of ensuring the correct and safe functioning of large complex systems.

Recently, the model based approach has been held for addressing fault detection and location in DES. Most of the published papers reported the use of finite automata (FA) as modeling formalism. This is achieved by the use of a special type of automaton, called the *diagnoser*, which is synthesized from the system model. The diagnoser can also be used to analyze the diagnosability properties of the system (off-line) according to the formal definition introduced in [15-16]. The DES model proposed in [21], includes "normal" as well as "failed" behavior for a given set of faults modeled as unobservable events (i.e., not directly measured by the system sensors). A modular structure approach to perform failure diagnosis in DES was proposed in [22] as an extension to the work published in [15-16].

The disadvantage of the FA is its need for explicit determination of all the system states (state explosion problem). In order to deal with such problem, Petri Net (PN) models have been used in the context of DES fault detection. Concerning the works using Petri nets as modeling formalism, recently [23-24] proposed a method that handles the reachability graph of the PN model in order to perform a similar analysis of the underlying FA model. Such works take the advantage of the descriptive power of PN in the modeling phase, but the diagnosability test, which is based on

reachability graph, is limited to small size systems.

There are two approaches for fault detection and isolation using Petri nets, the first is based on artificial intelligence while the second is model based. In [25], based on artificial intelligence, a method of constructing fault diagnosis systems for batch processes using Fuzzy Petri-Nets (FPNs) is presented. Based on Petri net models, two different approaches exist. With the first approach, events are observed and faulty behaviors are modeled as forbidden states in the Petri net [26]. In this case the firing of the transitions is measured and the marking of the places has to be estimated. In the second approach, the state is observed and the faulty behaviors are modeled as the firing of failure transitions [27]. In this case, the marking of the places is measured and the firing sequences are estimated.

In [18] an on-line approach for fault diagnosis of DES by the interpreted PN formalism was proposed. In that framework, many theoretical results are presented concerning diagnosability conditions (if the system is diagnosable or not), i.e. all faults can be detected. The main drawback of this approach is the difficulty to derive a diagnosability test.

The main objective of this paper is to propose a systematic supervisory control approach not only for fault detection and isolation but also for coordination. The proposed approach extends the work previously published in [18] and [28]. This paper considers Petri net models of discrete event systems with faulty behaviors, which are triggered by the firing of failure transitions. Events are observed and faulty behaviors are modeled as faulty states in the Petri net. In this case, the firing of transition is measured and the marking of the places has to be checked. In this paper, a modified algorithm is proposed, which is based on the online computation of the set of possible fault states. This is efficiently achieved by modeling the plant by Petri nets, since their mathematical representation allow formulating the fault diagnosis problems in terms of mathematical programming.

The main contributions of this paper can be summarized as follows:

1- Developing a discrete event model for a robotic system using Petri nets
2- Synthesizing a supervisor for the modeled system
3- Modifying the conventional fault diagnosis scheme
4- Developing a supervisory control scheme for online fault detection & isolation and coordination tasks by merging steps 1, 2, and 3
5- Testing the developed scheme using two-robot surveillance system

This paper is organized as follows: In section 2, the basic notations of Petri nets are formulated. Section 3 describes the employed supervision technique. The proposed algorithm is detailed in Section 4. Section 5 employs a robotic system to illustrate the ideas of coordination and fault detection and isolation; simulation results are also presented. Section 6 concludes the topics issued through the paper.

2- NOTATIONS OF PETRI NETS

Definition 1 (Petri net graph)

A Petri net graph (or structure) is a weighted bipartite graph $G = (P, T, F, W)$ where:

$P = \{p_1, p_2, \ldots p_n\}$ is a finite set of places represented by circles, $|P| = n$.

$T = \{t_1, t_2, \ldots t_m\}$ is a finite set of transitions represented by bars, $|T| = m$.

$F \subseteq (P \times T) \cup (T \times P)$ is a set of arcs from places to transitions and from transitions to places in the graph.

$W : I \cup O \rightarrow \{1, 2, 3, \ldots\}$ is a weight function on the arcs. A Petri net is said

to be ordinary if all of its arc weights are *1's*.

${}^{\bullet}t = I(t_j) = \{p_i \in P : (p_i, t_j) \in F\}$ is a set of input places of a transition t_j.

$t^{\bullet} = O(t_j) = \{p_i \in P : (t_j, p_i) \in F\}$ is a set of output places of a transition t_j.

Similar notation can be used for places i.e. ${}^{\bullet}p = I(p_j)$ and $p^{\bullet} = O(p_j)$.

Definition 2 (Petri net Marking)

The marking function $M : P \longrightarrow Z^{+}$ represents the number of tokens (depicted as dots) residing inside each place. The marking of a PN is usually expressed as an n-entry vector $M=\{m_{p1}, m_{p2}, \ldots m_{pn}\}$. Where Z^{+} are the non negative integers.

Definition 3 (Marked Petri net)

Marked net (G, M_0) is a net G with an initial marking M_0, where G is a Petri net graph.

Definition 4 (output function)

$\Omega : R(G, M_0) \longrightarrow (Z^{+})^{q \times n}$ is an output function, that associates to each marking in $R(G,M_0)$ q-entry output vector; q is the number of outputs. Ω has a dimension of $q \times n$. Each column of Ω is an elementary or null vector. If the output symbol i is present (turned on) every time that $M(p_j) \geq 1$, then $\Omega(i,j)=1$, otherwise $\Omega(i,j)=0$.

