Journal of System Simulation

Volume 27 | Issue 3 Article 4

8-20-2020

Partially Observed System Design Method Realizing Unambiguous Fault Diagnosis

Fang Huan
1. School of Computer & Information, Heifei University of Technology, Hefei 230009, China;;2. College of Science, Anhui University of Science & Technology, Huainan 232001, China;;

Lu Yang
1. School of Computer & Information, Heifei University of Technology, Hefei 230009, China;;3. The Anhui Provincial Key Laboratory of Mine IoT and Mine Safety Supervisory Control, Hefei 230088, China;

Yue Feng
1. School of Computer & Information, Heifei University of Technology, Hefei 230009, China;;

Junming Guan
3. The Anhui Provincial Key Laboratory of Mine IoT and Mine Safety Supervisory Control, Hefei 230088, China;

Partially Observed System Design Method Realizing Unambiguous Fault Diagnosis

Abstract

Abstract: The construction methodology of partially observed system realizing unambiguous fault diagnosis was studied. The observable places determination algorithm OPD was proposed, and the necessary and sufficient condition for solution existence of the algorithm with polynomial time complexity was presented, then it is proved that all transitions in the modeled system could be distinguished by setting S_O. The system operating state determination algorithm SOSD was presented based on the OPD algorithm. The proposed SOSD algorithm was realized by 1-step forward marking computation of observable places set S_O, it doesn't rely on the initial marking M_0 of the controlled system while judging system status. It is proved that the proposed construction method for partially observed system satisfies the optimum supervisory condition, which achieves the optimal supervisory purpose.

Keywords

partially observed system, system design method, fault diagnosis, Petri nets, Event Based System

Recommended Citation

Fang Huan, Lu Yang, Yue Feng, Guan Junming. Partially Observed System Design Method Realizing Unambiguous Fault Diagnosis[J]. Journal of System Simulation, 2015, 27(3): 470-479.

第 27 卷第 3 期 系统仿真学报© Vol. 27 No. 3
2015 年 3 月 Journal of System Simulation Mar., 2015

实现故障无二义诊断的部分可观系统设计方法

方欢[1,2]，陆阳[1,3]，岳峰[1]，官骏鸣[3]
(1.合肥工业大学计算机与信息学院，合肥 230009；2.安徽理工大学理学院，淮南 232001；
3.安徽省矿山物联网与安全监控技术重点实验室，合肥 230088)

摘要：针对故障无二义性诊断下的部分可观系统设计方法进行分析和研究，*提出系统可见库所集 S_O 的确定算法 OPD，给出算法 OPD 解存在的充要条件，证明系统在 S_O 可见的情况下，系统所有变迁都是可区分的，并指出该算法满足多项式级的时间复杂度。*在 OPD 算法的基础上，*提出系统运行状态诊断算法 SOSD，SOSD 是 S_O 的一步前向标识计算方法，不需要已知系统的初始状态 M_0 就可以进行系统状态诊断。证明所提出的部分可观系统设计方法满足最优监控条件，*达到了优化监控的目的。

关键词：部分可观系统；系统设计方法；故障诊断；Petri 网；事件驱动系统
中图分类号：TP301 文献标识码：A 文章编号：1004-731X (2015) 03-0470-10

Partially Observed System Design Method Realizing Unambiguous Fault Diagnosis

Fang Huan[1,2], *Lu Yang*[1,3], *Yue Feng*[1], *Guan Junming*[3]
(1. School of Computer & Information, Heifei University of Technology, Hefei 230009, China;
2. College of Science, Anhui University of Science & Technology, Huainan 232001, China;
3. The Anhui Provincial Key Laboratory of Mine IoT and Mine Safety Supervisory Control, Hefei 230088, China)

Abstract: The construction methodology of partially observed system realizing unambiguous fault diagnosis was studied. *The observable places determination algorithm OPD was proposed, and the necessary and sufficient condition for solution existence of the algorithm with polynomial time complexity was presented, then it is proved that all transitions in the modeled system could be distinguished by setting S_O.* The system operating state determination algorithm SOSD was presented based on the OPD algorithm. *The proposed SOSD algorithm was realized by 1-step forward marking computation of observable places set S_O*, it doesn't rely on the initial marking M_0 of the controlled system while judging system status. *It is proved that the proposed construction method for partially observed system satisfies the optimum supervisory condition*, which achieves the optimal supervisory purpose.

Keywords: partially observed system; system design method; fault diagnosis; Petri nets; Event Based System

引言

离散事件动态系统(Discrete Event Dynamic Systems, DEDS)的监控理论自 Ramadge 和 Wonham 提出以来[1-2]，一直受到了学者们广泛的关注，并取得了一系列研究成果。而 DEDS 是一类典型的事件驱动系统(Event Based System, EBS)，在 EBS 的研究范畴中，故障检测与诊断对于保障系统的安全性和可靠性发挥重要作用，是一项不可或缺的研究内容。如果将 DEDS 中“监控”的概念引入到 EBS 的故障检测与诊断研究中，分析如何在 EBS

收稿日期：2014-02-26 修回日期：2014-09-06；
基金项目：国家自然科学基金项目(61070220, 61472003, 61272153, 61340003, 61402011)；国家"863"计划项目(2011AA060406)；安徽省高等学校自然科学研究重点项目(KJ2014A067)；
作者简介：方欢(1982-)，女，安徽池州人，博士，副教授，研究方向为 Petri 网理论及应用，离散事件动态系统的建模与分析，智能控制等。

中设置监控条件，使得所有的系统故障都能得到无二义性的诊断，这无疑是一个非常具有实际意义的课题。

在近几年的相关研究中，故障事件按照发生的影响情况一般被分为永久型故障、暂时型故障和控制故障[3]。一个事件 t 是“永久型”故障当且仅当 t 发生后其影响对于系统是永久的；一个事件 t 是“暂时型”故障当且仅当 t 发生后，其影响随着另外一些事件的发生而自动消除；而“控制型”故障一般被用来描述控制系统中的操作不当或操作失误。因此，在实际应用系统中，一个鲁棒性的控制策略至少应该保证能检测到被控制系统中可能发生的所有故障，进而对可能的故障类型进行区分，从而提高系统的可靠性和安全性，这一点在关键工业工程中得到最明显的体现，如柔性制造系统[4]和运输调度系统[5]等。由于暂时型故障的诊断与分析比较特别，本文只研究发生在离散事件驱动系统中的永久型故障和控制故障。

在离散事件驱动系统中，有诸多离散事件的发生，这些事件有些是可观测的，也有些是不可观测的，因此在部分可观测条件下[6-23]的故障检测与诊断显得尤为重要，相关研究也取得了一系列的研究成果：

