

Diagnose de falhas para uma classe de Redes de Petri[★]

Ana C. Bonafin^{*} Felipe G. Cabral^{*} Marcos V. Moreira^{**}

^{*} Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Campus Trindade, Florianópolis, 88.040-900, (e-mail: ana.bonafin@posgrad.ufsc.br; felipe.gomes.cabral@ufsc.br).

^{**} COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, Cidade universitária, Ilha do Fundão, Rio de Janeiro, 21.945-970, (e-mail: moreira.mv@poli.ufrj.br)

Abstract: In this paper, an online fault diagnosis method for Discrete-Event Systems (DESs) modeled by a class of labeled Petri nets (LPNs) is proposed. It is assumed that some transitions of the Petri net are unobservable, including the fault transitions. The diagnosis method is based on the construction of an LPN from the fault-free system behavior model, called observable behavior Petri net (OBPN), whose transitions are all observable, and whose generated language is equal to the observable language of the fault-free system model. The number of places and transitions of the OBPN is always smaller than or equal to the number of places and transitions of the Petri net system model.

Resumo: Neste artigo, um método de diagnóstico de falhas para Sistemas a Eventos Discretos (SEDs) modelados por uma classe de redes de Petri rotuladas (RPRs) é proposto. É suposto que algumas transições da rede de Petri são não observáveis, incluindo as transições de falha. O método de diagnóstico é baseado na construção de uma RPR a partir do modelo do comportamento livre de falha do sistema, chamada de rede de Petri do comportamento observável (RPCO), cujas transições são todas observáveis, e cuja linguagem gerada é igual à linguagem observada do modelo livre de falha do sistema. O número de lugares e transições da RPCO é sempre menor ou igual ao número de lugares e transições do modelo do sistema em redes de Petri.

Keywords: Fault diagnosis; Petri nets; Discrete Event Systems.

Palavras-chaves: Diagnose de falhas; Redes de Petri; Sistemas a Eventos Discretos.

1. INTRODUÇÃO

Sistemas automatizados estão sujeitos à ocorrência de falhas que podem alterar seu comportamento nominal, podendo causar danos a equipamentos e oferecer riscos a operadores. Portanto, é necessário implementar um sistema de diagnóstico de falhas automático que detecte e isole a ocorrência da falha para prevenir danos ao sistema. Diversos trabalhos têm sido apresentados sobre diagnóstico de falhas em Sistemas a Eventos Discretos (SEDs) (Sampath et al., 1995; Cabral et al., 2015), sendo que as abordagens mais comuns consideram sistemas modelados por autômatos (Sampath et al., 1995; Santoro et al., 2017; Veras et al., 2021), ou redes de Petri (RPs) (Ran et al., 2017; Miyagi and Riascos, 2006; Giua et al., 2007).

Em Sampath et al. (1995), um diagnosticador para SEDs modelados por autômatos é proposto e uma condição necessária e suficiente para verificar a diagnosticabilidade da linguagem do sistema, *i.e.*, a capacidade de detectar a ocorrência do evento de falha após um número limitado de observações de eventos, é apresentada. O diagnosticador

apresentado em Sampath et al. (1995) é baseado no cálculo de um autômato observador do modelo do sistema, cujo número de estados pode crescer exponencialmente com a cardinalidade do conjunto de estados do sistema. Desde então, métodos que evitam o uso de observadores para verificação da diagnosticabilidade, e possuem complexidade polinomial com a cardinalidade do conjunto de estados da planta, têm sido propostos (Qiu and Kumar, 2006; Moreira et al., 2011; Moreira et al., 2016).

Com o objetivo de evitar a construção de um observador para o diagnóstico, um diagnosticador rede de Petri (DRP) para SEDs modelados por autômatos é proposto em Cabral et al. (2015). O DRP é capaz de estimar os estados do comportamento livre de falha do modelo do sistema online e pode ser calculado em tempo polinomial em relação ao tamanho do modelo. Se, após a observação de um evento, a estimativa de estados do DRP não corresponder ao comportamento livre de falha esperado com relação a uma falha específica, a falha é diagnosticada.

Diversas estratégias de diagnóstico de falhas de SEDs modelados por RPs também são propostas na literatura. A principal vantagem de modelar um sistema usando RPs ao invés de autômatos é que a estrutura mais complexa

[★] Este trabalho foi parcialmente financiado pelo CNPq, FAPERJ, e Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil CAPES - finance code 001

da RP é mais adequada para representar determinadas informações estruturais do sistema, como concorrência e sincronização, do que autômatos, o que gera modelos mais compactos. O diagnóstico de falhas de RPs limitadas pode ser feito diretamente a partir do grafo de alcançabilidade da rede utilizando-se métodos desenvolvidos para sistemas modelados por autômatos. Nesse caso, a vantagem da modelagem em RP é perdida e, portanto, trabalhos recentes em diagnóstico de falhas para sistemas modelados por RPs evitam a construção do grafo de alcançabilidade completo.

As chamadas *marcações base* e *explicações mínimas* foram propostas em Cabasino et al. (2010) para evitar a construção do grafo de alcançabilidade completo para sistemas modelados por RPs com transições não observáveis. Em Cabasino et al. (2010) também é proposto um grafo determinístico, chamado de Grafo de Alcançabilidade Base (GAB), para, principalmente, realizar a maior parte dos cálculos offline. Entretanto, na análise de pior caso, de acordo com Yue et al. (2019), o número de nós do GAB é igual ao número de marcações alcançáveis da rede. Assim, calcular o GAB tem a mesma complexidade computacional de calcular o grafo de alcançabilidade da RPR no pior caso.