Definition 5 (measurable and non measurable places)

A place $p_i \in P$ is said to be measurable if the i-th column vector of Ω is not null, i.e. $\Omega(\bullet, i) \neq 0$. Otherwise it is non-measurable [18]. $P_m=\{p_1, p_2, \ldots, p_{mp}\}$ is a set of measurable places.

Definition 6 (incidence matrix)

The architecture or layout of a Petri net can be represented with an integer matrix known as the incidence matrix. The incidence matrix D of a Petri net is an $n \times m$ matrix whose (i, j) entry is of the form $d_{ij} = w(t_j, p_j) - w(p_i, t_j)$. The incidence matrix is useful in studying the reachability problem [1], [29].

Definition 7 (enabled transition)

In a *PN* system, a transition $t_j \in T$ in a Petri net is said to be enabled if $m_k(p_i) \geq w(p_i, t_j)$ for all $p_i \in I(t_j)$. An enabled transition t_j can be fired reaching a new marking M_{k+1} which can be computed as (PN state equation):

$$M_{k+1} = M_k + Dv_k \quad (1)$$

$$y_{k+1} = \Omega M_{k+1} \quad (2)$$

Where $v_k(i)=0$, $i \neq j$, $v_k(j)=1$, v is the firing count vector.

Definition 8 (firing sequence)

A firing sequence from M_0 is a (possibly empty) sequence of transitions $\sigma = t_1 \ldots t_k$ such that $M_0[t_1\rangle M_1[t_2\rangle M_2 \ldots [t_k\rangle M_k$, where $M_0[\sigma\rangle M_k$ denotes that σ may be fired at M_0 yielding M_k.

Definition 9 (reachable states)

A marking M is *reachable* in (G, M_0) if there exists a firing sequence σ such that $M_0 [\sigma\rangle M$. Given a marked net (G, M_0), the set of firing sequences (also called *language* of the net) is denoted $L(G, M_0)$ and the set of reachable markings (also called *reachability set* of the net) is denoted $R(G, M_0)$.

Definition 10 (Place invariants (P-invariants))

P-invariants are sets of places whose weighted token count remains constant for all possible markings. *A P-invariant is defined as every integer vector X that satisfies* $X^T M = X^T M_0$. The place invariants of a net can be computed by finding integer solutions to $X^T D = 0$.

3- SUPERVISION OF DISCRETE EVENT SYSTEMS

The *Supervisory Control Theory (SCT)* was introduced as a conceptual framework for studying the supervision (control) of discrete event systems [2], [30]. Supervisory control is the process of limiting the actions of a discrete event system to a set of safe, allowable, and desirable behaviors.

One of the most efficient supervision techniques in the field of discrete event system is the Supervision Based on Place Invariants (SBPI) [31]. The technique has been successfully applied in many applications [11] and [32]. The resulting supervisors are themselves Petri nets. Supervisor synthesis based on place invariant technique forms the basics of the synthesis procedure in this paper. Briefly this method would be formally described as follows: the supervisory control goal is to restrict the reachable markings of a plant, m_p such that:

$$LM_p \le B \tag{3}$$

Where M_p is the marking vector of the plant, $L \in Z^{nc \times n}$, $B \in Z^{nc}$, $M_c \in Z^{nc}$ and n_c is the number of constraints to be enforced on the plant model. The system to be controlled is modeled by a Petri net with n places and m transitions and is known as the plant or process net. The incidence matrix of the plant is $D_p \in Z^{n \times m}$. It is possible that the process net will violate certain constraints posed on its behavior, thus the need for supervision. The inequality (3) can be transformed into equality by introducing a nonnegative slack variable M_c into it. Then (3) becomes:

$$Lm_p + M_c = B \tag{4}$$

The slack variable M_c in this case contains new places that hold the extra tokens required to meet the equality. The slack variable that enforces the equality (4) is a part of a separate net called the supervisor (controller) net. It is shown in [31] that if the initial marking does not violate the given set of constraints (3), these constraints can be enforced by a supervisor with the incidence matrix:

$$D_c = -LD_p \tag{5}$$

The initial marking of the controller is computed by:

$$M_{c0} = B - LM_{p0} \tag{6}$$

Where M_{p0} is the $n \times 1$ initial plant marking vector of non-negative integers. The controller net is a Petri net with incidence matrix D_c made up of the process net's transitions and a separate set of supervisor places. The supervised net is also called the controlled system or the closed loop system:

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix}, M = \begin{bmatrix} M_p \\ M_c \end{bmatrix} \tag{7}$$

4-PROPOSED FAULT DETECTION ALGORITHM

4.1. Problem Statement

DES approaches to fault diagnosis are suitable for failures that cause a distinct change in the state of system components but do not bring the system to stop: examples are equipment failures. Contributions of the supervisory control in fault detection have been developed considering faults as forbidden states [3], [30].

In the fault diagnosis literature, there are mainly two types of Petri-net-based diagnosers, namely *compiled* diagnosers, whose policy is to provide the set of faults that could have happened at each state transition [22]. This approach is based on the offline computation of the set of fault events that may have occurred at each reachable state, providing a fast online diagnosis at a price of excessive memory requirements. Another type is the

interpreted diagnoser, which computes online the set of faults that could have happened, or a set of fault states the system could have reached, after each observed event [28].