(1) 在研究方法上，在已有的研究成果中，一般都将故障事件视为不可见事件，相应的研究方法有很多种：代数编码方法[14-15](Algebraic coding techniques)、网展开方法[16-17](net unfolding techniques)、自动机方法[3,8-9,18-19]、解释 Petri 网形式化方法[20](Interpreted Petri net formulations)、Petri 网方法[3,6-7,10-13,21-23]等。除此之外，将离散事件系统的故障检测与诊断内容视为 DEDS 监控理论[1-2]的一部分，来设计基于 R.W.理论的故障监控器也是一种解决途径[6-7]。在这些研究方法中，以基于模型(model-based)的自动机方法[3,8-9]和 Petri 网方法[3,6-7,10-16,21-23]最为普遍。

(2) 在研究目标上，已有的文献研究主要分为两类：故障检测器(故障诊断器)设计方法以及故障的可诊断性研究。

在故障检测器(故障诊断器)设计方法相关文献的研究中，一般都是针对不可见的故障事件，通过相应的设计方法，来检测故障的发生或者诊断故障的类型，尤其是在部分可观测的条件下，如何根据已经观测到的信息来推知系统当前的状态[5,10,12,24]：是正常运行、有故障发生、不确定状态，或者是进一步给出每种状态的可信值[11]等。另外，针对故障的可诊断性问题已有少数研究结论[18-19,24-27]。

通过分析已有的研究结论，可以发现在基于 Petri 网的部分可观系统故障诊断研究领域中，比较成熟的方法体系是利用 Petri 网的状态图来对故障可能发生的路径进行具体分析，从而得到相应的故障是否发生的结论。而对于部分可观系统的可观测元素的设计上，一般均考虑部分变迁可见[3,6,7,10-16,21-23]，或者部分库所可见部分变迁可见[11]的情况。不论可观测元素是库所还是变迁，确定可观测元素的基本准则都是使被控系统要么满足状态可见(state observability)，要么满足结构可见(structural observability)[28]。

综上所述，已有的研究针对部分可观系统中的故障检测与诊断有了比较系统的结论，然而还未有研究从故障检测与诊断的角度出发考虑部分可观系统的设计问题。将部分可观系统的故障检测与诊断理论与系统设计方法联系起来，即在部分可观系统中结合故障检测与诊断的理论，研究部分可观系统设计方法，以最少数目的监控器，使得所有故障都能准确诊断，这对于大规模应用的 EBS 具有广泛的实际应用价值，可以使得所设计的 EBS 系统在具备故障检测与诊断的基础上，完成相应的功能。

因此，本文从实现故障无二义性诊断的部分可观系统设计方法出发，首先，研究仅有部分库所可见的部分可观系统设计方法：通过设计可见库所集 S_O，使被控系统中的所有变迁都能唯一区分；其次，给出系统运行状态的诊断算法，通过可见库所集 S_O 的一步前向标识计算方法，完成系统状态诊断和故障变迁定位；再次，针对最优监控的问题，给

出满足故障诊断的最优监控库所集 S_{con} 确定算法，并证明本文所提出的部分可观系统设计方法确定的 S_O 满足最优监控条件；最后，通过一个实例说明所提出方法的有效性。

1 变迁可区分的系统可见库所集设计方法

在部分可观系统中，若仅有部分库所可见(将可见库所的集合记为 S_O)，则需要通过 S_O 来判断当前系统的运行状态，因此 S_O 的构造方法必须使得系统中每个变迁的发生能够唯一区分。在已有的研究方法中，变迁的可区分研究都是建立在系统初始标识 M_0 已知的情形下，在本文的研究中，将利用可见库所的标识在相邻两个时刻上的变化进行判断，而不需要借助系统的初始标识 M_0。

1.1 可见库所集 S_O 的确定算法

关联矩阵 $\boldsymbol{A}$ 除了传统的表示变迁和库所之间的关联关系的作用外，还能体现变迁引发所引起的各库所标识的变化情况。以图 1 为例，令 $V=[s_1,s_2,...,s_{14}]$，$D=[t_1,t_2,...,t_{13}]^T$，则图 1 对应的关联矩阵 $\boldsymbol{A}(D,V)$ 可以向量化表示为：

$$\boldsymbol{A}(t_1,:)=[-1,2,0,0,0,0,1,0,0,0,0,0,0,0];$$
$$\boldsymbol{A}(t_2,:)=[0,-1,1,0,0,0,0,0,0,0,0,0,0,0];\ldots$$
$$\boldsymbol{A}\ (t_{13},:)=[0,0,0,1,0,0,0,0,0,0,0,0,0,-1]。$$

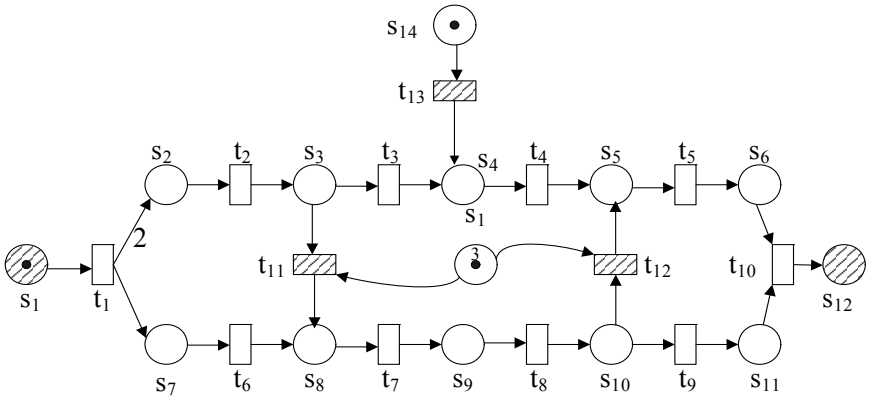

图 1 一个柔性制造系统的流程监控图

下面以关联矩阵 $\boldsymbol{A}$ 为基础，研究部分可观系统中可见库所集 S_O 的确定方法。

定义 2（变迁可区分）设 $N=(S,T;F,W)$ 是一个加权网，$\boldsymbol{A}$ 为 N 的关联矩阵，$S_O\subseteq S$ 为系统的可见库所集，τ_i 和 τ_{i+1} 为相邻的两个离散时间点，记在 τ_i 和 τ_{i+1} 时刻，S_O 的系统标识分别为 $M(S_O)_i$ 和 $M(S_O)_{i+1}$。则 N 是变迁可区分的，当且仅当任何一个变迁 $t\in T$ 都可以通过 $\boldsymbol{A}$，$M(S_O)_i$ 和 $M(S_O)_{i+1}$ 唯一确定。

定义 3 （变迁覆盖）设 $N=(S,T;F,W)$ 为一个加权 Petri 网，$\boldsymbol{A}$ 为相应的关联矩阵。若 $t_1,t_2\in T$，满足条件

