DOI: 10.48011/asba.v2i1.1060

Diagnosticabilidade Síncrona Distribuída de Sistemas a Eventos Discretos Sujeita a Atrasos de Comunicação de Eventos ⋆

Guilherme T. Araújo * **Felipe G. Cabral** * **Marcos V. Moreira** **

* *PPGEAS - Programa de Pós-Graduação em Engenharia de Automação e Sistemas, Universidade Federal de Santa Catarina, Campus Trindade, Florianópolis, 88.040-900, SC, Brasil, (e-mail: guilherme.teixeira.araujo@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, RJ, Brasil, (e-mail: moreira.mv@poli.ufrj.br)*

Abstract: Recently, the distributed synchronous diagnosis architecture for Discrete Event Systems has been proposed. In this scheme, local diagnosers, computed from the fault free behavior of the system components, which are capable of communicating event observations and state estimates between them, are proposed. In this context, the local diagnosers network is supposed to be ideal, *i.e.*, there are no communication delays or package losses. However, in complex systems, this assumption cannot be guaranteed, and communication delays can occur. In order to address this problem, in this work, the distributed synchronous diagnosis subject to event communication delays is considered. In order to do so, we consider that the local diagnosers can only communicate event observations. A modification of the component models is proposed in order to take into account the effect of the maximum delay that can be observed by each local diagnoser. Moreover, a new distributed synchronous diagnosability definition is proposed.

Resumo: Recentemente, a arquitetura de diagnóstico síncrono distribuído de sistemas a eventos discretos foi proposta. Nesse esquema, diagnosticadores locais, construídos a partir do comportamento sem falha dos componentes do sistema, que comunicam observações de eventos e estimativas de estado são propostos. Nesse contexto, é suposto que a comunicação entre diagnosticadores é ideal, ou seja, não há atrasos ou perdas de pacote. Entretanto, em sistemas complexos, nem sempre é possível supor que a rede de comunicação é ideal, podendo haver atrasos na comunicação entre diagnosticadores locais. Para abordar esse problema, neste trabalho, o diagnóstico síncrono distribuído sujeito a atrasos de comunicação de eventos é considerado. Para tanto, é suposto que os diagnosticadores locais comunicam apenas observações de eventos. Uma modificação dos modelos dos componentes do sistema é proposta para levar em consideração o atraso máximo de observação de eventos por cada diagnosticador local. Além disso, uma nova definição de diagnosticabilidade síncrona distribuída, que leva em consideração possíveis atrasos de observação é proposta.

Keywords: Discrete event systems; Fault diagnosis; Distributed architecture; Communication delays; Diagnosability verification.

Palavras-chaves: Sistemas a eventos discretos; Diagnóstico de falhas; Arquitetura distribuída; Atraso de comunicação; Verificação de diagnosticabilidade.

1. INTRODUÇÃO

A quarta revolução industrial, conhecida como Indústria 4.0, e o aumento no uso de dispositivos conectados à internet, formando a chamada Internet das Coisas, vêm tornando os sistemas de engenharia mais conectados e descentralizados, com capacidade de processamento local e troca de informações (Gilchrist, 2016). Esses sistemas são conhecidos como Sistemas Ciber-Físicos (SCFs). Essa tecnologia tem permitido que sistemas de manufatura sejam desenvolvidos em arquiteturas descentralizadas, em que, embora seus componentes estejam fisicamente distribuídos, podem trocar informações entre si.

Nesse contexto, o diagnóstico de falhas tem um papel importante, já que as consequências da ocorrência de uma falha em um componente de um SCF pode se espalhar para outros módulos do sistema, potencialmente causando danos a equipamentos e operadores. Um dos primeiros trabalhos que abordam o problema de diagnóstico de falhas

⋆ Este trabalho foi parcialmente financiado por Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil CAPES - Finance Code 001, FAPERJ e CNPq.

de sistemas modelados como Sistemas a Eventos Discretos (SEDs) é apresentado em Sampath et al. (1995). Nesse trabalho, um diagnosticador construído a partir de um observador do modelo completo do sistema em autômato é proposto. Geralmente, o cálculo desse diagnosticador é evitado, uma vez que seu espaço de estados cresce exponencialmente com o número de estados da planta (que por sua vez, cresce exponencialmente com o número de componentes do sistema) na análise de pior caso. A abordagem de se construir um único diagnosticador com base no modelo completo do sistema é chamada de abordagem centralizada, ou monolítica.

Em geral, SEDs são formados por diversos componentes que interagem entre si com o objetivo de completar determinada tarefa. Assim sendo, uma técnica de diagnóstico de falhas que se aproveita da natureza distribuída e capacidade de comunicação de SCFs pode ser mais eficaz do que arquiteturas monolíticas. Por conta disso, abordagens descentralizadas e distribuídas de diagnóstico de falhas de SEDs têm sido propostas na literatura. Em Debouk et al. (2000); Qiu and Kumar (2006); Wang et al. (2007), uma arquitetura de diagnóstico de falhas descentralizada é considerada. Nessa arquitetura, diagnosticadores locais baseados no modelo completo do sistema, que têm acesso apenas à observação local de eventos, são considerados. Se um diagnosticador local identifica a ocorrência do evento de falha, o diagnóstico é enviado a um coordenador que encaminha essa informação ao operador do sistema. Uma arquitetura diferente de diagnóstico de falhas chamada de diagnóstico distribuído é proposta em Qiu and Kumar (2008); Keroglou and Hadjicostis (2014, 2018). Nessa arquitetura, diagnosticadores locais podem trocar informações sobre a observação de eventos ou estimativas de estado, evitando-se a necessidade de um coordenador. Embora essas técnicas sejam mais adequadas a SCFs, uma vez que nesses sistemas a informação é fisicamente distribuída, os diagnosticadores locais são calculados a partir do modelo global da planta.