Em Basile et al. (2009), um diagnosticador interpretado é proposto. O método evita o cálculo de todas as possíveis marcações alcançáveis da rede após a observação de uma sequência de eventos. Entretanto, é suposto que a rede não tem duas transições observáveis distintas rotuladas com o mesmo evento.

Neste artigo, um método de diagnóstico de falhas para sistemas modelados por uma classe de RPRs é proposto. O método é baseado na construção de uma RPR a partir do comportamento livre de falha do modelo do sistema, chamada de rede de Petri do comportamento observável (RPCO), cujas transições são rotuladas apenas com eventos observáveis e cuja linguagem gerada é igual à linguagem observada do comportamento livre de falha do sistema. Diferente de Basile et al. (2009), o modelo em RP do sistema pode ter mais de uma transição observável rotulada com o mesmo evento. A RPCO fornece todos os eventos observáveis do comportamento livre de falha do sistema que são possíveis após a observação de uma sequência de eventos. Assim, o diagnóstico é feito analisando-se a marcação da RPCO após a observação de uma sequência de eventos. Se um evento que não rotula nenhuma das transições habilitadas da RPCO é observado, a falha é diagnosticada.

Diferente dos trabalhos apresentados em Cabasino et al. (2010, 2011), o método proposto não se baseia no cálculo de grafos de alcançabilidade para realizar o diagnóstico. É importante destacar que o número de lugares e transições da RPCO é menor ou igual ao número de lugares e transições da RPR que modela o comportamento livre de falha do sistema. Portanto, a memória necessária para implementar o diagnosticador é menor do que a memória necessária para implementar outros diagnosticadores em redes de Petri propostos na literatura.

2. CONCEITOS PRELIMINARES

Uma rede de Petri \mathcal{N} é uma quintupla $\mathcal{N} = (P, T, Pre, Post, x_0)$, em que $P = \{p_1, p_2, \dots, p_n\}$ é o conjunto

finito de lugares, $T = \{t_1, t_2, \dots, t_m\}$ é o conjunto finito de transições, $|P| = n$, $|T| = m$, em que $|\cdot|$ denota cardinalidade, $Pre : (P \times T) \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$ é a função de arcos que conectam lugares a transições, $Post : (T \times P) \rightarrow \mathbb{N}$ é a função de arcos que conectam transições a lugares e $x_0 : P \rightarrow \mathbb{N}$ é a função de marcação inicial (Murata, 1989; David and Alla, 2005).

O conjunto de lugares de entrada (resp. transições de entrada) de uma transição $t_j \in T$ (resp. lugar $p_i \in P$) é denotado como $I_p(t_j)$ (resp. $I_t(p_i)$) e é formado pelos lugares $p_i \in P$ (resp. transições $t_j \in T$) tais que $Pre(p_i, t_j) > 0$ (resp. $Post(t_j, p_i) > 0$). De forma análoga, o conjunto de lugares de saída (resp. transições de saída) de uma transição $t_j \in T$ (resp. lugar $p_i \in P$) é denotado como $O_p(t_j)$ (resp. $O_t(p_i)$) e é formado pelos lugares $p_i \in P$ (resp. transições $t_j \in T$) tais que $Post(t_j, p_i) > 0$ (resp. $Pre(p_i, t_j) > 0$). Um lugar $p_i \in P$ é dito ser um lugar fonte se $I_t(p_i) = \emptyset$.

O número de fichas associadas a um lugar p_i é representado por $x(p_i)$, em que $x : P \rightarrow \mathbb{N}$. Assim, a marcação de uma RP é o vetor coluna $\underline{x} = [x(p_1) \ x(p_2) \ \dots \ x(p_n)]^T$ formado pelo número de fichas em cada lugar p_i , para $i = 1, \dots, n$.

Uma transição $t_j \in T$ em uma RP é dita estar habilitada quando

$$x(p_i) \geq Pre(p_i, t_j), \forall p_i \in I_p(t_j). \quad (1)$$

Se uma transição t_j está habilitada para uma marcação \underline{x} , t_j pode disparar e quando dispara, uma nova marcação \underline{x}' é alcançada. A evolução das marcações em uma RP é dada por

$$x'(p_i) = x(p_i) - Pre(p_i, t_j) + Post(t_j, p_i), \quad (2)$$

para $i = 1, \dots, n$.

Neste artigo, a função de transição da RP é representada por $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$. Portanto, o disparo de uma transição habilitada t_j no estado \underline{x} leva a uma nova marcação $\underline{x}' = f(\underline{x}, t_j)$. O domínio de f pode ser estendido para $\mathbb{N}^n \times T^*$, em que \star denota o Fecho de Kleene, da seguinte forma: $f(\underline{x}, \theta) = \underline{x}$, em que θ denota a sequência vazia de transições; e $f(\underline{x}, st_j) = f(f(\underline{x}, s), t_j)$, para todo $s \in T^*$ e $t_j \in T$.