(1) $(\bullet t_1\subseteq\bullet t_2)\wedge(t_1^\bullet\subseteq t_2^\bullet)$；

(2) $\forall s'\in\bullet t_1,\ \forall s\in t_1^\bullet: w(s',t_1)=w(s',t_2)$，$w(t_1,s)=w(t_2,s)$，则称 t_1 被 t_2 覆盖，记为 $\prec(t_1,t_2)$。

例 1 图 2 所示的加权 Petri 网 $N=(S,T;F,W)$ 中，$(\bullet t_3\subseteq\bullet t_2)\wedge(t_3^\bullet\subseteq t_2^\bullet)$ 并且满足条件 $\forall s'\in\bullet t_3,\forall s\in t_3^\bullet$：$w(s',t_3)=w(s',t_2)$，$w(t_3,s)=w(t_2,s)$，则根据定义 2，可得 t_3 被 t_2 覆盖，记为 $\prec(t_3,t_2)$。

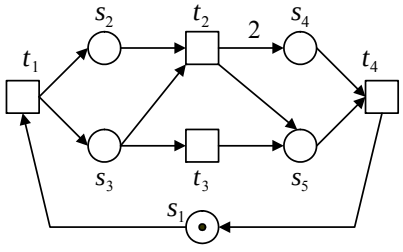

图 2 一个变迁覆盖的例子

定理 1 设 $N=(S,T;F,W)$ 为一个加权 Petri 网，$\boldsymbol{A}$ 为相应的关联矩阵，$t_1,t_2\in T$ 。则 $\prec(t_1,t_2)$ 当且仅当 $A(t_1,:)[-]A(t_2,:)=0_{1\times n}$。其中 $0_{1\times n}$ 为维数为 $1\times n$ 的 0 向量，向量 $\boldsymbol{X}=A(t_1,:)$ 和向量 $\boldsymbol{Y}=A(t_2,:)$ 的第 i 个分量分别记为 $\boldsymbol{X}[1,i]$ 和 $\boldsymbol{Y}[1,i]$，$[-]$ 为向量运算符，具体计算定义如下：

$$\boldsymbol{X}[1,i][-]\boldsymbol{Y}[1,i]=\begin{cases}0 & if\ \boldsymbol{X}[1,i]=\boldsymbol{Y}[1,i]\\ \boldsymbol{Y}[1,i] & else\end{cases}。$$

证明：根据定义 1 和定义 3，可以明显得出结论，证明从略。（证毕）

定义 4 （变迁区分表 $TD(T,\Delta)$ ）

一个被控系统 $N=(S,T;F,W)$ 的变迁区分表记为 $TD(T,\Delta)$：T 为系统中所有的变迁；Δ 为 $\otimes(s,A(t,s))$，其中 $S_c\subseteq S_O$ 为变迁 t 的输入或输出库所，$\boldsymbol{A}(t,s)$ 表示关联矩阵 $\boldsymbol{A}$ 中变迁 t 所在行与库所 s 所在列的交叉元素，$\otimes$ 表示逻辑“与”运算。

算法 1 系统可见库所集确定算法(Observable Places Determination algorithm，简记 OPD algorithm)

输入：被控系统的关联矩阵 $\boldsymbol{A}$，$n=|T|$;

输出：可见库所集 S_O ，变迁区分表 $TD(T,\Delta)$。

步骤：

1. $S_O=\varnothing$;

2. for i=1 to n do

{

2.1 选择 $\forall s\in t_i^\bullet$;

if $^\bullet s=\{t_i\}$ then

$\{S_O=S_O\cup\{s\}$;

$TD(t_i,\Delta)=(s,A(t_i,s))$； $\}$ (1)

// s 是 t_i 的唯一输出库所。

else 转 2.2。// t_i 与其他变迁共用部分输出库所。

2.2 选择 $\forall s'\in{}^\bullet t_i$ ；

if $s'^\bullet=\{t_i\}$ then

$\{S_O=S_O\cup\{s'\}$;

$TD(t_i,\Delta)=(s',A(t_i,s'))$； $\}$ (2)

// s' 是 t_i 的唯一输入库所。

else 转 2.3。// t_i 与其他变迁共用部分输入库所和输出库所。

2.3 对式(3)求解：

$\forall s'\in{}^\bullet t_i\wedge s\in t_i^\bullet\wedge s\cap(s'^\bullet-t_i)^\bullet$ (3)

if s,s' 有多个候选解，优先选择其中已经在 S_O 中的元素；

$\{S_O=S_O\cup\{s,s'\}$;

$TD(t_i,\Delta)=(s,A(t_i,s))\otimes(s',A(t_i,s'))$ ；$\}$

3. 算法结束，输出 S_O 和 $TD(T,\Delta)$。

其中算法 1 的步骤 2.1，2.2 和 2.3 中的 3 个判断条件 $^\bullet s=\{t_i\}$，$s'^\bullet=\{t_i\}$ 以及式(3)都可以通过关联矩阵 $\boldsymbol{A}$ 来进行计算，计算方法如下所示：

(1) $^\bullet s=\{t_i\}$ 的判断只需要查找 $A(t_i,:)$ 是否有唯一的正整数元素 $A(t_i,s)$；

(2) $s'^\bullet=\{t_i\}$ 的判断只需要查找 $A(t_i,:)$ 是否有唯一的负整数元素 $A(t_i,s')$；

(3) 公式(3) $\forall s'\in{}^\bullet t_i\wedge s\in t_i^\bullet\wedge s\cap(s'^\bullet-t_i)^\bullet$ 的求解要稍微复杂一些。令

$bs'=\{s_{k_p}\mid A(t_i,s_{k_p})<0\}$；

$bs=\{s_{k_q}\mid A(t_i,s_{k_q})>0\}$；

$bt=\{t_d\mid\forall s\in bs':(A(t_d,s)<0\wedge t_d\neq t_i)\}$；

$\forall t\in bt:bs'n=\{s_{c_l}\mid A(t_d,s_{c_l})>0\}$，

则公式(3)最后简化为 $bs\cap bs'n$。

由此步骤 2.1，2.2 和 2.3 的 3 个判断条件都通过关联矩阵 $\boldsymbol{A}$ 进行了求解。

1.2 算法分析

算法 1 给出了可见库所集和变迁区分表的确定算法，针对这个算法，有两点内容至关重要：算法解的存在性以及算法的时间复杂度问题。

首先考虑第一个问题，算法解的存在性问题。

定理 2 设 $N=(S,T;F,W)$ 是一个加权 Petri 网，$\boldsymbol{A}$ 为相应的关联矩阵，S_O 和 $TD(T,\Delta)$ 是根据算法 1 得到的可见库所集和变迁区分表。S_O 是存在的当且仅当 $\forall t_i\in T$ 不存在 $\forall t_l\in T$ 满足条件 $\prec(t_i,t_l)$。