Uma arquitetura de diagnóstico diferente que evita o uso do modelo global da planta para o diagnóstico, conhecida como arquitetura modular, é proposta em Debouk et al. (2002); Contant et al. (2006); Schmidt (2013). Nesses trabalhos, um diagnosticador local é proposto apenas para o componente em que o evento de falha é modelado, chamado de módulo de falha. Definições de diagnosticabilidade modular que dependem de uma propriedade conhecida como excitação persistente, são propostas. Um componente tem excitação persistente se, para toda sequência de eventos de comprimento arbitrariamente longo gerada pela planta composta, a projeção nos eventos do componente também seja formada por uma sequência arbitrariamente longa. É importante notar que nenhum método de verificação dessa propriedade é apresentado em Debouk et al. (2002); Contant et al. (2006); Schmidt (2013). Além disso, nesses trabalhos é suposto que os módulos não têm eventos não observáveis em comum, o que limita sua aplicabilidade.

Recentemente, uma nova técnica para o diagnóstico de falhas de SEDs modelados por autômatos, chamada de diagnóstico síncrono, foi proposta em Cabral and Moreira (2020). Nesse trabalho, diagnosticadores locais são calculados para cada componente do sistema, a partir de seus modelos sem falha. Diferentemente do método de diagnóstico modular, na arquitetura síncrona nenhuma hipótese é feita em relação aos modelos dos componentes do sistema. Em Cabral and Moreira (2020), as arquiteturas centralizada e descentralizada são consideradas para o diagnóstico síncrono. Em Veras et al. (2018), essa proposta foi estendida para considerar uma arquitetura distribuída, em que os diagnosticadores locais podem ser conectados em rede e trocar informações sobre observação de eventos e estimativas de estado.

Em Veras et al. (2018) é suposto que a comunicação entre diagnosticadores locais é ideal, ou seja, não há atrasos ou perdas de pacote nos canais de comunicação entre diagnosticadores. Essa condição nem sempre pode ser garantida, principalmente em aplicações envolvendo a Indústria 4.0 e Internet das Coisas. Nesses casos, o método de diagnóstico síncrono proposto em Veras et al. (2018) não pode ser implementado. Alguns trabalhos na literatura lidam com o problema de atraso de comunicação de eventos para o diagnóstico considerando uma arquitetura descentralizada (Qiu and Kumar, 2008; Nunes et al., 2018).

Neste artigo, o problema de diagnóstico síncrono distribuído considerando atraso de comunicação entre diagnosticadores locais é considerado. A arquitetura usada neste trabalho leva em consideração que apenas a informação da ocorrência de eventos é comunicada entre locais e, portanto, pode sofrer atrasos decorrentes do canal de comunicação. Para tanto, uma definição de diagnosticabilidade síncrona distribuída robusta a atrasos de comunicação de eventos é definida, bem como um método de verificação dessa propriedade. Um exemplo é usado ao longo do texto para ilustrar o método.

2. FUNDAMENTAÇÃO TEÓRICA

2.1 Linguagens e autômatos

Seja $G = (Q, \Sigma, f, q_0)$ o modelo em autômato de um SED, em que Q é o conjunto de estados, Σ denota o conjunto de eventos, $f : Q \times \Sigma^\star \to Q$ é a função de transição, em que $\Sigma^\star$ o fecho de Kleene de Σ, e q_0 é o estado inicial. Seja $\Gamma_G : Q \to 2^\Sigma$ a função de eventos ativos, em que $\Gamma_G(q) = \{\sigma \in \Sigma : f(q, \sigma)!\}$, e $f(q, \sigma)!$ denota que $f(q, \sigma)$ é definida. Uma transição $f(q, \sigma) = q'$ de G também é denotada como (q, σ, q').

A linguagem gerada por G, $L(G)$, é denotada neste trabalho simplesmente por L. A sequência vazia é denotada por ε. O fecho de prefixo de $L \subseteq \Sigma^\star$ é dado por $\overline{L} = \{s \in \Sigma^\star : (\exists t \in \Sigma^\star)[st \in L]\}$. Uma linguagem $L \subseteq \Sigma^\star$ é dita ser viva se para todo $s \in L$, existe $\sigma \in \Sigma$ tal que $s\sigma \in L$. A sequência de eventos com menor comprimento tal que σ é o último evento é denotada por s_σ.

A parte acessível de G é denotada por $Ac(G)$. Sejam G_1 e G_2 dois autômatos, então $G_1 \times G_2$ e $G_1 \| G_2$ denotam a composição produto e composição paralela de G_1 e G_2, respectivamente. A projeção $P_s^l : \Sigma_l^\star \to \Sigma_s^\star$, em que $\Sigma_s \subset \Sigma_l$, e a projeção inversa $P_s^{l^{-1}} : \Sigma_s^\star \to 2^{\Sigma_l^\star}$ são definidas da forma usual (Cassandras and Lafortune, 2008).