In [18] a reduced interpreted diagnoser has been devised only for safe PNs with an output function that associates an output vector to each net marking (interpreted PNs). The main drawbak of this approach is the difficulty to derive a diagnosability test (conditions to establish if the system is diagnosable or not, i.e. all faults can be detected). In [28], an interpreted diagnoser based on the online solution of programming problems was proposed. The problem of diagnosability check is not addressed. The notation of *g-markings* which extends the classic net marking is introduced in that work.

4.2. Contribution of this Paper

This paper proposes a new supervisory control algorithm for online fault detection and isolation as well as coordination based on Petri net models. The proposed algorithm is based on the SBPI technique merged with modified fault isolation one. The supervisor and diagnoser are also Petri nets-based models, which are merged with the original Petri net model of the system. Such nets are used to force the system to desired behavior and to detect faults under the current net marking.

From the fault detection point of view, this paper proposes an interpreted diagnoser. The idea was borrowed from [18] and enhanced using the notation of g-marking described in [28]. The proposed diagnoser is promising for complex industrial automation systems with a large number of system states.

4.3. The Proposed Algorithm

This paper develops a supervisory control schemes for online fault detection, isolation and coordination tasks. The proposed scheme consists of two components in a unified framework. These components are the supervisor, and the diagnoser. The developed scheme can be detailed as follows:

1- Constructing the normal and faulty models of the system

- The set T can be partitioned into the disjoint sets of *normal* transitions T_N and *faulty* transitions T_f, where:
  $T_N = \{t_{n1}, t_{n2} \ldots t_{nR}\}$, *such that* $|T_N| = R$; R is number of normal transitions.
  $T_f = \{t_{f1}, t_{f2} \ldots t_{fL}\}$, *such that* $|T_f| = L$; L is number of faulty transitions
- Similarly, The set of places of this model is partitioned into *normal* places P_N and *faulty* places P_f, *where:*
  $P_N = \{p_{n1}, p_{n2} \ldots p_{nu}\}$, *where* $|P_N| = u$;
  $P_f = \{p_{f1}, p_{f2} \ldots p_{fs}\}$, *where* $|P_f| = s$
- (G, M_0) represents the *normal* behavior of the system i.e. when no failures are considered.
- (G_f, M_{0f}) represents the *faulty* behavior of the system.
- $P_R = {}^\bullet T_f$ is the set of risky places of (G_f, M_{0f}) i.e. all input places of faulty transitions.
- $T_R = P_R^\bullet \cap T_N$ is the post-risk transition set of (G_f, M_{0f}).

2- Supervisor Synthesizing

- Given the set of constraints (3), these constraints can be enforced by a supervisor with the incidence matrix and initial marking computed using (4) and (5).
- (G_c, M_{c0}) represents the behavior of the supervisor with the incidence matrix D_c and initial marking M_{c0}.

3- Combining the plant model with the supervisor

- (G_N, M_{0N}) represents the *normal* behavior of the supervised system (combined plant/supervisor model).

- Connect (G_N, M_{0N}) to the places of P_f through the transitions representing the faults T_f.
- The matrix Ω_N represents the output matrix of (G_N, M_{0N}). Similarly Ω_f is extended to represent (G_f, M_{0f}) i.e. one column is added to each faulty place in P_f.

4- Computation of matrices Ψ_N and Ψ_f matrices

A matrix Ψ_N is computed as follows [18]:

1- A *base number* Θ is computed using the following formula:

$$\Theta = 2 \max(\text{abs}(D_N(i,j))) + 1 \quad (8)$$

2- If $p_i \in P_N$ is a non-mensurable place i.e. $\Omega_N(\bullet,i)=0$ then $\Psi_N(i)=0$ (the information of this place cannot be used by the diagnoser). It is named a zero entry of Ψ_N; otherwise it is named non-zero entry. The first non-zero entry of Ψ_N, named $\Psi_N(i)$ should be equal to Θ^0, i.e. $\Psi_N(i) = 1$. Then the next non-zero entry of Ψ_N, named $\Psi_N(j)$ should be equal to Θ^1. This procedure continues until $i = (n+n_c)$. Hence matrix Ψ_N can be computed with the dimensions $(n+n_c)\times 1$ (*where* n_c is the number of supervisor places). All entries *of* Ψ_N vector are non-negative. The matrix Ψ_f can be computed to cover the faulty places in the plant model i.e. its dimensions are $(n+n_c+s) \times 1$ (*where* s is the number of faulty places).

5- Diagnoser synthesis

The proposed diagnoser model structure for the system normal behavior (G_N, M_0) is a PN(G_d,M_{d0}) with a single place p_d and a set of transitions $T_d=T_N$, the incidence matrix D_d can be computed as stated in [18] as following:

$$D_d = [\Psi_N]^T([\Omega_N]^T \Omega_N) D_N \quad (9)$$

Where D_N is the incidence matrix of (G_N, M_0). The marking of the diagnoser M_{d0} and M_{dk} are computed as:

$$M_{d0} = [\Psi_N]^T([\Omega_N]^T \Omega_N) M_0,$$
$$M_{dk} = [\Psi_N]^T([\Omega_N]^T \Omega_N) M_k \quad (10)$$

- (G_d, M_{d0}) represents the behavior of the diagnoser with the incidence matrix D_d and initial marking M_{d0}.