证明：充分性。若对 $\forall t_i\in T$ 不存在 $t_l\in T$ 满足条件 $\prec(t_i,t_l)$，则根据变迁覆盖的定义(定义 3)可知：要么 $({}^\bullet t_i\subseteq{}^\bullet t_l)\wedge(t_i^\bullet\subseteq t_l^\bullet)$ 不能被满足，要么

$(w(s',t_i)=w(s',t_l))\wedge(w(t_i,s)=w(t_l,s))$ 不被满足。不论是哪一个条件不被满足，都可以找到两个二元组 $(s,A(t_i,s))$，$(s',A(t_i,s'))$ 来唯一标志变迁 t_i 的发生，这样算法 1 的步骤 2.3 就可以有解，而步骤 2.1 和 2.2 是显然有解的。由此，充分性得到证明。

必要性。若 S_O 是根据算法 1 得到的可见库所集，并且 S_O 是存在的，则表示算法 1 的 2.1，2.2 和 2.3 都未出现无解的情况，步骤 2.1 和 2.2 的解存在性很明显，而解 2.3 的解存在性的保证在于公式(3)的满足上。同样使用反证法证明。假设存在一个变迁 $t_l\in T$ 满足条件 $\prec(t_i,t_l)$，则根据定义 3 可知：$({}^\bullet t_i\subseteq{}^\bullet t_l)\wedge(t_i^\bullet\subseteq t_l^\bullet)$ 和 $(w(s',t_i)=w(s',t_l))\wedge(w(t_i,s)=w(t_l,s))$ 成立，而根据算法 1 选择的两个二元组 $(s,A(t_i,s))$，$(s',A(t_i,s'))$ 中的库所 s 和 s' 必然满足条件 $(s'\in{}^\bullet t_i)\wedge(s\in t_i^\bullet)$，由于 $\prec(t_i,t_j)$，则 $(s'\in{}^\bullet t_l)\wedge(s\in t_l^\bullet)$，从而 $s\cap(s'^\bullet-t_i)^\bullet=s\cap t_l^\bullet=s\neq\varnothing$，与条件(3)矛盾。由此假设不可能成立，从而必要性得到证明。

由此，定理得证。（证毕）

定理 3 设 $N=(S,T;F,W)$ 为一个加权 Petri 网，$\boldsymbol{A}$ 为相应的关联矩阵，S_O 和 $TD(T,\Delta)$ 是根据算法 1 得到的可见库所集和变迁区分表。则 N 中的所有变迁都是可区分的，当且仅当 $\forall t_i$ 不存在 t_j 满足条件 $\forall t_i\in T\wedge\forall t_j\in T\wedge t_i\neq t_j\nprec(t_i,t_j)$。

证明：首先证明必要性。

针对任意一个变迁 $t_i\in T$，只可能存在以下 3 种可能性：

(1) t_i 具有唯一的输出库所 s，即满足 $|t_i^\bullet|=1$。此时可选定库所 s 作为 t_i 的监控库所，使用二元组 $(s,A(t_i,s))$ 来标志 t_i 的发生，由此可知步骤 2.1 是正确的；

(2) t_i 具有唯一的输入库所 s'，除了满足 $|{}^\bullet t_i|=1$ 以外，还满足 $s'^\bullet=\{t_i\}$，这两个条件说明 t_i 不与其他任何变迁 t 共用输入库所 s'。此时可选定 s' 作为 t_i 的监控库所，使用二元组 $(s',A(t_i,s'))$ 来标志 t_i 的发生，由此可知步骤 2.2 是正确的；

(3) t_i 与一组变迁 $t_{p_1},t_{p_2},...,t_{p_k}$ 共用部分输入库所和输出库所，即满足条件 $\forall j\in\{1,2,...,k\}$：$(({t_{p_j}}^\bullet\cap t_i^\bullet\neq\varnothing)\vee({}^\bullet{t_{p_j}}\cap{}^\bullet t_i\neq\varnothing))$。此时可以选择 2 个二元组 $(s,A(t_i,s))$，$(s',A(t_i,s'))$ 来唯一标识 t_i 的发生，其中 s,s' 满足公式(4)。

$$(s'\in{}^\bullet t_i)\wedge(s\in t_i^\bullet)\wedge(s\cap(s'^\bullet-t_i)^\bullet)=\varnothing \quad (4)$$

下面证明公式(3)中确定的监控库所 s,s' 对 t_i 标识的唯一性。用反证法证明。

假设使用监控库所 s,s' 以及相应的二元组 $(s,A(t_i,s))$，$(s',A(t_i,s'))$ 除了可以标志 t_i 的发生，还可以至少标志另一个变迁 t_l 的发生，则 $(\{s'\}\subseteq{}^\bullet t_l)\wedge(\{s\}\subseteq t_i^\bullet)$，因此 $s\cap(s'^\bullet-t_i)^\bullet=s\cap t_l^\bullet=s\neq\varnothing$，这与公式(3)相矛盾，因此使用公式(3)可以选择唯一标志 t_i 发生的两个二元组 $(s,A(t_i,s))$，$(s',A(t_i,s'))$。

定理充分性是显然的，在此省略。

由此，定理得证。（证毕）

第二个问题是算法的时间复杂度问题。OPD 算法(算法 1)中可见库所集 S_O 和变迁区分表 $TD(T,\Delta)$ 是基于关联矩阵 $\boldsymbol{A}$ 的结构算法，实现过程是可以通过对关联矩阵 $\boldsymbol{A}$ 的搜索完成的，复杂度是 $O(n^2)$ 的，因此整个算法 1 的时间复杂度为多项式级的。

2 故障无二义诊断下的部分可观系统设计方法

由于一般离散事件驱动系统的动态运行过程十分复杂，为了研究保证无二义性故障诊断基础下的系统设计方法，就必须实现两点内容：首先，在系统设计完成后，系统中各类故障的诊断算法必须是在线的，这就要求算法不能利用 Petri 网系统的整体可达标识图信息，因为整体可达标识图的构造过程是指数复杂度的，并且基于整体状态可达图的故障诊断算法一定是离线的；其次，诊断算法的输出结果必须实现无二义性。其中无二义性的实现也是最困难的，已有文献的结果目前只能实现如下结果：根据当前系统状态判断当前状态的性质，譬如当前状态是肯定无故障发生的(记为 N)、肯定是有

故障发生的(记为 F)和不确定是否有故障发生(记为 U)[3,10,12]；或者更进一步，当以上三种状态通过引入信念函数[11]或模糊函数[29]，将其判定结果转换为实数区间 $[0, 1]$ 内的实数。为了克服诊断结果不确定性的缺点，期望通过系统设计方法的改进来弥补这些缺陷和不足。