O conjunto de eventos de G pode ser particionado como $\Sigma = \Sigma_o \dot{\cup} \Sigma_u$, em que Σ_o e Σ_u denotam os eventos

observáveis e não observáveis, respectivamente. Seja $\Sigma_f \subseteq \Sigma_u$ o conjunto de eventos de falha. Por simplicidade, sem perda de generalidade, é suposto que existe apenas um tipo de falha, ou seja, $\Sigma_f = \{\sigma_f\}$. Uma sequência de falha é uma sequência de eventos s tal que σ_f é um dos eventos que forma s. Uma sequência livre de falha não contém o evento σ_f. A linguagem livre de falha $L_N \subset L$ denota o conjunto de todas as sequências livres de falha de L e o subautômato de G que gera L_N é denotado por G_N. Portanto, o conjunto de todas as sequências de falha é dado por $L_F = L \setminus L_N$, em que $\setminus$ denota a diferença de conjuntos e G_F denota o autômato que gera a linguagem $\overline{L_F}$.

O alcance não observável de $q \in Q$, com relação a Σ_u é definido como $UR(q) = \{p \in Q : (\exists t \in \Sigma_u^\star)[f(q,t) = p]\}$. Essa definição pode ser estendida para um conjunto de estados $A \subseteq Q$ como $UR(A) = \cup_{q\in A} UR(q)$.

2.2 Diagnóstico síncrono descentralizado de SEDs

Seja $G = \|_{i=1}^r G_i$, $G_i = (Q_i, \Sigma_i, f_i, q_{0,i})$, em que $\Sigma_i = \Sigma_{i,o} \dot\cup \Sigma_{i,u}$, sendo que $\Sigma_{i,o}$ e $\Sigma_{i,u}$ são os conjuntos de eventos observáveis e não observáveis de G_i, respectivamente. O conjunto de eventos observáveis de G, Σ_o, é dado por $\Sigma_o = \cup_{i=1}^r \Sigma_{i,o}$. Na arquitetura de diagnóstico síncrono descentralizado, diagnosticadores locais D_i, $i = 1, \ldots, r$, obtidos a partir do comportamento sem falha dos componentes do sistema G_{N_i}, cuja linguagem gerada é denotada por L_{N_i}, são implementados localmente. Se ao menos um diagnosticador local D_i identifica a ocorrência de um evento de falha, D_i envia a informação do diagnóstico a um coordenador que é responsável por repassar essa informação ao operador do sistema. É importante notar que um determinado evento $\sigma \in \Sigma_{i,o}$ pode não observável para o local j, isto é, $\sigma \in \Sigma_{j,u}$, para $i \neq j$.

A definição de codiagnosticabilidade síncrona de SEDs é definida a seguir (Cabral and Moreira, 2020).

Definição *1.* (Codiagnosticabilidade síncrona) Seja L_{N_i} a linguagem gerada por G_{N_i}. L é dita ser sincronamente codiagnosticável em relação à L_{N_i}, projeções $P_{i,o} : \Sigma^\star \to \Sigma_{i,o}^\star$, $i = 1, \ldots, r$, e σ_f se

$$(\exists z \in \mathbb{N})(\forall s \in L_F)(\forall st \in L_F, \|t\| \geq z) \Rightarrow$$
$$(\exists i \in \{1, 2, \ldots, r\})[P_{i,o}(st) \notin P_{i,o}(L_{N_i})],$$

em que $\|t\|$ denota o comprimento da sequência t.

Em Cabral and Moreira (2020) é mostrado que a diagnosticabilidade síncrona é um caso particular da codiagnosticabilidade síncrona. No caso da implementação centralizada, todo evento $\sigma \in \Sigma_{i,o}$ observável a um determinado diagnosticador local D_i é observável a todo diagnosticador local D_j em que σ é definido. A definição de diagnosticabilidade síncrona pode ser diretamente obtida a partir da definição 1 atualizando-se os conjuntos de eventos observáveis de cada local. Algoritmos para verificação de ambas as propriedades são também apresentados em Cabral and Moreira (2020).

3. DIAGNÓSTICO SÍNCRONO DISTRIBUÍDO SUJEITO A ATRASOS DE COMUNICAÇÃO

3.1 Formulação do problema

Neste trabalho, o problema do diagnóstico síncrono distribuído sujeito a atrasos de comunicação é considerado. Para tanto, considere que o modelo do sistema G é composto por r componentes, ou seja, $G = \|_{i=1}^r G_i$ e associado a cada componente $G_i, i \in \{1, \ldots, r\}$, existe um diagnosticador local D_i, calculado a partir do comportamento livre de falha dos componentes G_{N_i}. Com o objetivo de refinar o diagnóstico síncrono, os diagnosticadores locais D_i são implementados em rede e podem trocar informações relacionadas à observação local de eventos. Nessa configuração, existe um local de medição LM_i associado a cada componente que comunica observações de eventos unidirecionalmente a D_i por meio de um canal de comunicação ideal, ou seja, sem atrasos ou perdas de pacotes. A figura 1 ilustra a arquitetura considerada neste trabalho, em que, por exemplo, o diagnosticador local D_1 recebe diretamente de LM_1 as observações dos eventos do conjunto $\Sigma_o^{1,1}$, podendo enviar essas informações a D_2 e D_3, e recebe informações relacionadas à observação de eventos por D_2 e D_3 por meio dos canais $ch_{1,2}$ e $ch_{1,3}$, respectivamente.

Assim como no diagnóstico síncrono descentralizado, o conjunto de eventos do componente G_i é denotado por $\Sigma_i = \Sigma_{i,o} \dot\cup \Sigma_{i,u}$, em que $\Sigma_{i,o}$ e $\Sigma_{i,u}$ são os conjuntos de eventos observáveis e não observáveis de G_i, respectivamente, e $\Sigma_o = \cup_{i=1}^r \Sigma_{i,o}$. O conjunto de eventos observáveis $\Sigma_{i,o} = \cup_{j=1}^r \Sigma_o^{i,j}$, em que $\Sigma_o^{i,i}$ é o conjunto formado pelos eventos cuja observação é realizada pelo local de medição LM_i, e $\Sigma_o^{i,j}$ para $i \neq j$ é o conjunto formado pelos eventos cuja observação é informada para o diagnosticador local D_i pelo diagnosticador local D_j pelo canal de comunicação $ch_{i,j} = ch_{j,i}$, $i, j \in \{1, \ldots, r\}$.