6- Error computation

If a transition $t_i \in T - (T_R \cup T_f)$ is fired in (G_N, M_0) then it is fired in (G_d, M_{d0}). If a transition $t_j \in T_R$ is enabled in the system then t_j must be fired in diagnoser. Thus, if t_j is not fired in the system then the output of the system and the output of the diagnoser are different. In this case, the system reached a faulty marking M_f. The error is computed by the following equation:

$$e_k = M_{dk} - [\Psi_f]^T([\Omega_f]^T \Omega_f) M_k \quad (11)$$

7- Fault Isolation

When $e_k \neq 0$, it means that there is a difference between the system output and the diagnoser, then a faulty marking is reached. The mechanism used to find out the faulty marking is called fault isolation. A modified fault isolation algorithm is proposed in this paper. The modification extends the work published in [18] and uses the concept of g-marking in [28]. The proposed algorithm is as follows:

Inputs: M_k, M_{dk}, e_k; where, M_k is the marking vector of the supervised normal process, M_{dk} is the marking of the diagnoser, and e_k is the error between them.

Outputs: p_f (faulty place), M_f (faulty marking)

Constants: D_d is the diagnoser incidence matrix described above.

Conventional diagnoser:

For $i=1 : |T_N|$

If $e_k = D_d(1,i)$ Then

$\forall p \in {}^\bullet t_i$ Put $M_k(p) = 0$

$\forall p \in t_i^\bullet$ Put $M_k(p) = 0$

$\forall p_f \in ({}^\bullet t_i)^{\bullet\bullet} \cap P_f$ Put $M_k(p_f) = 1$ and $M_f = M_k$ where $p_f \in P_f$

End
End

Proposed diagnose: *Searching for the input place of the transition that is in conflict with the faulty transition* is as follows:

For $x=1:u+n_c$

If $(M_{k+1}(x)=-1)$ Then /*M_{k+1} is a *g-marking vector* (A marking that may have negative components is called g-marking (generalized marking)). x is the index of the place whose marks are stolen due to fault occurrence.*/

$\forall p \in {}^{\bullet}t_x$ Put $M_k(p)=0$

$\forall p \in t_x^{\bullet}$ Put $M_k(p)=0$

$\forall p_f \in ({}^{\bullet}t_x)^{\bullet\bullet} \cap P_F$ Put $M_k(p_f)=1$ and $M_f=M_k$

End
End

The schematic diagram for the proposed algorithm is shown in Figure 1.

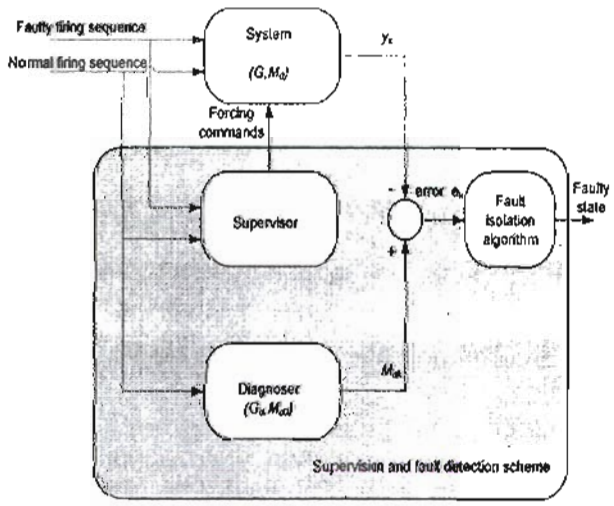


Fig. 1. The proposed supervision and fault detection scheme.

5. APPLICATION EXAMPLE: TWO-ROBOT SURVEILLANCE SYSTEM

This section employs two-robot remote surveillance system [33] as an application example to test the proposed algorithm. The scenario is to employ a team of robots to complete more sophisticated tasks than what each robot is capable of achieving. These autonomous robots will be exposed to internal and external threats, such as system faults/failures and component damages, hence they will be required to operate with the ability of being reconfigured in real-time. These requirements pose great challenges to the command and control of the robot team as well as fault detection and isolation.

Such system, which is composed of two robots named Robot-h and Robot-c. The application is similar to the "cat and mouse" problem which is a popular example in the field of discrete event system supervision [2]. These two robots are placed on a floor with five rooms. The rooms are connected with doors through which the two robots can pass and the moving directions for each robot are shown in Figure 2.a and Figure 2.b, respectively.

The Petri net model of the two robots is shown in Figure 3. The upper Petri net concerns robot-h while the lower one concerns robot-c. The transitions model the ability of each robot to pass from one room to the other. Table 1 explains places and transitions.

The supervisor is synthesized using SBPI technique. It is desired to coordinate the movement of the two robots such that collisions must not be occurred. The problem is to control the doors so that the two robots can never be in the same room simultaneously. This can be achieved forcing the model to obey constraints in the form of (3). Hence the coordination constraints can be formulated in the form:

$$m_{h2} + m_{c6} \le 1 \quad (12)$$

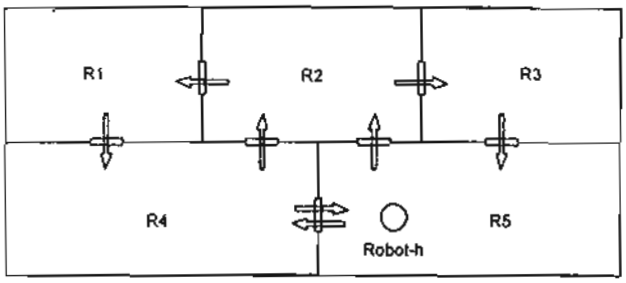
$$m_{h4} + m_{c4} \le 1 \quad (13)$$

$$m_{h6} + m_{c2} \le 1 \quad (14)$$

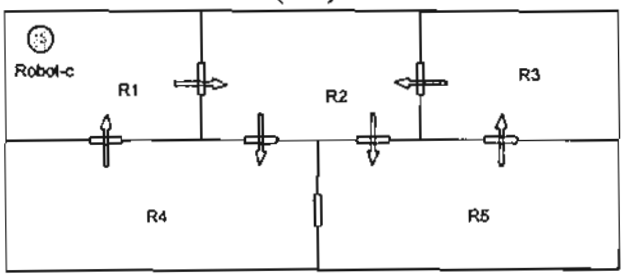
$$m_{h7} + m_{c11} \le 1 \quad (15)$$

$$m_{h9} + m_{c9} \le 1 \quad (16)$$

$$m_{h11} + m_{c7} \le 1 \quad (17)$$



(2.a)