2.1 故障无二义性诊断下的部分可观系统设计方法

根据算法 1，可以在所得到的变迁区分表 $TD(T,\Delta)$ 中得到故障变迁 t_f ($t_f \in T_F \subset T$)的变迁区分条件，不妨假定 $\forall t_f \in T_F$ ，$\underset{s\in S_C}{\otimes}(s, A(t_f, s))$ 为对应的变迁区分条件，则称输入输出库所集 S_C 为变迁 t_f 的监控库所集。

下面给出保证故障无二义性诊断下的部分可观系统设计方法步骤。

Step 1: 将被控系统经过 Petri 网的形式化建模得到一个加权 Petri 网 $N=(S,T;F,W)$；

Step 2：计算关联矩阵 $\boldsymbol{A}$ 和故障变迁集合 T_F；

Step 3：通过 OPD 算法（算法 1）计算 S_O 和 $TD(T,\Delta)$；

Step 4：将 N 中故障变迁 $\forall t_f \in T_F$ 在 $TD(T,\Delta)$ 中的变迁区分条件 $\underset{s\in S_C}{\otimes}(s, A(t_f, s))$ 作为 t_f 的监控条件，而 $\underset{s\in S_C}{\otimes}(s, A(t_f, s))$ 中的 S_C 为变迁 t_f 的监控库所；

Step 5: 将故障变迁 T_F 表示成带阴影的矩形框，将可见库所 S_O 表示成带阴影的圆形，则带变迁区分表 $TD(T,\Delta)$ 的部分可观系统记为 $Pos(N,T_F,S_O,TD(T,\Delta))$。

2.2 系统运行状态的一步前向标识计算

根据设计得到的部分可观系统 $Pos(N,T_F,S_O,TD(T,\Delta))$，以下给出系统运行状态诊断的算法(算法 2)，系统的运行状态用 $(status, occ_{t_f})$ 表示，其中 $status=\{Nor, Fau\}$：Nor 表示无故障发生，Fau 表示有故障发生且故障为 occ_{t_f} 中的故障；当系统状态为 Nor 时 $occ_{t_f}=\varnothing$。

算法 2 系统运行状态诊断的算法（System Operating State Diagnosis algorithm，简记为 SOSD 算法）

输入：被控系统 $Pos(N,T_F,S_O,TD(T,\Delta))$，可见库所 S_O 的系统当前标识 M_{cur} 、当前标识的前一个标识 M_{pre}；

输入：系统的运行状态 $(status, occ_{t_f})$。

步骤：

1. $occ_{t_f}=\varnothing$；

2. $subM=M_{cur}-M_{pre}$；

3. 查找 $TD(T,\Delta)$ 每一行 $TD(t_i,\Delta)=(s,A(t_i,s))$ 或 $TD(t_i,\Delta)=(s,A(t_i,s))\ \ \otimes(s',A(t_i,s'))$，不妨假设 s 或 s' 分别是 N 的第 k 个或第 k' 个库所：

if $subM(k)=A(t_i,s)$ 或 $subM(k)=A(t_i,s)$
$\wedge subM(k')=A(t_i,s')$ then
{if $t_i \in T_F$ then
$occ_{t_f}=occ_{t_f}\cup\{t_i\}$， $status=Fau$；
}
else
{ $occ_{t_f}=\varnothing$， $status=Nor$； }

4. 算法结束，返回 $(status, occ_{t_f})$。

从算法 2 可以看出，根据系统的两个可见库所标识的减法（当前状态及其之前的状态），因此被称为一步前向标识计算方法。结合算法 1 得到的变迁区分表和系统的关联矩阵就可以对系统的状态进行识别，识别后的系统运行状态只有 2 种 Nor 或 Fau，并且还可以判定具体的故障类型。由此，通过系统运行状态的一步前向标识计算，就可以对系统当前的运行状态及其发生的故障类型进行准确判断，因此实现了故障的无二义性诊断。

3 系统可见库所的最优监控问题

定义 5 (最优监控库所集)一个被控系统的加权 Petri 网 $N=(S,T;F,W)$，$\boldsymbol{A}$ 为 N 的关联矩阵，故障变迁 $T_F \subseteq T$。若 $\exists S_{con} \subseteq S$ 满足条件：

(1) $\forall t_f \in T_F$ 都有唯一的监控条件 $\lambda(t_f, S_C)$

（$S_C \subseteq S_{con}$）；

(2) $\bigcup_{\forall t_f \in T_F, \exists \lambda(t_f, S_C)} S_C = S_{con}$；

(3) $S_{con} = S_O \cap (\bigcup_{\forall t_f \in T_F} t_f^{\bullet} \cup {}^{\bullet}t_f)$；

则称当前的可见库所集 S_{con} 是满足故障监控条件的最优监控库所集。

定理 4 设 $N=(S,T;F,W)$ 是一个 Petri 网，$T_F \subseteq T$ 是表示故障事件发生的变迁集，S_O 是根据算法 1 得到的可见库所集。若 $\forall t_f \in T_F$ 不存在 $t_l \in T$ 满足条件 $\prec(t_f, t_l)$，则 S_O 是针对故障事件监控的最优监控可见库所集。

证明：从算法 1 的步骤 2.1 和 2.2 可以看出，针对比较简单的网结构，如代表故障发生的变迁具有唯一的输出库所或输入库所，则只利用一个可见库所实现对变迁的无二义性鉴别，而一些具有复杂结构的故障事件变迁（与其他变迁共用部分输入库所和输出库所），则分别利用一个输入库所 s'、一个输出库所 s 以及相应的标识变化组成 2 个二元组“与”运算 $(s', subM(s')) \otimes (s, subM(s))$ 来实现监控，并且在选择 s' 和 s 的时候，若存在多个选择，则以优先选择已经在 S_O 中的元素为先（步骤 2.3），因此步骤 2.1~2.3 都达到了最少库所监控的条件。由于故障变迁集 T_F 满足条件 $T_F \subseteq T$，因此 S_O 是针对故障事件监控的最优监控可见库所集，而 $\exists S_{con} \subseteq S$ 是满足故障变迁 T_F 的最优监控集。

下面使用反证法来证明算法 1 得到的 S_O 是最优监控可见库所集。

假设通过算法 1 可以得到另一个不同于 S_O 的监控库所集 S_O'，满足条件 $|S_O| \geq |S_O'|$。则至少存在一个 $t_{f_1} \in T_F \subseteq T$ 满足 $|S_{cf_1}| > |S_{cf_1}'|$，其中 S_{cf_1}, S_{cf_1}' 分别为 t_{f_1} 在 S_O 和 S_O' 中的监控库所集。由于在系统中 $\forall t_f \in T_F$ 不存在 $t_l \in T$ 满足条件 $\prec(t_f, t_l)$，且 $|S_{cf_1}|$ 和 $|S_{cf_1}'|$ 不能为 0，则唯一符合的情况为 $|S_{cf_1}| = 2$，而 $|S_{cf_1}'| = 1$。$|S_{cf_1}'| = 1$ 时表示 t_{f_1} 具有唯一的输入库所或输出库所，根据算法 1 中的步骤 2.3，$|S_{cf_1}| = 2$ 表示 t_{f_1} 具有非唯一的输出库所或非唯一的输入库所，因此两者相矛盾，从而假设不成立。