As seguintes hipóteses são consideradas para o diagnóstico síncrono distribuído sujeito a atrasos de comunicação de observação de eventos.

- **H1.** O atraso na comunicação de um evento $\sigma \in \Sigma_o^{i,j}$ é contado em passos (Tripakis, 2004), sendo que um passo corresponde à ocorrência de um evento em G e é definido para o canal de comunicação $ch_{i,j} = ch_{j,i}$. Assim, o atraso é medido pelo número de eventos gerados na planta após a ocorrência de σ e antes de sua efetiva comunicação ao diagnosticador D_i.
- **H2.** Todo canal de comunicação $ch_{i,j}$, $i, j \in \{1, \ldots, r\}$ e $i \neq j$, tem atraso limitado.
- **H3.** Os canais de comunicação se comportam como uma fila FIFO (*first-in first-out*).
- **H4.** Existe apenas um canal $ch_{i,j} = ch_{j,i}$ entre os diagnosticadores locais D_i e D_j, cujo atraso máximo, denotado por $k_{i,j} = k_{j,i}$, $k_{i,j}, k_{j,i} \in \mathbb{N}$, é previamente conhecido. Neste trabalho, consideramos que é possível que $k_{i,j} = k_{j,i} = 0$.
- **H5.** $\Sigma_o^{i,i} \cap \Sigma_o^{j,j} = \emptyset$ para todo $i, j \in \{1, \ldots, r\}, i \neq j$.
- **H6.** Não há perdas de pacote em nenhum canal de comunicação e apenas a informação relacionada à observação de eventos é comunicada.
- **H7.** Toda a comunicação entre diagnosticadores locais é feita dois a dois, ou seja, eventos observados em um

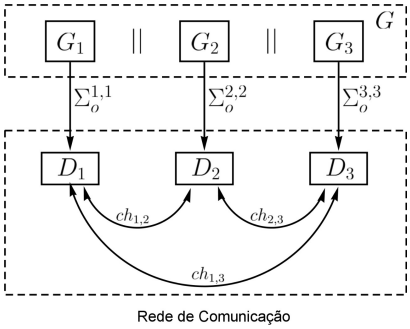


Figura 1. Arquitetura do diagnóstico síncrono distribuído.

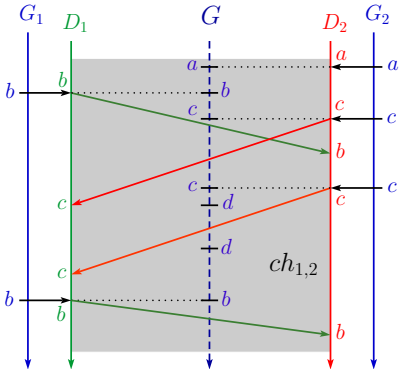


Figura 2. Linha do tempo do exemplo 1.

diagnosticador local D_i são diretamente comunicados a outro diagnosticador local D_j, para $i \neq j$.

O efeito prático do atraso de comunicação é a possível observação de uma sequência de eventos com a ordem de ocorrência errada, como mostrado em Nunes et al. (2018) para uma arquitetura descentralizada. Para lidar com esse problema, em Nunes et al. (2018) é proposto alterar o modelo da planta para representar todas as possibilidades de observação que podem ser feitas. Neste trabalho, diferentemente de Nunes et al. (2018), a arquitetura é distribuída e os diagnosticadores locais são baseados nos componentes do sistema, o que faz com que nem sempre o atraso de comunicação de um canal $ch_{i,j}$ resulte em uma observação equivocada por determinado diagnosticador (ou ainda, pode resultar em um atraso menor do que $k_{i,j}$). Para ilustrar esse fenômeno, considere o exemplo a seguir.

Exemplo *1.* Considere novamente a arquitetura de diagnóstico síncrono distribuído apresentada na figura 1 e suponha que o conjunto de eventos observáveis de G é $\Sigma_o = \Sigma_{1,o} \cup \Sigma_{2,o} \cup \Sigma_{3,o} = \{a, b, c, d\}$, em que $\Sigma_{1,o} = \{b, c\}$, $\Sigma_{2,o} = \{a, b, c\}$ e $\Sigma_{3,o} = \{a, d\}$. Neste exemplo, vamos considerar apenas o canal de comunicação $ch_{1,2}$. Para tanto, sejam $\Sigma_o^{1,1} = \{b\}$, $\Sigma_o^{1,2} = \{c\}$, $\Sigma_o^{2,2} = \{a, c\}$ e $\Sigma_o^{2,1} = \{b\}$. Suponha que a sequência $s = abccddb$ tenha sido gerada pelo sistema e considere que não há atrasos no canal de comunicação $ch_{1,2}$, então os diagnosticadores locais D_1 e D_2 observam as sequências $s_{o_1} = P_{1,o}^o(s) = bccb$ e $s_{o_2} = P_{2,o}^o(s) = abccb$, respectivamente, em que $P_{i,o}^o : \Sigma_o^\star \to \Sigma_{i,o}^\star$, $i = 1, 2$.