Uma RP ordinária é uma RP tal que $Pre(p_i, t_j) \leq 1$ e $Post(t_j, p_i) \leq 1$, para $i = 1, \dots, n$ e $j = 1, \dots, m$. Uma classe de RPs ordinárias é a chamada Rede de Petri Máquina de Estados (RPME). Em uma RPME cada transição tem um único lugar de entrada e um único lugar de saída. RPMEs permitem a representação de conflitos, mas não sincronização de atividades concorrentes (Murata, 1989). Um Grafo Marcado (GM), também chamado de Grafo de Eventos, é outra classe de RPs ordinárias em que cada lugar tem apenas uma transição de entrada e uma transição de saída. Diferente das RPMEs, GMs permitem a representação de sincronização de atividades concorrentes, mas não de conflitos.

Um lugar $p_i \in P$ é dito ser seguro se $x(p_i) \leq 1$ para todas as marcações alcançáveis da rede. Uma RP é dita ser segura para uma dada marcação inicial \underline{x}_0 se todos os lugares são seguros para todas as marcações alcançáveis da rede. Uma RP é dita ser livre de bloqueio se nenhuma marcação alcançável é bloqueante, *i.e.*, todas as marcações

alcançáveis habilitam ao menos uma transição da rede (David and Alla, 2005).

Seja $T' \subseteq T$. Então, a subrede T' -induzida de uma RP $\mathcal{N} = (P, T, Pre, Post, x_0)$ é definida como $\mathcal{N}' = (P, T', Pre', Post', x_0)$, em que $Pre' : P \times T' \rightarrow \mathbb{N}$ e $Post' : T' \times P \rightarrow \mathbb{N}$, com $Pre'(p_i, t_j) = Pre(p_i, t_j)$ e $Post'(t_j, p_i) = Post(t_j, p_i)$, para todo $t_j \in T'$ e $p_i \in P$ (Cabasino et al., 2010).

O grafo de uma RP \mathcal{N} é dito ser fracamente conexo se existir um caminho entre quaisquer pares de vértices pertencentes a $P \cup T$ no grafo não direcionado subjacente a \mathcal{N} . Além disso, uma subrede de \mathcal{N} é dita ser um componente fracamente conexo se ela é uma subrede maximal fracamente conexa de \mathcal{N} .

Para que RPs possam ser usadas para modelar SEDs, eventos são associados a transições, levando à chamada rede de Petri rotulada (RPR) (Cassandras and Lafortune, 2009). Uma RPR é uma séptupla $\mathcal{N}_l = (P, T, Pre, Post, x_0, \Sigma, \ell)$, em que $(P, T, Pre, Post, x_0)$ é uma RP, Σ é o conjunto de eventos e $\ell : T \rightarrow \Sigma$ é a função de rotulação que associa eventos às transições. O domínio da função de transição ℓ pode ser estendido para considerar sequências de disparos de transições, i.e., $\ell : T^* \rightarrow \Sigma^*$, sendo $\ell(\theta) = \varepsilon$, em que ε denota a sequência vazia de eventos, e $\ell(st) = \ell(s)\ell(t)$, para todo $s \in T^*$ e $t \in T$. A função de rotulação também pode ser aplicada a um conjunto de sequências de transições $B \subseteq T^*$ como $\ell(B) = \cup_{s \in B} \ell(s)$.

2.1 Diagnosticabilidade de SEDs modelados por RPRs

Seja \mathcal{N}_l uma RPR que modela o sistema. O conjunto de todas as sequências de transições de comprimento finito que podem disparar a partir da marcação inicial x_0 é dado por $S = \{s \in T^* : f(x_0, s)!\}$, em que $!$ denota *é definido*. A linguagem gerada por \mathcal{N}_l , L , é obtida usando a função de rotulação ℓ e é definida como $L = \ell(S)$. O conjunto de eventos Σ pode ser particionado como $\Sigma = \Sigma_u \dot{\cup} \Sigma_o$, em que Σ_u e Σ_o denotam, respectivamente, os conjuntos de eventos não observáveis e observáveis. O conjunto de eventos de falha é denotado por $\Sigma_f \subseteq \Sigma_u$. Neste artigo, por simplicidade, é suposto que o sistema tem apenas um tipo de evento de falha, i.e., $\Sigma_f = \{\sigma_f\}$. Os conjuntos de transições rotuladas por eventos de Σ_o , Σ_u e Σ_f são denotados por T_o , T_u e T_f , respectivamente.

A função máscara $M : T^* \rightarrow \Sigma_o^*$ é recursivamente definida como $M(t_j) = \sigma$, se $t_j \in T_o$, em que $\sigma = \ell(t_j)$, $M(t_j) = \varepsilon$, se $t_j \in T_u$, $M(st_j) = M(s)M(t_j)$, para todo $s \in T^*$ e $t_j \in T$, e $M(\theta) = \varepsilon$. A função máscara inversa $M^{-1} : \Sigma_o^* \rightarrow 2^{S^*}$ é definida como $M^{-1}(\rho) = \{s \in S : M(s) = \rho\}$.

Seja S_F o conjunto formado por todas as sequências de transições de S que têm ao menos uma transição de falha $t_f \in T_f$. A seguinte definição de diagnosticabilidade de uma RPR pode ser apresentada para redes de Petri que modelam linguagens regulares (Sampath et al., 1995).