由此，定理得证。(证毕)

4 案例应用及分析

图 3 是一个柔性制造系统，由于其描述的系统具有典型性，因此被很多研究者作为典型案例进行研究[12,31-33]，从各个角度进行分析，如故障诊断器的构造方法[12,32-33]、禁止状态避免[31]等等，是一个具有代表性的复杂系统控制问题。在这些研究成果中，将变迁 $t_1 \sim t_{12}$ 视为可见事件，$\varepsilon_{13} \sim \varepsilon_{24}$ 被视为正常事件发生的不可见事件，代表故障事件发生的变迁为 ε_{25} 和 ε_{26}，分别代表两种类型的故障 $T_f^1 = \{\varepsilon_{25}\}$ 和 $T_f^2 = \{\varepsilon_{26}\}$。

在本文中，我们对这个案例进行一些修改，即假定图 3 中的所有变迁都不可见。与文献[12,32]类似，为了保证该 FMS 的正常运行，需要满足三个互斥条件约束(a)~(c)：

$$\begin{cases} \sum_{i=2}^{9} m(s_i) \leqslant 8 & \text{(a)} \\ \sum_{i=15}^{19} m(s_i) \leqslant 8 & \text{(b)} \\ \sum_{i=2}^{9} m(s_i) + \sum_{i=15}^{19} m(s_i) \leqslant 9 & \text{(c)} \end{cases}$$

为了实现这 3 个约束条件，可以通过 3 个监视库所 s_{36}, s_{37}, s_{38}，它们的初始标志分别设置 8, 8, 9。

下面使用本文所提出的部分可观系统设计方法，对图 3 进行可见库所集和故障监控库所集的设计，结果如表 1 所示。其中库所 s_{36}, s_{37}, s_{38} 必须存在于 S_O，这样就能保证约束条件(a) ~(c)满足，除了这三个库所，其余 S_O 中的库所是为了实现变迁 $t_1 \sim t_{12}$ 以及故障变迁 ε_{25} 和 ε_{26} 可区分而设置的。

根据表 1 得到的结果，为了实现图 3 系统的故障检测与诊断，只需要将系统中的故障监控库所 S_C 设置为 $S_C = \{s_4, s_{12}, s_{15}, s_{23}\}$。通过 S_C 的库所标识变化结合表 1 的监控条件，就可以实现该部分可观系统的设故障检测与诊断。

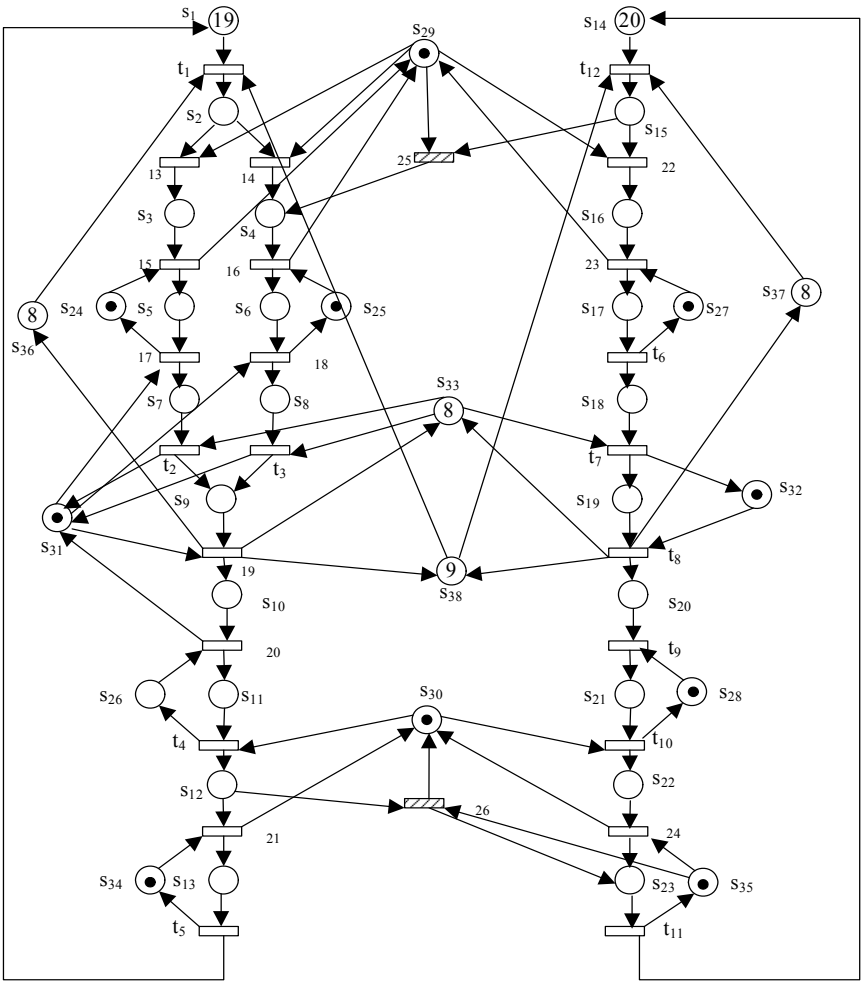

图 3 一个典型的柔性制造系统 Petri 网模型

表 1 各类变迁的监控条件

变迁	可见监控库所 S_O	监控条件 $\underset{s\in S_O}{\otimes}(S,A(t,s))$
ε_{25}	$\{s_4,s_{15}\}$	$(s_{15},-1)\otimes(s_4,1)$
ε_{26}	$\{s_{12},s_{23}\}$	$(s_{12},-1)\otimes(s_{23},1)$
t_1	$\{s_2,s_{36}\}$	$(s_{36},-1)\otimes(s_2,1)$
t_2	$\{s_7,s_{31}\}$	$(s_7,-1)\otimes(s_{31},1)$
t_3	$\{s_8,s_{31}\}$	$(s_8,-1)\otimes(s_{31},1)$
t_4	$\{s_{12},s_{30}\}$	$(s_{30},-1)\otimes(s_{12},1)$
t_5	$\{s_{13},s_{34}\}$	$(s_{13},-1)\otimes(s_{34},1)$
t_6	$\{s_{17},s_{27}\}$	$(s_{17},-1)\otimes(s_{27},1)$
t_7	$\{s_{32},s_{33}\}$	$(s_{33},-1)\otimes(s_{32},1)$
t_8	$\{s_{32},s_{33}\}$	$(s_{33},1)\otimes(s_{32},-1)$
t_9	$\{s_{21},s_{28}\}$	$(s_{28},-1)\otimes(s_{21},1)$
t_{10}	$\{s_{30},s_{28}\}$	$(s_{30},-1)\otimes(s_{28},1)$
t_{11}	$\{s_{23},s_{35}\}$	$(s_{23},-1)\otimes(s_{35},1)$
t_{12}	$\{s_{14},s_{15}\}$	$(s_{14},-1)\otimes(s_{15},1)$
S_O	$\{s_2,s_4,s_7,s_8,s_{12},s_{13},s_{14},s_{15},s_{17},s_{21},s_{23},s_{27},s_{28},s_{30},$ $s_{31},s_{32},s_{33},s_{34},s_{35},s_{36},s_{37},s_{38}\}$	