Suponha agora que o atraso no canal de comunicação $ch_{1,2}$ seja $k_{1,2} = 2$, *i.e.*, um evento comunicado de D_1 para D_2 ou de D_2 para D_1 pode levar até duas ocorrências de eventos gerados por G para ser de fato observado em D_2 ou D_1, respectivamente. Essa situação é ilustrada na figura 2, em que a direção das setas indica a direção de evolução do sistema G, dos componentes G_1 e G_2 e dos diagnosticadores locais D_1 e D_2. Na situação ilustrada na figura 2, quando D_1 observa a primeira ocorrência do evento b, ele envia essa informação para D_2 e, por causa do atraso de comunicação, o evento c ocorre, sendo registrado por D_2 antes de b, fazendo com que a observação da sequência s em D_2 seja diferente de $s_{o_2} = P_{2,o}^o(s) = abccb$ e igual a $s'_{o_2} = acbcb$. Quando a segunda ocorrência do evento c é registrada por D_2, essa informação é enviada para D_1, entretanto, como $k_{1,2} = 2$ é definido em referência aos eventos gerados na planta G, enquanto c está sendo comunicado para D_1, duas ocorrências do evento d, que não é definido em G_1, são geradas pela planta G, o que faz com que D_1 observe corretamente a sequência de eventos. Neste exemplo, se após a segunda ocorrência do evento c, a única sequência possível de ser gerada pela planta é formada por dois eventos d sucessivos, o atraso de comunicação não teria efeito prático na observação de D_1 e, portanto, não deve ser considerado para o diagnóstico síncrono distribuído. □

É importante notar que, como é mostrado no exemplo 1, se o método proposto em Nunes et al. (2018) fosse diretamente aplicado à arquitetura síncrona distribuída, o resultado seria muito conservador, ou seja, mais possibilidades de observação pelos diagnosticadores locais seriam consideradas, apesar de que, na prática, isso não seria observado. Portanto, para levar esse efeito em consideração, é necessário uma conversão de referência do atraso da planta para os locais/componentes. Essa mudança de referencial é proposta na seção seguinte.

3.2 Conversão de referência do atraso máximo para o diagnóstico síncrono distribuído

Para realizar o diagnóstico síncrono distribuído sujeito a atrasos de comunicação, é necessário modelar a consequência do atraso nos modelos dos componentes, que são usados para gerar os diagnosticadores locais. Para tanto, nesta seção, propomos um método para converter o referencial do atraso de comunicação de eventos da planta global para os diagnosticadores locais.

Como a planta é a referência para os atrasos de comunicação, é possível que, dependendo do sistema, o valor de atraso máximo $k_{i,j}$ definido para o canal $ch_{i,j}$ não tenha efeitos práticos para os diagnosticadores locais D_i e D_j. Nesse caso, um valor menor do que $k_{i,j}$ deve ser considerado. Isso é feito analisando-se o comportamento global do sistema com o objetivo de identificar as situações em que o valor de $k_{i,j}$ pode ser menor, o que torna o diagnóstico menos conservador.

Com o objetivo de calcular os modelos dos componentes sujeitos a atrasos de comunicação de eventos, é necessário primeiro introduzir as seguintes funções:

- Projeção $P_i : \Sigma^\star \to \Sigma_i^\star$;
- função de pós-linguagem limitada em ν, $\ell : Q \times \Sigma \times \mathbb{N} \to \Sigma^\star$ definida como:
$$\ell(q, \sigma, \nu) = \{s \in \Sigma^\star : f(q, \sigma s)! \wedge \|s\| = \nu\}; \qquad (1)$$
- comprimento máximo de sequência $\mu : \Sigma^\star \to \mathbb{N}$ definida como:

$$\mu(A) = max(\|s\| : s \in A), \quad (2)$$

em que A é uma linguagem;

- função de renomeação de eventos $R_i^d : \Sigma_o^{i,j} \times k \to \Sigma_{o_d}^{i,j}$, $k \in \mathbb{N}$, $k \leq k_{i,j}$, que atribui a cada evento de $\sigma \in \Sigma_{i,o}$ o valor do atraso máximo corrigido com que σ pode ser observado:

$$R_i^d(\sigma, k) = \sigma_k. \quad (3)$$

A seguir, introduzimos um algoritmo que calcula os modelos G_{d_i}, $i \in 1, \ldots, r$ que correspondem a uma modificação dos autômatos G_i de tal forma a registrar o atraso máximo possível de ser observado por cada diagnosticador local D_i.

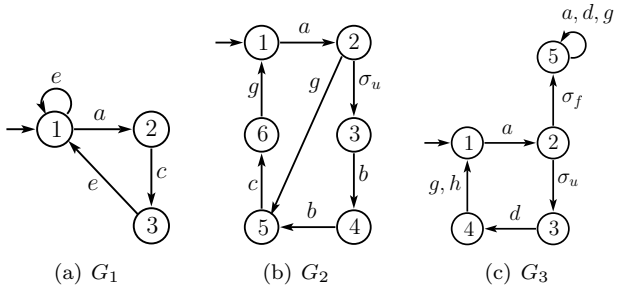
Algoritmo 1 Modelos dos componentes com registro de atraso máximo

Entradas: $G = (Q, \Sigma, f, q_0)$, $G_i = (Q_i, \Sigma_i, f_i, q_{0,i})$, $\Sigma_o^{i,j}$, $k_{i,j}$, $j \in \{1, \ldots, r\}$ e $i \neq j$.
Saída: $G_{d_i} = (Q_i, \Sigma_{i,u} \cup \Sigma_o^{i,i} \cup \Sigma_{o_d}^{i,j}, f_{d_i}, q_{0,i})$.