Definição 1. Uma RPR livre de bloqueio \mathcal{N}_l é diagnosticável com relação a $M : T^* \rightarrow \Sigma_o^*$ e T_f se

$$(\exists z \in \mathbb{N})(\forall \nu \in S_F)(\forall \nu\mu \in S_F)(\|\mu\| \geq z \Rightarrow \forall \rho \in M^{-1}(M(\nu\mu)), \rho \in S_F), \quad (3)$$

em que $\|\cdot\|$ denota o comprimento de uma sequência. \square

O conjunto de todas as sequências de disparo de transições que não contêm nenhuma transição do conjunto T_f é denotado por S_N . Portanto, $S_N = S \setminus S_F$. A RPR que modela S_N é denotada neste artigo por $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$, em que $\Sigma_N = \Sigma \setminus \Sigma_f$. Note que T_N é particionado como $T_N = T_{N_o} \dot{\cup} T_{N_u}$, em que $T_{N_o} \subseteq T_o$ e $T_{N_u} \subseteq T_u$ denotam os conjuntos de transições observáveis e não observáveis de \mathcal{N}_N , respectivamente.

3. FORMULAÇÃO DO PROBLEMA

Com o objetivo de implementar um método de diagnóstico de falhas para SEDs modelados por RPRs é necessário calcular o conjunto de eventos observáveis viáveis da RPR após a observação de uma sequência de eventos, e verificar se o próximo evento observável gerado pelo sistema pertence a esse conjunto. Caso contrário, a falha é detectada. Neste artigo, um esquema de diagnóstico de falhas para uma classe de RPRs é proposto. O esquema de diagnóstico de falhas calcula online o conjunto de eventos observáveis viáveis do comportamento livre de falha do sistema, modelado por \mathcal{N}_N , após a observação de uma sequência de eventos, evitando-se o cálculo de todas as marcações alcançáveis da rede. Para isso, uma RPR $\mathcal{N}_o = (P_o, T_{N_o}, Pre_o, Post_o, x_{0,o}, \Sigma_o, \ell_o)$ tal que $M(S_N) = \ell_o(S_o)$, em que $S_o = \{s \in T_{N_o}^* : f_o(x_{0,o}, s)!\}$ e $f_o : \mathbb{N}^n \times T_{N_o}^* \rightarrow \mathbb{N}^n$ é a função de transição de \mathcal{N}_o , é obtida a partir de \mathcal{N}_N . Como a linguagem gerada por \mathcal{N}_o é igual à linguagem observada por \mathcal{N}_N , \mathcal{N}_o é chamada de rede de Petri do comportamento observável (RPCO).

Para caracterizar a classe de RPRs que podem ser consideradas para o esquema de diagnóstico proposto neste artigo, é necessário primeiro apresentar as seguintes hipóteses relacionadas ao modelo do sistema:

- H1.** O modelo do sistema em RPR \mathcal{N}_l é seguro.
- H2.** Não existe uma sequência de disparo de transições $s \in S$ tal que $s = uv$, em que $u \in T^*$, $v \in T_u^*$, e v tem comprimento arbitrariamente longo.
- H3.** Duas ou mais transições de \mathcal{N}_N rotuladas com o mesmo evento observável não podem estar habilitadas simultaneamente para qualquer estimativa de estado.

É importante destacar que diversos sistemas práticos podem ser modelados por RPs seguras (Murata, 1989), portanto, a hipótese **H1** é verificada em diversos casos. A hipótese **H2** requer que o sistema não tenha ciclos de eventos não observáveis, o que é uma hipótese comum feita no diagnóstico de SEDs (Sampath et al., 1995). A hipótese **H3** implica que a observação de um evento está relacionada com o disparo de uma única transição observável de \mathcal{N}_N , que é viável para todas as marcações alcançáveis da estimativa de estado atual. Portanto, após a observação de um evento gerado pelo sistema, a transição que disparou é unicamente identificada. É importante ressaltar que RPs que não possuem duas transições observáveis distintas rotuladas com o mesmo evento, como as RPs consideradas em Basile et al. (2009), satisfazem a hipótese **H3**.

Para caracterizar a classe de RPRs que é considerada neste artigo, é necessário encontrar todos os componentes fracamente conexos de \mathcal{N}_N , $\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k},$

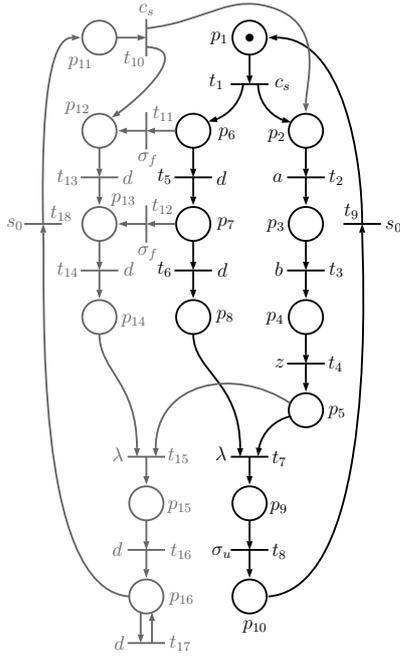


Figura 1. Modelo em RPR do sistema, \mathcal{N}_1 , do exemplo 1, em que o modelo em rede de Petri do comportamento livre de falha \mathcal{N}_N é apresentado em preto.