除了图 3 给出的 FMS 的案例应用，还可以将本文所提出的部分可观系统的设计方法应用到具体的实际系统中，例如该方法可以直接应用在矿井机车调度系统中，用来分析系统中故障的监控传感器节点的安放位置，从而保证企业机车运输系统的安全性和可监控性，由于篇幅所限，我们将另文做详细阐述。将本文所提出的部分可观系统设计方法应用到实际的 EBS 中，可以为实际系统的安全运行提供坚实的理论支撑。

与文献[28]所研究的离散事件系统的最优库所(变迁)感应器选择问题(OPSS(OTSS))相比，本文所提出的方法主要有 3 点不同：

(1) 文献[28]考虑了 DES 中库所（变迁）选择向量 V 以及变迁标记函数 L ($L:T\to\Sigma\cup\{\varepsilon\}$)满足条件：在系统结构可观条件下使得 $\|V\|+\Sigma$ 最优；

而本文所提出的方法仅考虑部分部分库所可见，而不存在变迁标记函数的问题，也可以理解为所有变迁标记均不可见；

(2) 文献[28]提出的研究结论均满足前提条件：系统中不存在两个行为相同的变迁；而本文所提出的可见库所设计方法前提条件为：系统中任意两个变迁的关系不是本文所提出的变迁覆盖关系；

(3) 文献[28]以及已有研究的大部分研究结论[3, 6, 7, 10-16, 21-23]都要求系统初始标识 M_0 已知，通过 M_0 以及观测到的变迁序列还推算系统的运行状态，而本文的研究不需要借助初始标识 M_0，只需要对可见库所集 S_O 在相邻 2 个时刻下的标识进行简单计算，就可以得出系统的运行状态，因此可以减少计算量。

5 结论

针对离散事件驱动系统，提出保证所有故障都能无二义诊断条件下的部分可观系统设计方法，给出故障诊断条件和可见库所集的设计算法，并证明提出的部分库所可见系统设计方法满足最优监控条件。

给出的仅有部分库所可见的部分可观系统设计方法，针对一般的离散事件驱动系统，当系统中可见库所 S_O 的数量固定且覆盖范围有限的情况，本文提出的利用变迁区分表 $TD(T,\Delta)$ 的一步前向标识诊断方法就不再适用，在以后的研究中将继续对这些内容进行继续探讨。除此之外，在今后的研究中，可以将本论文提出的部分可观系统设计方法应用于复杂系统的设计与监控中，得出更加实用的理论结果。

附录

一般地，令 $N=(S,T;F)$ 为一个 Petri 网，$N=(S,T;F,W)$ 为一个加权 Petri 网，并称 $\Sigma=(S,T;F,W,M)$ 为一个加权 Petri 网系统。其中 S 为有限库所集，T 为有限变迁集，$F\subseteq(S\times T)\cup(T\times S)$ 是有向弧集，W 为弧权函数，M 是系统的库所标识[29]。

给定节点 $x\in S\cup T$，记

${}^\bullet x=\{y\in S\cup T\mid(y,x)\in F\}$，$x^\bullet=\{y\in S\cup T\mid(x,y)\in F\}$。

给定标识 M，称变迁 t 在标识 M 下是使能的，当且仅当公式(1)成立。

$$\forall s\in {}^\bullet t: M(s)\geqslant W(s,t) \tag{1}$$

在 M 下使能变迁 t 可以引发，引发 t 后产生新的标识 M'，记为 $M[t>M'$。用 $R(PN,M_0)$ 表示 Petri 网 PN 从初始状态 M_0 可以到达的所有状态的集合。

定义 1（关联矩阵[29]）设 $N=(S,T;F,W)$ 是一个加权 Petri 网，$S=\{s_1,s_2,...,s_m\}$，$T=\{t_1,t_2,...,t_n\}$，则 Petri 网 N 的结构可以用一个 n 行 m 列矩阵 $A=[a_{ij}]_{n\times m}$ 来表示，其中 $a_{ij}=a_{ij}^+-a_{ij}^-$，$i\in\{1,2,...,n\}$，$j\in\{1,2,...,m\}$，

$$a_{ij}^+=\begin{cases}w(t_i,s_j) & if\ (t_i,s_j)\in F\\ 0 & else\end{cases},\quad a_{ij}^-=\begin{cases}w(s_j,t_i) & if\ (s_j,t_i)\in F\\ 0 & else\end{cases},$$

则称 $\boldsymbol{A}$ 为 Σ 的关联矩阵。

在本文的建模过程中，库所用圆圈表示，变迁使用矩形方框表示，表示故障发生的变迁 t_f（$t_f\in T_F\subset T$）用加阴影的矩形框表示。在本文的研究过程中，不论是永久型故障还是控制故障，均使用加阴影的矩形框表示故障变迁。为了加以区分，将被控系统中可见的库所集记为 $S_O=\{s_{o1},s_{o2},...,s_{op}\}$（$p\geqslant 1$），用加阴影的圆圈表示。例如，图 1 为文献[10]所研究的一个柔性加工系统的流程监控图，其中 $S_O=\{s_1,s_{12}\}$ 为当前可见库所集合，t_{11},t_{12},t_{13} 代表 3 种不同类型的故障事件。

参考文献：

[1] P J Ramadge, W M Wonham. Supervisory Control of a Class of Discrete Event Processes [J]. Siam Journal on Control and Optimization (S0363-0129), 1987, 25(1): 206-230.

[2] P J Ramadge, W M Wonham. Modular Feedback Logic for Discrete Event System [J]. Siam Journal on Control and Optimization (S0363-0129), 1987, 25(5): 1202-1218.

[3] M Dotoli, M P Fanti, A M Mangini, W Ulkovich. On-line Fault Detection in Discrete Event Systems by Petri Nets and Integer Linear Programming [J]. Automatica (S0005-1098), 2009, 45(11): 2665-2672.