1: $\Sigma_{o_d}^{i,j} \leftarrow \emptyset$, $k \leftarrow 0$, $B \leftarrow \emptyset$.
2: **para todo** $j \in \{1, \ldots, r\}$, $j \neq i$ **faça**
3: **para todo** (q_i, σ, q_i') de G_i tal que $\sigma \in \Sigma_o^{i,j}$ **faça**
4: **para todo** (q, σ, q') de G em que q_i é a i-ésima coordenada de q **faça**
5: $B \leftarrow B \cup \ell(q, \sigma, k_{i,j})$.
6: $k \leftarrow \mu(P_i(B))$.
7: **se** $k \neq 0$ **então**
8: $\Sigma_{o_d}^{i,j} \leftarrow \Sigma_{o_d}^{j,i} \cup R_i^d(\sigma, k)$.
9: Defina $f_{d_i}(q_i, R_i^d(\sigma, k)) = f_i(q_i, \sigma)$.
10: **senão se** $k = 0$ **então**
11: $\Sigma_{o_d}^{i,j} \leftarrow \Sigma_{o_d}^{j,i} \cup \{\sigma\}$.
12: Defina $f_{d_i}(q_i, \sigma) = f_i(q_i, \sigma)$.
13: $B \leftarrow \emptyset$, $k \leftarrow 0$.
14: **para todo** $\sigma \notin \Sigma_{o_d}^{i,j}$ **faça** $f_{d_i}(q_i, \sigma) = f_i(q_i, \sigma)$.

A ideia por trás do algoritmo 1 é mapear as transições dos componentes locais que podem de fato ser observadas em ordem trocada por causa do atraso de comunicação. Para tanto, verifica-se na planta G quais situações permitem uma troca de observação em um diagnosticador local D_i construído a partir do componente G_i. O exemplo a seguir ilustra a construção dos autômatos G_{d_i}, $i = 1, \ldots, r$ utilizando-se o algoritmo 1.

Exemplo *2.* Considere novamente a arquitetura ilustrada na figura 1, em que a planta G é formada pelos componentes G_1, G_2 e G_3, apresentados na figura 3. O sistema completo $G = G_1 \| G_2 \| G_3$ pode ser visto na figura 4. Neste exemplo, $\Sigma_1 = \Sigma_{1,o} \dot{\cup} \Sigma_{1,u} = \{a, c, e\}$, $\Sigma_2 = \Sigma_{2,o} \dot{\cup} \Sigma_{2,u} = \{a, b, c, g, \sigma_u\}$ e $\Sigma_3 = \Sigma_{3,o} \dot{\cup} \Sigma_{3,u} = \{a, d, g, h, \sigma_u, \sigma_f\}$, em que $\Sigma_{1,o} = \{a, c\}$, $\Sigma_{1,u} = \{e\}$, $\Sigma_{2,o} = \{a, b, c, g\}$, $\Sigma_{2,u} = \{\sigma_u\}$, $\Sigma_{3,o} = \{a, d, g, h\}$ e $\Sigma_{3,u} = \{\sigma_u, \sigma_f\}$. Nesse cenário, consideramos que apenas os canais $ch_{1,2}$ e $ch_{2,3}$ são usados pelos diagnosticadores locais D_1, D_2 e D_3 para enviar informações sobre a observação de eventos. Sejam $\Sigma_o^{1,1} = \{c\}$, $\Sigma_o^{1,2} = \{a\}$, $\Sigma_o^{2,2} = \{a, b\}$, $\Sigma_o^{2,1} = \{c\}$, $\Sigma_o^{2,3} = \{g\}$, $\Sigma_o^{3,3} = \{d, g, h\}$ e $\Sigma_o^{3,2} = \{a\}$, ou seja, o evento a é observado pelo local de medição LM_2 e enviado por D_2 para D_1 e D_3, o evento c é observado por LM_1 e enviado de D_1 para D_2, e o evento g é observado por LM_3 e enviado de D_3 para D_2. Neste exemplo, considera-se que o atraso máximo nos canais $ch_{1,2}$ e $ch_{2,3}$ é de $k_{1,2} = 2$ e $k_{2,3} = 1$, respectivamente. Note que $k_{1,3} = 0$, já que não há troca de eventos entre os diagnosticadores locais D_1 e D_3.



(a) G_1 (b) G_2 (c) G_3

Figura 3. Modelos dos componentes G_1, G_2, e G_3 do exemplo 2.

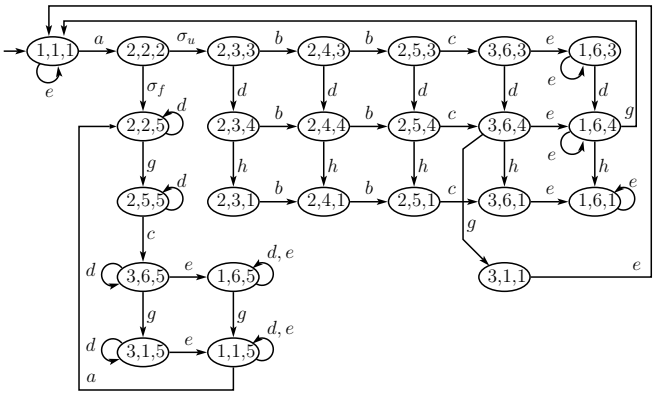
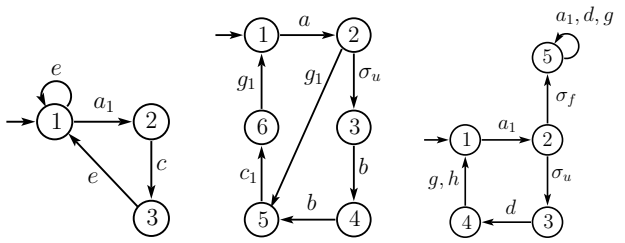


Figura 4. Modelo da planta $G = G_1 \| G_2 \| G_3$ do exemplo 2.