(2.b)

Fig. 2. Two-robot remote surveillance system with the moving directions for (a) robot-h, and (b) robot-c.

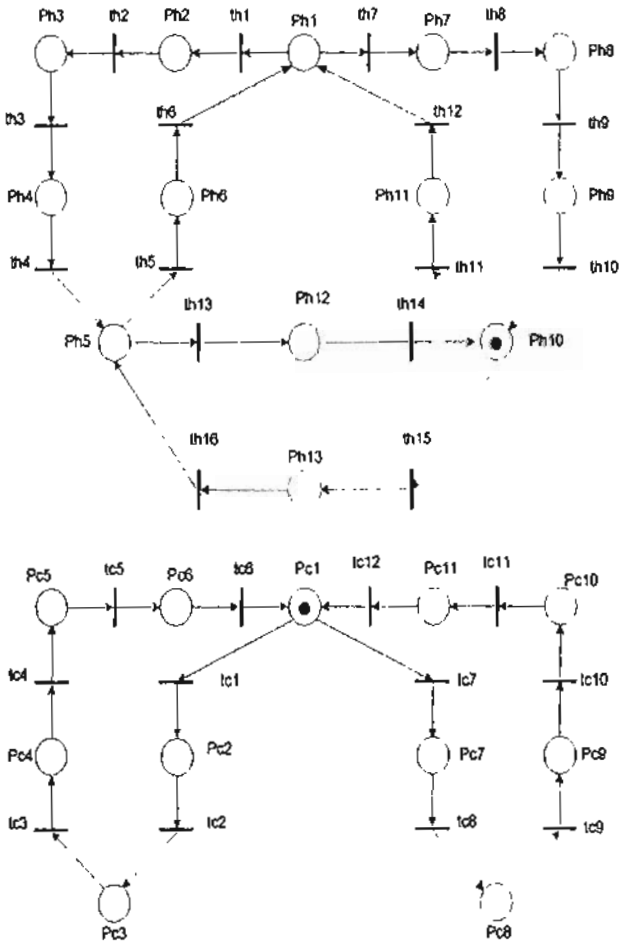


Fig. 3. PN model of the two robot surveillance system.

Table 1. Illustration of places and transitions.

Place	Associated action
p_{h1}	Robot-h is in R_2
p_{h2}	Moving from R_2 to R_1
p_{h3}	Robot-h is in R_1
p_{h4}	Moving from R_1 to R_4
p_{h5}	Robot-h is in R_4
p_{h6}	Moving from R_4 to R_2
p_{h7}	Moving from R_2 to R_3
p_{h8}	Robot-h is in R_3
p_{h9}	Moving from R_3 to R_5
p_{h10}	Robot-h is in R_5
p_{h11}	Moving from R_5 to R_2
p_{h12}	Moving from R_4 to R_5
p_{h13}	Moving from R_5 to R_4
p_{c1}	Robot-c is in R_2
p_{c2}	Moving from R_2 to R_4
p_{c3}	Robot-c is in R_4
p_{c4}	Moving from R_4 to R_1
P_{c5}	Robot-c is in R_1
p_{c6}	Moving from R_1 to R_2
p_{c7}	Moving from R_2 to R_5
p_{c8}	Robot-c is in R_5
p_{c9}	Moving from R_5 to R_3
p_{c10}	Robot-c is in R_3
p_{c11}	Moving from R_3 to R_2
Transition	**Associated event**
t_{h1}	start moving from R_2 to R_1
t_{h2}	end moving from R_2 to R_1
t_{h3}	start moving from R_1 to R_4
t_{h4}	end moving from R_1 to R_4
t_{h5}	start moving from R_4 to R_2
t_{h6}	end moving from R_4 to R_2
t_{h7}	start moving from R_2 to R_3
t_{h8}	end moving from R_2 to R_3
t_{h9}	start moving from R_3 to R_5
t_{h10}	end moving from R_3 to R_5
t_{h11}	start moving from R_5 to R_2
t_{h12}	end moving from R_5 to R_2
t_{h13}	start moving from R_4 to R_5
t_{h14}	end moving from R_4 to R_5
t_{h15}	start moving from R_5 to R_4
t_{h16}	end moving from R_5 to R_4
t_{c1}	start moving from R_2 to R_4
t_{c2}	end moving from R_2 to R_4
t_{c3}	start moving from R_4 to R_1
t_{c4}	end moving from R_4 to R_1
t_{c5}	start moving from R_1 to R_2
t_{c6}	end moving from R_1 to R_2
t_{c7}	start moving from R_2 to R_5
t_{c8}	end moving from R_2 to R_5
t_{c9}	start moving from R_5 to R_3
t_{c10}	end moving from R_5 to R_3
t_{c11}	start moving from R_3 to R_5
t_{c12}	end moving from R_3 to R_5

The supervisor is computed by equations (5) and (6). Here the marking vector of the plant has a dimension of

24x1, the matrix L is of dimension 6x24 and the incidence matrix D_p is of dimension 24x28. The supervisor consists of six places P_{s1}, P_{s2}, P_{s3}, P_{s4}, P_{s5}, and P_{s6} that are linked to the Petri net model of the plant as shown in Figure 4.