[4] A V Ramesh, D W Twigg, U R Sandadi, *et al*. Reliability Analysis of Systems with Operation-time Management [J]. IEEE Transactions on Reliability (S0018-9529), 2002, 51(1): 39-48.

[5] J Jorge, R K Boel. A continuous Petri Net Approach for Model Predictive Control of Traffic Systems [J]. IEEE Transactions on Systems, Man, and Cybernetics, A (S1083-4427), 2010, 40(4): 686-697.

[6] 古天龙, 周衿畅, 周春晖. 离散事件系统的一种 N 步在线监控策略 [J]. 自动化学报, 1997, 23(3): 404-407.

[7] 王飞, 胡奇英. 离散事件系统的混合分散监控 [J]. 控制理论与应用, 2005, 22(2): 277-280.

[8] S Z Hashtrudi, R H Kwong, W M Wonham. Fault

Detection in Discrete-event Systems: Framework and Model Reduction [J]. IEEE Transaction on Automatic Control (S0018-9286), 2003, 48(7): 1199-1212.

[9] D Lefebvre, C Delherm. Diagnosis of DES with Petri Net Models [J]. IEEE Transactions on Automation Science and Engineering (S1545-5955), 2007, 4(1): 114-118.

[10] M P Cabasino, A Giua, C Seatzu. Fault detection for discrete event systems using Petri nets with unobservable transitions [J]. Automatica (S0005-1098), 2010, 46(9): 1531-1539.

[11] Yu Ru, C N Hadjicostis. Fault Detection in Discrete Event Systems Modeled by Partially Observed Petri Nets [J]. Discrete Event Dynamic Systems (S1573-7594), 2009, 19(4): 551-575.

[12] C Mahulea, C Seatzu, M P Cabasino, *et al.* Fault Diagnosis of Discrete-event Systems Using Continuous Petri Nets [J]. IEEE Transactions on Systems, Man and Cybernetics A (S1083-4427), 2012, 42(4): 970-984.

[13] S Jiang, R Kumar. Failure Diagnosis of Discrete-event Systems with Linear-time Temporal Logical Specifications [J]. IEEE Transaction on Automatic and Control (S0018-9286), 2004, 49(6): 934-945.

[14] C. N. Hadjicostis, G. C Verghese. Monitoring Discrete Event Systems Using Petri Net Embeddings [J]. Lecture Notes in Computer Science, 1999, 1639(1): 188-208.

[15] Y Wu, C N Hadjicostis. Algebraic Approaches for Fault Identification in Discrete-event Systems [J]. IEEE Transactions on Automatic Control (S0018-9286), 2005, 50(12): 2048-2055.

[16] A Aghasaryan, E Fabre, A Benveniste, *et al*. Fault Detection and Diagnosis in Distributed Systems: An Approach by Partially Stochastic Petri Nets [J]. Discrete Event Dynamic System: Theory and Applications (S0924-6703), 1998, 8(2): 203-231.

[17] A Benveniste, E Fabre, S Harr, *et al*. Diagnosis of Asynchronous Discrete-event Systems: A Net Unfolding Approach [J]. IEEE Transactions on Automatic Control (S0018-9286), 2003, 48(5): 714-727.

[18] M Sampath, R Sengupta, S Lafortune, *et al*. Diagnosability of Discrete Event Systems [J]. IEEE Transactions on Automatic Control (S0018-9286), 1995, 40(9): 1555-1575.

[19] S H Zad, R H Kwong, W M Wonham. Diagnosis in Discrete-event Systems: Incorporating Timing Information [J]. IEEE Transactions on Automatic Control (S0018-9286), 2005, 50(7): 1010-1015.

[20] A Ramírez-Treviño, E Ruiz-Beltrăn, I Rivera-Rangel, *et al*. Online Fault Diagnosis of Discrete Event Systems: A Petri Net-based Approach [J]. IEEE Transactions on Automation Science and Engineering (S1545-5955), 2007, 4(1): 31-39.

[21] D Lefebvre, E Leclercq. Stochastic Petri Net Identification for the Fault Detection and Isolation of Discrete Event Systems [J]. IEEE Transactions on System, Man and Cybernetic A (S1083-4427), 2011, 41(2): 213-225.

[22] G Jiroveanu, R K Boel. A Distributed Approach for Fault Detection and Diagnosis based on Time Petri Nets [J]. Mathematics and Computers in Simulation (S1998-0159), 2006, 70(3): 287-313.

[23] S Genc, S Lafortune. Distributed Diagnosis of Place-bordered Petri Nets [J]. IEEE Transactions on Automation Science and Engineering (S1545-5955), 2007, 4(2): 206-219.

[24] F Basil, P Chiacchio, G De Tommasi. On K-diagnosability of Petri Nets via Integer Linear Programming [J]. Automatica (S0005-1098), 2012, 48(9): 2047-2058.

[25] J Lunze, J Schroder. State Observation and Diagnosis of Discrete-event Systems Described by Stochastic Automata [J]. Discrete Event Dynamic Systems (S1573-7594), 2001, 11(4): 319-369.

[26] R Debouk, S Lafortune, D Teneketzis. Coordinated decentralized protocols for failure diagnosis of discrete event systems [J]. Discrete Event Dynamic Systems (S1573-7594), 2000, 10(1): 33-86.

[27] A Paoli, S Lafortune. Safe Diagnosability for Fault-tolerant Supervision of Discrete-event Systems [J]. Automatica (S0005-1098), 2005, 41(8): 1335-1347.

[28] Yu Ru, C N Hadjicostis. Sensor Selection for Structural Observability in Discrete Event Systems Modeled by Petri Nets [J]. IEEE Transactions on Automatic Control (S0018-9286), 2010, 55(8): 1751-1764.

[29] 吴哲辉. Petri 网导论 [M]. 北京: 机械工业出版社, 2006: 1-108.

[30] J Sun, S Y Qin, Y H Song. Fault Diagnosis of Electric Power Systems based on Fuzzy Petri Nets [J]. IEEE Transactions on Power Systems (S0885-8950), 2004, 19(4): 2053-2059.

[31] M Zhou, F Dicesare. Petri Net Synthesis for Discrete Event Control of Manufacturing System [M]. Norwell, MA, USA: Kluwer, 1993.

[32] F Basile, A Giua, C Seatzu. Petri Net Control Using Event Observers and Timing Information [C]// Proceedings of 41th IEEE Conference on Decision Control (S1573-2878). USA: IEEE, 2002: 787-792.

[33] M P Cabasino, A Giua, M Pocci, C Seatzu. Discrete Event Diagnosis Using Labeled Petri Nets: An Application to Manufacturing Systems [J]. Control Engineering and Practice (S0967-0661), 2011, 19(9): 989-1001.