Aplicando-se o algoritmo 1 aos autômatos G_1, G_2 e G_3, obtém-se os autômatos G_{d_1}, G_{d_2} e G_{d_3} apresentados na figura 5. Para ilustrar o uso do algoritmo 1, considere a transição $(1, a, 2)$ do autômato G_1 da figura 3. Como o canal de comunicação $ch_{1,2}$, por onde o evento a é comunicado, tem atraso máximo $k_{1,2} = 2$, é necessário verificar todas as sequências s possíveis em G com comprimento igual a 2 após a ocorrência do evento a nos estados $q = (1, q_2, q_3)$, em que $q_2 \in Q_2$ e $q_3 \in Q_3$. Os estados de G cuja primeira coordenada é 1 e o evento a é possível são $(1, 1, 1)$ e $(1, 1, 5)$ e as sequências possíveis com comprimento $k_{1,2} = 2$ após a ocorrência do evento a a partir dos estados $(1, 1, 1)$ e $(1, 1, 5)$ são $s_1 = \sigma_u b, s_2 = \sigma_u d, s_3 = \sigma_f d, s_4 = \sigma_f g, s_5 = dd, s_6 = dg, s_7 = gc$ e $s_8 = gd$. Em seguida, na linha 6 do algoritmo 1, o valor de k é atualizado com o maior comprimento $\|P_1(s_l)\|$, $l = 1, \ldots, 8$, $P_1 : \Sigma^\star \to \Sigma_1^\star$, que nesse caso é igual a $\|P_1(s_7)\| = \|P_1(gc)\| = \|c\| = 1$. Como $k \neq 0$, o evento a que rotula a transição $(1, a, 2)$ em G_1 é renomeado para $a_k = a_1$, resultando na transição $(1, a_1, 2)$ em G_{d_1}. Isso significa que o atraso máximo de observação do evento a para o diagnosticador local D_1 nessa situação é igual a $1 < k_{1,2} = 2$. □

Observação 1. É importante observar que ao aplicar o algoritmo 1 em G_i, o conjunto $\Sigma_{o_d}^{i,j}$ pode ter $|\Sigma_{o_d}^{i,j}| = \sum_{j=1, j\neq i}^{r} k_{i,j} \times |\Sigma_o^{i,j}|$ eventos no pior caso em que todos os eventos comunicáveis são renomeados com um valor diferente de atraso, em que $|.|$ denota cardinalidade. □

Uma vez obtido os autômatos G_{d_i} utilizando-se o algoritmo 1, o método apresentado em Nunes et al. (2018) pode ser usado para construir os autômatos $G_{d_{i,j}}^k$ cuja linguagem gerada representa todas as possíveis observações de sequências geradas pela planta no diagnosticador local D_1



(a) G_{d_1} (b) G_{d_2} (c) G_{d_3}

Figura 5. Modelos dos componentes com registro de atraso máximo de eventos G_{d_1}, G_{d_2}, e G_{d_3} do exemplo 2.

devido aos atrasos de comunicação no canal $ch_{i,j}$. Assim, a implementação de diagnosticadores locais robustos ao atraso de comunicação de eventos observados pode ser feita usando um estimador de estados dos autômatos livres de falha $G^k_{N_{d_{i,j}}}$ obtidos a partir de $G^k_{d_{i,j}}$.

3.3 Diagnosticabilidade distribuída sujeita a atrasos de comunicação de eventos

Em Nunes et al. (2018), um algoritmo para o cálculo de autômatos Δ_i que modelam todas as possíveis trocas de observação por conta de atrasos na comunicação de eventos para a arquitetura descentralizada é proposto. Para considerar a arquitetura síncrona distribuída, o algoritmo apresentado em Nunes et al. (2018) pode ser usado com pequenas modificações. Para tanto, vamos associar para cada evento $\sigma_k \in \Sigma^{i,j}_{o_d}$ o evento $\sigma_k^{s_i}$ que representa a observação com sucesso do evento σ, comunicado pelo diagnosticador local D_j, com possibilidade de atraso máximo k, para o diagnosticador D_i. Vamos definir o conjunto $\Sigma^s_{o_{i,j}} = \{\sigma_k^{s_i} : \sigma_k \in \Sigma^{i,j}_{o_d}\}$ como o conjunto de eventos observáveis pelo diagnosticador local D_i que são comunicados pelo canal $ch_{i,j}$.