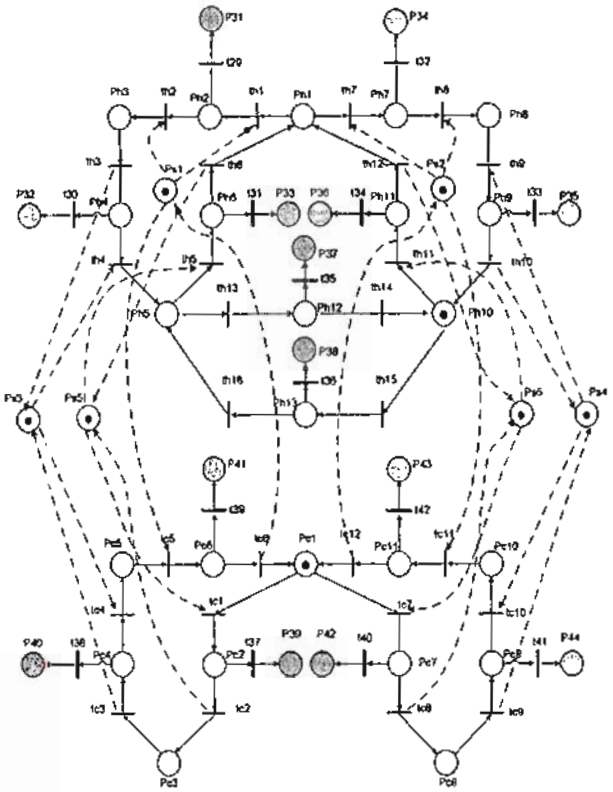


Fig. 4. The model of supervised system with faulty states.

Based on the faulty system model shown also in Figure 4, the following sets can be inspected:

T_N= {t_{h1}, t_{h2}, t_{h3}, t_{h4}, t_{h5}, t_{h6}, t_{h7}, t_{h8}, t_{h9}, t_{h10}, t_{h11}, t_{h12}, t_{h13}, t_{h14}, t_{h15}, t_{h16}, t_{c1}, t_{c2}, t_{c3}, t_{c4}, t_{c5} t_{c6}, t_{c7}, t_{c8}, t_{c9}, t_{c10}, t_{c11}, t_{c12}}

T_f= {t_{29}, t_{30}, t_{31}, t_{32}, t_{33}, t_{34}, t_{35}, t_{36}, t_{37}, t_{38}, t_{40}, t_{41}, t_{42}}

P_N= {p_{h1}, p_{h2}, p_{h3}, p_{h4}, p_{h5}, p_{h6}, p_{h7}, p_{h8}, p_{h9}, p_{h10}, p_{h11}, p_{h12}, p_{h13}, p_{c1}, p_{c2}, p_{c3}, p_{c4}, p_{c5}, p_{c6}, p_{c7}, p_{c8}, p_{c9}, p_{c10}, p_{c11}}

P_f= {p_{31}, p_{32}, p_{33}, p_{34}, p_{35}, p_{36}, p_{37}, p_{38}, p_{39}, p_{40}, p_{41}, p_{42}, p_{43}, p_{44}}

$P_R = {}^{\bullet}T_f$ = { p_{h2}, p_{h4}, p_{h6}, p_{h7}, p_{h9}, p_{h11}, p_{h12}, p_{h13}, p_{c2}, p_{c4}, p_{c6}, p_{c9}, p_{c11}}

$T_R = P_R^{\bullet} \cap T_N$= {$t_{h2}$, t_{h4}, t_{h6}, t_{h8}, t_{h10}, t_{h12}, t_{h14}, t_{h16}, t_{c2}, t_{c4}, t_{c6}, t_{c8}, t_{c10}, t_{c12}}

P_m= {p_{h1}, p_{h3}, p_{h5}, p_{h8}, p_{h10}, p_{c1}, p_{c3}, p_{c5}, p_{c8}, p_{c10} }

The output matrix Ω of the normal plant has 10 rows ($q=10$; mumber of outputs), and 30 columns ($n=30$; number of supervisory and normal places). The diagnoser is computed by equations (9) and (10). The diagnoser is a single place connected to all the normal transitions of the systems. For clarity, the diagnoser is shown in Figure 5. The computation values of Ψ_N as follows: Θ=3; The non-zero entries of Ψ_N are tabulated in Table 2.