Σ_N, ℓ_k), tais que $T_k \neq \emptyset$ e $T_k \subseteq T_{N_u}$, para $k = 1, \dots, \rho$, em que ρ denota o número de componentes fracamente conexos de \mathcal{N}_N desse tipo.

Algoritmo 1 Cálculo de \mathcal{N}_k , $k = 1, \dots, \rho$, de \mathcal{N}_N

Entrada: $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$

Saída: $\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k}, \Sigma_N, \ell_k)$, $k = 1, \dots, \rho$

- 1: Calcule a subrede T_{N_u} -induzida de \mathcal{N}_N , \mathcal{N}_u
- 2: Encontre todos os componentes fracamente conexos \mathcal{N}_k de \mathcal{N}_u tais que $T_k \neq \emptyset$

No passo 1 do algoritmo 1, a subrede T_{N_u} -induzida de \mathcal{N}_N , \mathcal{N}_u , é calculada. Como T_{N_u} é formado apenas por transições não observáveis, \mathcal{N}_u pode ser formado por diversos grafos desconexos entre si. Assim, no passo 2 do algoritmo 1, subgrafos maximais fracamente conexos de \mathcal{N}_u que possuem pelo menos uma transição são obtidos, formando os componentes fracamente conexos \mathcal{N}_k .

Exemplo 1. Considere o modelo em RPR \mathcal{N}_1 do sistema apresentado na figura 1. O modelo do comportamento livre de falha \mathcal{N}_N é apresentado em preto, enquanto que a subrede relacionada com o comportamento pós-falha é representada em cinza. Os conjuntos de transições observáveis e não observáveis de \mathcal{N}_N são $T_{N_o} = \{t_1, t_4, t_5, t_6, t_9\}$ e $T_{N_u} = \{t_2, t_3, t_7, t_8\}$, respectivamente. Se aplicarmos o algoritmo 1 à \mathcal{N}_N , os componentes fracamente conexos \mathcal{N}_1 e \mathcal{N}_2 , representados na figura 2, são obtidos. Note que todas as transições de \mathcal{N}_1 e \mathcal{N}_2 são não observáveis e que ambos os subgrafos são acíclicos e não possuem transições fonte e sorvedouro. \square

A seguir, as transições de \mathcal{N}_k são classificadas de acordo com o número de lugares de entrada e saída.

Definição 2. Uma transição $t_j \in T_k$ de \mathcal{N}_k é dita ser SIMO — do inglês *single-input-multiple-output* — se $|I_p(t_j)| = 1$

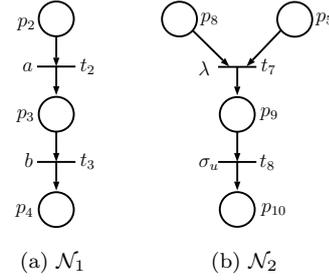


Figura 2. Componentes fracamente conexos de \mathcal{N}_N , \mathcal{N}_1 e \mathcal{N}_2 , do exemplo 1.

e $|O_p(t_j)| > 1$; a transição t_j é dita ser MISO — do inglês *multiple-input-single-output* — se $|I_p(t_j)| > 1$ e $|O_p(t_j)| = 1$; e t_j é dita ser SISO — do inglês *single-input-single-output* — se $|I_p(t_j)| = 1$ e $|O_p(t_j)| = 1$. \square

Note que uma transição SIMO modela a ativação de processos em paralelo na RPR, enquanto que transições MISO representam sincronização. As seguintes hipóteses em relação ao modelo em RP do sistema são também consideradas.

- H4.** Todos os lugares $p_i \in P_k$ que não são lugares fonte de \mathcal{N}_k , não podem ter transições de entrada observáveis em \mathcal{N}_N , *i.e.*, $Post_N(t_j, p_i) = 0$ para todo $t_j \in T_{N_o}$.
- H5.** Os lugares de \mathcal{N}_k que não são lugares fonte não podem ter uma ficha na marcação inicial de \mathcal{N}_N .
- H6.** Cada \mathcal{N}_k , para $k = 1, \dots, \rho$, é uma RPME com um único lugar fonte, ou todos os lugares de \mathcal{N}_k têm ao menos uma transição de entrada e uma transição de saída e as transições de \mathcal{N}_k que não são SISO são todas SIMO ou todas MISO.
- H7.** Não existe um lugar de entrada $p_{in} \in I_p(t_j)$ e um lugar de saída $p_{out} \in O_p(t_j)$, em que $t_j \in T_k$, tais que $x(p_{in}) = x(p_{out}) = 1$ para alguma marcação alcançável x de \mathcal{N}_k .

Neste artigo um método para o cálculo de \mathcal{N}_o , usada no esquema de diagnóstico para gerar todas as possíveis sequências de eventos observáveis do comportamento livre de falha do sistema, é apresentado para uma classe de RPR que satisfaz as hipóteses **H1-H7**.