Table 2. Non-zero entries of Ψ_N.

i	1	3	5	8	10
Ψ_M(i,1)	1	3	9	27	81
i	14	16	18	21	23
Ψ_M(i,1)	243	729	2187	6561	19683

The simulation of the supervised system is carried out, using MATLAB environment, and the results are indicated in Table 3. Firing the transition t_{33} using the conventional diagnoser [18] results in two faulty places, P_{35} and P_{37} respectively. This is due to that the incidence matrix of the conventional diagnoser has two similar values in the 10th and 14th columns that match the error detected (which is 81). Unlike the conventional diagnoser, the proposed diagnoser detects only one faulty place (state), P_{35}, due to the usage of the *g-marking* vector. In table 3, for the last row, A fault is simulated to occur while Robot-h was moving from room 3 to room 5; detection of faulty state indicated by the place p_{35},

M=[p_{h1},p_{h2},p_{h3},p_{h4},p_{h5},p_{h6},p_{h7},p_{h8},p_{h9},p_{h10}, p_{h11},p_{h12},p_{h13},p_{c1},$p_{c2}$$p_{c3}$,$p_{c4}$,$p_{c5}$,$p_{c6}$,$p_{c7}$,$p_{c8}$, p_{c9},p_{c10},p_{c11},p_{s1},p_{s2},p_{s3},p_{s4},p_{s5},p_{s6},p_{31},p_{32}, p_{33},p_{34},p_{35},p_{36},p_{37},p_{38},p_{39},p_{40},p_{41},p_{42},p_{43}, p_{44}].

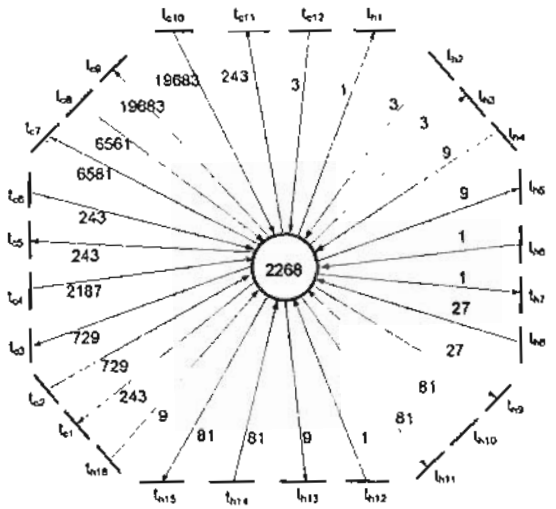


Figure 5. Diagnoser model for two robots surveillance system.

Table 3. simulation results of the two robots surveillance system.

Event	Marking	y_k	M_{dk}	c_k
initiali-zation	[0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,1,0, 0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0] Robot-h is in room 5 and Robot-c is in room 1	2268	2268	0
t_{h15} fires	[0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0, 0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0]	2187	2187	0
t_{h16} fires	[0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0 ,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2196	2196	0
t_{h5} fires	[0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0, 0,0,1,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2187	2187	0
t_{h6} fires	[1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0, 0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2188	2188	0
t_{h7} fires	[0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0, 0,0,1,1,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2187	2187	0
t_{h8} fires	[0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0, 0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2214	2214	0
t_{h9} fires	[0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,1,0,0,0,0, 0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0]	2187	2187	0
t_{33} fires	[0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0, 0,0,1,1,0,1,1,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0]	2187	2268	81

6. CONCLUSIONS

This paper develops a new scheme for coordination and on-line monitoring and diagnosis of discrete event systems modeled by Petri nets. The scheme comprises two components in a unified framework. These components are the supervisor, and the diagnoser. The latter is a modified version of a conventional interpreted diagnoser that is limited to certain types of Petri nets. The proposed scheme also detects any fault in the system online in an efficient manner. The simulation results obtained reflects that the proposed scheme is promising for real time control and monitoring of complex industrial automation systems.

REFERENCES

[1] T. Murata, "*Petri nets: Properties, analysis and applications*", Proceedings of IEEE, vol. 77, pp. 541–580, 1989.

[2] P. J. Ramadge, and W. M .Wonham, "*The Control of Discrete Event Systems*", Proceedings of the IEEE, vol. 77, pp. 81-98, 1989.

[3] M. V. Iordache and P. J. Antsaklis, "*Supervision Based On Place Invariants: A Survey*", Tech. Report of the ISIS Group at the University of Notre Dame ISIS-2004-003, 2004.

[4] F. Basile, A. Giua, C. Seatzu, "*Some new results on supervisory control of Petri nets with decentralized monitor places*" 17th IFAC World Congress (Seoul, Korea), July 2008.

[5] M. Caradec, and F. Prunet, "*A new modeling tool for hybrid flexible systems*", IEEE International Conference on Systems, Man, and Cybernetics, vol. 3, pp. 2246 – 2251, 1997.

[6] R. Zurawski, and M. Zhou, "*Petri nets and industrial applications: A tutorial*", IEEE Transactions on Industrial Electronics, vol. 41, pp. 567–582,1994.

[7] I. Demongodin, and N. T. Koussoulas "*Differential Petri Net models for industrial automation and supervisory control*", IEEE Trans. on Systems, Man, And Cybernetics-part c: Application and Reviews, vol.36, no.4, 2006.

[8] P. Freedman, "*Time, Petri Nets, and Robotics*", IEEE Transaction on Control Systems Technology, vol. 7, pp. 417-433,1991.

[9] T. M. Sobh, and B. Benhabib, "*Discrete event and hybrid systems in robotics and automation: an overview*", IEEE Robotics & Automation Magazine, vol. 4, pp. 16–19, 1997.

[10] T. Gu, and Parsia A. Bahri, "*A survey of PN applications in batch processes*", Computer in Industry, vol. 47, pp. 99-111, 2002.

[11] M. M. Gomaa, H. A. Awad, and A. R. Anwar, "*Design and Implementation of Supervisory Control Schemes in Industrial Automation Systems*", Proceedings of IEEE International Conference on Computer Engineering and Systems, ICCEC'08, pp. 398-404, 25-27, Cairo, Egypt, 2008.