4. CÁLCULO DA RPCO

O método proposto neste trabalho consiste em modificar o modelo livre de falha em RPR, \mathcal{N}_N , para calcular a RPCO \mathcal{N}_o ao remover todas as transições não observáveis de \mathcal{N}_N sem alterar sua linguagem observável. Para tanto, cada componente fracamente conexo \mathcal{N}_k é primeiramente identificado em \mathcal{N}_N , e então \mathcal{N}_k é modificado para remover suas transições não observáveis. De acordo com a hipótese **H6**, \mathcal{N}_k pode ser uma rede de Petri máquina de estados, ou as transições que não são SISO são todas SIMO ou todas MISO. Na sequência, o algoritmo que modifica \mathcal{N}_N , eliminando-se as transições não observáveis da subrede \mathcal{N}_k , quando \mathcal{N}_k é uma RPME é apresentado.

De acordo com o algoritmo 2, todos os lugares da RPME \mathcal{N}_k podem ser substituídos por um único lugar p_s . Para mostrar que essa redução não altera a linguagem observável do modelo livre de falha do sistema, primeiro o conjunto de todas as sequências de disparo de transições

Algoritmo 2 Algoritmo de modificação quando \mathcal{N}_k é uma RPME

Entradas: $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$,
 $\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k}, \Sigma_N, \ell_k)$

Saída: $\mathcal{N}_r = (P_r, T_r, Pre_r, Post_r, x_{0,r}, \Sigma_N, \ell_r)$

- 1: Seja p_s o lugar fonte de \mathcal{N}_k , e defina $P_{k,s} = P_k \setminus \{p_s\}$
- 2: $P_r = P_N \setminus P_{k,s}$
- 3: $T_r = T_N \setminus T_k$
- 4: **para cada** $p_i \in P_k$ **faça**
- 5: **Se** $\exists t_j \in T_{N_o}$ tal que $Pre_N(p_i, t_j) = 1$ **faça**
 $Pre_r(p_s, t_j) = 1$
- 6: $Pre_r(p_i, t_j) = Pre_N(p_i, t_j), \forall t_j \in T_r, \forall p_i \in P_r \setminus \{p_s\}$
- 7: $Post_r(t_j, p_i) = Post_N(t_j, p_i), \forall t_j \in T_r, \forall p_i \in P_r$
- 8: $x_{0,r}(p_i) = x_{0,N}(p_i), \forall p_i \in P_r$
- 9: $\ell_r(t_j) = \ell_N(t_j), \forall t_j \in T_r$

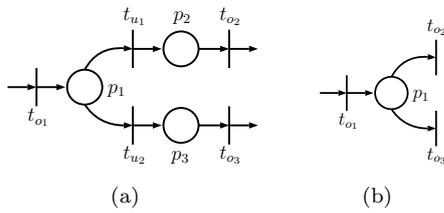


Figura 3. Partes das redes de Petri \mathcal{N}_N (a) e \mathcal{N}_r (b) do exemplo 2.

de \mathcal{N}_r é definido como $S_r = \{s \in T_r^* : f_r(x_{0,r}, s)!\}$, em que $f_r : \mathbb{N}^{n_r} \times T_r \rightarrow \mathbb{N}^{n_r}$ é a função de transição de \mathcal{N}_r , e n_r é o número de lugares de \mathcal{N}_r .

Teorema 1. Seja \mathcal{N}_r a RPR obtida a partir de \mathcal{N}_N de acordo com o algoritmo 2. Sejam S_N e S_r os conjuntos de todas as sequências de disparo de transições de comprimento finito de \mathcal{N}_N e \mathcal{N}_r , respectivamente. Então, $M(S_N) = M(S_r)$.

Prova. A prova foi omitida por falta de espaço. ■

Exemplo 2. Considere parte de uma rede de Petri \mathcal{N}_N apresentada na figura 3(a), em que o componente fracamente conexo \mathcal{N}_1 é formado pelos lugares p_1, p_2 , e p_3 , e as transições não observáveis t_{u1} e t_{u2} . Aplicando-se o algoritmo 2, a rede reduzida da figura 3(b) é obtida. □

Em seguida, são apresentadas as regras de redução para componentes fracamente conexos de \mathcal{N}_k que não são RPME. No algoritmo 3, a redução de \mathcal{N}_N eliminando-se uma transição não observável SIMO de uma dada subrede \mathcal{N}_k é apresentada, e no algoritmo 4, é feita a redução de \mathcal{N}_N eliminando-se uma transição não observável MISO.

O teorema 2 mostra que a linguagem observável de \mathcal{N}_N é igual à linguagem observável de \mathcal{N}_r obtida utilizando-se o algoritmo 3 ou 4.

Teorema 2. Seja \mathcal{N}_r a RPR obtida a partir de \mathcal{N}_N de acordo com os algoritmos 3 ou 4. Sejam S_N e S_r os conjuntos de todas as sequências de disparo de transições de comprimento finito de \mathcal{N}_N e \mathcal{N}_r , respectivamente. Então, $M(S_N) = M(S_r)$.