[12] M. D. Jeng, and F. DiCesare, "*A Review of Synthesis Techniques for Petri Nets with Applications to Automated Manufacturing Systems*", IEEE Transaction on Systems, Man, and Cybernetics, vol. 23, pp. 301-312, 1993.

[13] Zhu Miaofen, Chen Guojin, and Wang Yaka, "*Fault Diagnosis Based on Petri Model in Machining Process*", Proceedings of the IEEE Instrumentation and Measurement Technology Conference, Sorrento, Italy, pp. 2077-2080, April 2006.

[14] A. Jalivand, and S. Khanmohammadi, "*Integrating of Mode Detection and Mode Recognition in Hybrid Systems by Fuzzy Petri Nets*", Proceeding of the IEEE Conference on Robotics, Automation, and Mechatronics, Singhapore, pp. 265-270, Dec. 2004.

[15] M. Sampath, R. Sengupta, S. Lafortune, K. Sinnamohideen, and D. Teneketzis, "*Diagnosability of discrete event systems*", IEEE Transaction on Automatic Control, vol. 40, no. 9, pp. 1555–1575, Sept. 1995.

[16] A. Paoli, and S. Lafortune, "*Safe diagnosability for fault-tolerant supervision of discrete-event systems*", Automatica , vol. 41, pp. 1335 – 1347, 2005.

[17] T. Both, H. M. Hanisch, and J. Jorn, "*Fault treatment with net condition/event systems: a first approach*", The 8th IEEE International Conference on Emerging Technologies and Factory Automation Proceedings, 2001.

[18] E. R. Beltrán, I. J. Ochoa , A. R. Treviño, E. L. Mellado, and M. M. Campaña. "*Fault detection and location in DES modeled using Petri Nets*", Proceedings of the International Conference on Systems, Man and Cybernetics, pp. 1645-1650, 2005.

[19] M. P. Cabasino, A. Giua, C.N. Hadjicostis, and C. Seatzu, "*Fault model identification with Petri nets*", The 9th International Workshop on Discrete Event Systems (Gteborg, Sweden), pp. 455-461, May 2008.

[20] E. R. Loures, and J. C. Pascal, "*Detection and Diagnosis of Hybrid Dynamic Systems Based on Time Fuzzy Petri Nets*", IEEE International Conference on Systems, Man and Cybernetics, vol.2, pp. 1825–1831, Oct. 2004.

[21] A. Giua, and C. Seatzu, " *Fault detection for discrete event systems using Petri nets with unobservable transitions*", Proceedings of 44th International Conference on Decision and Control and European Control Conferences, eville, Spain, pp. 6323- 6328, Dec. 2005.

[22] O. Contant, S. Lafortune, and D. Teneketzis, "*Diagnosability of Discrete Event Systems with Modular Structure*", Discrete Event Dynamic Systems, vol. 16, pp.9–37, 2006.

[23] Y. L. Wen, and M. D. Jeng, "*Diagnosability of Petri Nets*", IEEE Intematlonal Conference on Systems, Man and Cybernetics, vol. 5, pp. 4891 – 4896, Oct. 2004.

[24] D. Lefebvre, and C. Delherm, "*Diagnosis of DES With Petri Net Models*", IEEE Trans. on Automation Science and Engineering, vol. 4, pp. 114-118, Jan. 2007.

[25] L. Zhenjuan, and P. B. L. Hongguang, "*Batch Process Fault Diagnosis Based on Fuzzy Petri Nets*", The 1st International Conference on Innovative Computing, Information and Control, ICICIC'06, vol. 2, pp. 474 - 477, Aug. 2006.

[26] A. Giua, and C. Seatzu, "*Observability of Place/Transition Nets*", IEEE Trans. on Aut. Control, vol. 47, pp. 1424 -1437, Sept. 2002.

[27] A. Giua, and C. Seatzu, "*Design of observers/controllers for discrete event systems using Petri nets*", Synthesis and Control of DES, B. Caillaud, X. Xie, Ph. Darondeau and L. Lavagno (Eds.), pp. 167-182, Kluwer, 2001.

[28] F. Basile, P. Chiacchio, and G. De Tommasi, "*An Efficient Approach for Online Diagnosis of Discrete Event Systems*", IEEE Trans. on Aut. Control, vol. 54, pp. 748-759, April 2009.

[29] R. David, and H .Alla, " *Petri Nets for Modeling of Dynamic Systems: A Survey*", Automatica, vol.30, pp. 175-202, 1994.

[30] W. M. Wonham, "*Supervisory control of discrete event systems*", Systems Control Group, Dept. of ECE, University of Toronto, ECE 1636F/1637S, 2008-2009. URL:*www.control.utoronto.ca/ people/profs/wonham/wonham.html.*

[31] A. Ragab, "*Modeling and supervision of discrete event systems*", M.Sc. Thesis, Faculty of Electronic Eng., Menofyia Unv., Egypt, 2007.

[32] D. Kezic, I. Vujovic, and I. Kuzmanic, "*Maximally Permissive Supervisor of Marine Canal Traffic System*", Proc. of the IEEE ITSC 2006 Conf. for Intelligent Transportation Sys., Toronto, Canada, pp.1424-1429, Sept. 2006.

[33] Jin-Shyan Lee and Pau-Lo Hsu, "*Applications of Petri Nets to Human-in-the-Loop Control for Discrete Automation Systems*", Manufacturing the Future, Concepts-Technologies-Visions, ISBN 3-86611-198-3, pp. 908, ARS/plV, Germany, July 2006, Edited by: V. Kordic, A. Lazinica, M. Merdan.