Prova. A prova foi omitida por falta de espaço. ■

Exemplo 3. Considere partes de uma rede de Petri \mathcal{N}_N com transições SIMO e MISO apresentadas na figura 4(a) e figura 5(a), respectivamente. Então, as RP reduzidas

Algoritmo 3 Algoritmo de eliminação de uma transição SIMO de \mathcal{N}_k

Entradas: $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$,
 $\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k}, \Sigma_N, \ell_k)$,

Transição SIMO $t_u \in T_k$

Saída: $\mathcal{N}_r = (P_r, T_r, Pre_r, Post_r, x_{0,r}, \Sigma_N, \ell_r)$

- 1: $T_r = T_N \setminus \{t_u\}$
- 2: $P_r = P_N \setminus \{p_u\}$, em que $p_u = I_p(t_u)$
- 3: $Post_r(t_j, p_i) = Post_N(t_j, p_i), \forall t_j \in T_r$ e $\forall p_i \in P_r$
- 4: Seja $t_{in} \in I_t(p_u)$. Defina $Post_r(t_{in}, p_i) = 1, \forall p_i \in O_p(t_u)$
- 5: $Pre_r(p_i, t_j) = Pre_N(p_i, t_j), \forall p_i \in P_r$ e $\forall t_j \in T_r$
- 6: **para cada** lugar $p_i \in O_p(t_u)$ **faça**
 $Pre_r(p_i, t_j) = 1, \forall t_j \in O_t(p_u) \cap T_N$.
- 7: **Se** $x_{0,N}(p_u) = 1$ **faça**
 $x_{0,r}(p_i) = 1, \forall p_i \in O_p(t_u)$
- 8: **Senão**
 $x_{0,r}(p_i) = 0, \forall p_i \in O_p(t_u)$
- 9: $x_{0,r}(p_i) = x_{0,N}(p_i), \forall p_i \in P_r \setminus O_p(t_u)$
- 10: $\ell_r(t_j) = \ell_N(t_j), \forall t_j \in T_r$

Algoritmo 4 Algoritmo de eliminação de uma transição MISO de \mathcal{N}_k

Entradas: $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$,
 $\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k}, \Sigma_N, \ell_k)$,

Transição MISO $t_u \in T_k$

Saída: $\mathcal{N}_r = (P_r, T_r, Pre_r, Post_r, x_{0,r}, \Sigma_N, \ell_r)$

- 1: $T_r = T_N \setminus \{t_u\}$
- 2: $P_r = P_N \setminus \{p_u\}$, em que $p_u = O_p(t_u)$
- 3: $Pre_r(p_i, t_j) = Pre_N(p_i, t_j), \forall t_j \in T_r$ e $\forall p_i \in P_r$
- 4: Seja $t_{out} \in O_t(p_u)$. Defina $Pre_r(p_i, t_{out}) = 1$, para todo $p_i \in I_p(t_u)$
- 5: $Post_r(t_j, p_i) = Post_N(t_j, p_i)$, para todo $p_i \in P_r$ e $t_j \in T_r$
- 6: $x_{0,r}(p_i) = x_{0,N}(p_i), \forall p_i \in P_r$
- 7: $\ell_r(t_j) = \ell_N(t_j), \forall t_j \in T_r$

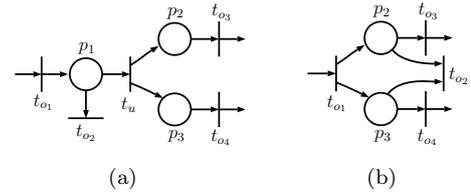


Figura 4. Parte de uma RP \mathcal{N}_N com uma transição SIMO (a) e a correspondente \mathcal{N}_r (b) do exemplo 3.

\mathcal{N}_r , calculadas utilizando-se os algoritmos 3 e 4, são apresentadas nas figuras 4(b) e 5(b), respectivamente. □

Um método para o cálculo de \mathcal{N}_o a partir de \mathcal{N}_N , usando os algoritmos 2, 3 e 4, é apresentado no algoritmo 5.

Teorema 3. Seja \mathcal{N}_o a RPR obtida de acordo com o algoritmo 5 a partir do modelo livre de falha \mathcal{N}_N . Então $M(S_N) = \ell_o(S_o)$.

Prova. A prova foi omitida por falta de espaço. ■

Como \mathcal{N}_o é obtido aplicando-se apenas as regras de redução dos algoritmos 2, 3, e 4, então o número de lugares e transições de \mathcal{N}_o é sempre menor ou igual ao número de lugares e transições de \mathcal{N}_N . Aplicando-se o algoritmo 5 na

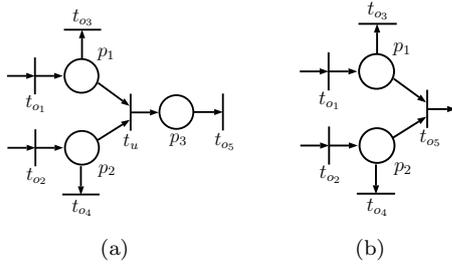


Figura 5. Parte de uma RP \mathcal{N}_N com uma transição MISO (a) e a correspondente \mathcal{N}_r (b) do exemplo 3.

Algoritmo 5 Cálculo de \mathcal{N}_o

Entradas: $\mathcal{N}_N = (P_N, T_N, Pre_N, Post_N, x_{0,N}, \Sigma_N, \ell_N)$

$\mathcal{N}_k = (P_k, T_k, Pre_k, Post_k, x_{0,k}, \Sigma_N, \ell_k), k = 1, \dots, \rho$

Saída: $\mathcal{N}_o = (P_o, T_{N_o}, Pre_o, Post_o, x_{0,o}, \Sigma_N, \ell_o)$

- 1: Defina $\mathcal{N}_c = \mathcal{N}_N$
- 2: **Para** $k = 1, \dots, \rho$ **faça**
- 3: **Se** \mathcal{N}_k é uma RPME **faça**
- 4: Elimine todas as transições não observáveis em T_k de \mathcal{N}_c usando algoritmo 2, obtendo-se a RPR \mathcal{N}'_N
- 5: Defina $\mathcal{N}_c = \mathcal{N}'_N$
- 6: **Se** \mathcal{N}_k tem transições SIMO **faça**
- 7: Elimine todas as transições não observáveis em T_k de \mathcal{N}_c aplicando-se recursivamente o algoritmo 3, formando a RP reduzida \mathcal{N}'_N
- 8: Defina $\mathcal{N}_c = \mathcal{N}'_N$
- 9: **Se** \mathcal{N}_k tem transições MISO **faça**
- 10: Elimine todas as transições não observáveis em T_k de \mathcal{N}_c aplicando-se recursivamente o algoritmo 4, formando a RP reduzida \mathcal{N}'_N
- 11: Defina $\mathcal{N}_c = \mathcal{N}'_N$
- 12: $\mathcal{N}_o = \mathcal{N}_c$

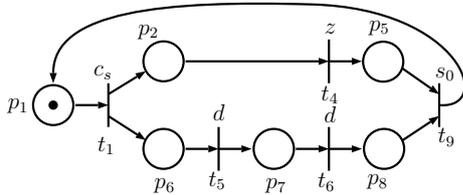


Figura 6. RPCO \mathcal{N}_o .

rede de Petri \mathcal{N}_N do exemplo 1 (figura 1), obtém-se a rede de Petri \mathcal{N}_o ilustrada na figura 6.

5. CONCLUSÕES

Neste artigo, um esquema de diagnóstico de falhas para SEDs modelados por uma classe específica de redes de Petri rotuladas é apresentado. O método não requer a construção do grafo de alcançabilidade (ou parte dele) da rede e sempre leva a uma RPR com número de lugares e transições menor ou igual ao da RP que modela o sistema. Outras classes de RPR que possam ser usadas evitando-se a construção de estruturas complexas para o diagnóstico de falhas está sendo investigada pelos autores.

REFERÊNCIAS

Basile, F., Chiacchio, P., and De Tommasi, G. (2009). An efficient approach for online diagnosis of discrete event

systems. *IEEE Transactions on Automatic Control*, 54(4), 748–759.

Cabasino, M.P., Giua, A., Poggi, M., and Seatzu, C. (2011). Discrete event diagnosis using labeled Petri nets. an application to manufacturing systems. *Control Engineering Practice*, 19(9), 989–1001.

Cabasino, M.P., Giua, A., and Seatzu, C. (2010). Fault detection for discrete event systems using Petri nets with unobservable transitions. *Automatica*, 46(9), 1531–1539.

Cabral, F.G., Moreira, M.V., Diene, O., and Basilio, J.C. (2015). A Petri net diagnoser for discrete event systems modeled by finite state automata. *IEEE Transactions on Automatic Control*, 60(1), 59–71.

Cassandras, C.G. and Lafortune, S. (2009). *Introduction to discrete event systems*.

David, R. and Alla, H. (2005). *Discrete, continuous, and hybrid Petri nets*, volume 1.

Giua, A., Seatzu, C., and Corona, D. (2007). Marking estimation of Petri nets with silent transitions. *IEEE Transactions on Automatic Control*, 52(9), 1695–1699.

Miyagi, P. and Riascos, L. (2006). Modeling and analysis of fault-tolerant systems for machining operations based on Petri nets. *Control Engineering Practice*, 14(4), 397–408.

Moreira, M.V., Basilio, J.C., and Cabral, F.G. (2016). “polynomial time verification of decentralized diagnosability of discrete event systems” versus “decentralized failure diagnosis of discrete event systems”: A critical appraisal. *IEEE Transactions on Automatic Control*, 61(1), 178–181.

Moreira, M.V., Jesus, T.C., and Basilio, J.C. (2011). Polynomial time verification of decentralized diagnosability of discrete event systems. *IEEE Transactions on Automatic Control*, 56(7), 1679–1684.

Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4), 541–580.

Qiu, W. and Kumar, R. (2006). Decentralized failure diagnosis of discrete event systems. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 36(2), 384–395.

Ran, N., Su, H., Giua, A., and Seatzu, C. (2017). Co-diagnosability analysis of bounded Petri nets. *IEEE Transactions on Automatic Control*, 63(4), 1192–1199.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D. (1995). Diagnosability of discrete-event systems. *IEEE Transactions on automatic control*, 40(9), 1555–1575.

Santoro, L.P., Moreira, M.V., and Basilio, J.C. (2017). Computation of minimal diagnosis bases of discrete-event systems using verifiers. *Automatica*, 77, 93 – 102.

Veras, M.Z., Cabral, F.G., and Moreira, M.V. (2021). Distributed synchronous diagnosis of discrete event systems modeled as automata. *Control Engineering Practice*, 115, 104892.

Yue, H., Xu, S., Zhou, G., Hu, H., Guo, Y., and Zhang, J. (2019). Estimation of least-cost transition firing sequences in labeled petri nets by using basis reachability graph. *IEEE Access*, 7, 165387–165398.