O método usado em Nunes et al. (2018) para descrever todas as possíveis trocas de observação de eventos pode ser adaptado para o caso síncrono distribuído calculando-se os autômatos Δ_i, $i = \{1, \ldots, r\}$, levando-se em consideração o máximo atraso de observação k de todos os eventos $\sigma_k^{s_i} \in \Sigma^s_{o_{i,j}}$. O autômato que modela o comportamento do componente local G_i sujeito a atraso de comunicação de eventos é dado por $G'_{d_i} = G_{d_i} \| \Delta_i = (Q'_{d_i}, \Sigma_{i,u} \cup \Sigma^{i,i}_o \cup \Sigma^s_{o_{i,j}} \cup \Sigma^{i,j}_o, f'_{d_i}, q'_{0,d_i})$. Uma vez obtido G'_{d_i}, os eventos $\sigma_k^{s_i}$ podem ser renomeados utilizando-se a função $\phi : \Sigma_{d_i} \to \Sigma_i$, em que $\Sigma_{d_i} = \Sigma_{i,u} \cup \Sigma^{i,i}_o \cup \Sigma^s_{o_{i,j}} \cup \Sigma^{i,j}_o$, tal que $\phi(\sigma_k^{s_i}) = \sigma$, para $\sigma_k^{s_i} \in \Sigma^s_{o_{i,j}}$ e $\phi(\sigma) = \sigma$, para $\sigma \in \Sigma_{d_i} \setminus \Sigma^s_{o_{i,j}}$. Assim, o autômato $G_{\delta_i} = (Q_{\delta_i}, \Sigma_i, f_{\delta_i}, q_{0,\delta_i})$, que modela o comportamento do componente G_i sujeito a atrasos de observações, pode ser obtido fazendo-se $Q_{\delta_i} = Q'_{d_i}$, $\Sigma_i = \Sigma_{i,u} \cup \Sigma^{i,i}_o \cup \Sigma^{i,j}_o$, $f_{\delta_i}(q, \phi(\sigma)) = f'_{d_i}(q, \sigma)$ e $q_{0,\delta_i} = q'_{0,d_i}$.

A definição da diagnosticabilidade síncrona distribuída sujeita a atrasos de comunicação de eventos (DSDSA) é definida a seguir.

Definição *2.* (DSDSA). Sejam $G_\delta = \|^r_{i=1} G_{\delta_i}$, $L(G_\delta) = L_\delta$, e $L_{N_\delta} = L(G_{N_\delta})$. Seja $L_{F_\delta} = L_\delta \setminus L_{N_\delta}$. Seja L_{N,δ_i} a linguagem gerada por G_{N,δ_i}, em que G_{N,δ_i} é o autômato que modela o comportamento livre de falha de G_{δ_i}, para $i = 1, \ldots, r$. Então, L_δ é dita ser diagnosticável sincronamente de forma distribuída sujeita a atrasos de comunicação em relação a L_{N,δ_i}, $P_o : \Sigma^\star \to \Sigma^\star_o$, $P^o_{i,o} : \Sigma^\star_o \to \Sigma^\star_{i,o}$, $P^i_{i,o} : \Sigma^\star_i \to \Sigma^\star_{i,o}$, L_{F_δ} se

$$(\exists z \in \mathbb{N})(\forall s \in L_{F_\delta})(\forall st \in L_{F_\delta}, \|t\| \geq z) \Rightarrow$$
$$P_o(st) \notin \bigcap_{i=1}^{r} P^{o^{-1}}_{i,o} \left[P^i_{i,o}(L_{N,\delta_i}) \right].$$

□

De acordo com a definição 2, a linguagem L_δ é diagnosticável, se toda sequência st de falha de comprimento arbitrariamente longo após a falha não tiver a mesma projeção em $\Sigma_o = \cup^r_{i=1} \Sigma_{i,o}$ do que outra sequência que pertença à linguagem gerada pela composição paralela dos observadores de G_{N,δ_i} (Cabral and Moreira, 2020).

Uma vez que os modelos dos componentes locais sejam calculados, a verificação da DSDSA pode ser feita de forma direta seguindo o método apresentado em Cabral and Moreira (2020) para a verificação da diagnosticabilidade síncrona descentralizada. Para tanto, ao invés de se utilizar os modelos dos componentes do sistema G_i, os autômatos G_{δ_i} devem ser usados como entradas do algoritmo de verificação apresentado em Cabral and Moreira (2020).

4. CONCLUSÃO

Neste trabalho, o diagnóstico síncrono distribuído robusto a atrasos de comunicação de eventos é proposto. O método consiste em modificar os modelos dos componentes do sistema de tal forma a incorporar o efeito do atraso de comunicação dos eventos entre diagnosticadores locais. Por causa dos possíveis atrasos na comunicação dos eventos, é necessário definir a diagnosticabilidade síncrona distribuída sujeita a atrasos de comunicação de eventos. Como trabalho futuro, será investigado o problema de atraso de comunicação para o protocolo proposto em Veras et al. (2018), em que, além da observação de eventos, estimativas de estado do comportamento livre de falha dos componentes também são comunicados.

REFERÊNCIAS

Cabral, F.G. and Moreira, M.V. (2020). Synchronous diagnosis of discrete-event systems. *IEEE Transactions on Automation Science and Engineering*, 17(2), 921–932.

Cassandras, C. and Lafortune, S. (2008). *Introduction to Discrete Event System*. Springer-Verlag New York, Inc., Secaucus, NJ.

Contant, O., Lafortune, S., and Teneketzis, D. (2006). Diagnosability of discrete event systems with modular structure. *Discrete Event Dynamic Systems: Theory And Applications*, 16(1), 9–37.

Debouk, R., Lafortune, S., and Teneketzis, D. (2000). Coordinated decentralized protocols for failure diagnosis of discrete event systems. *Discrete Event Dynamic Systems: Theory and Applications*, 10(1), 33–86.

Debouk, R., Malik, R., and Brandin, B. (2002). A modular architecture for diagnosis of discrete event systems. In *41st IEEE Conference on Decision and Control*, 417–422. Las Vegas, Nevada USA.

Gilchrist, A. (2016). *Industry 4.0: The Industrial Internet of Things*. Berkeley, CA, USA: Apress, 2016, 1st ed edition.

Keroglou, C. and Hadjicostis, C.N. (2014). Distributed diagnosis using predetermined synchronization strategies. In *Proceedings of 53rd IEEE Conference on Decision and Control (CDC)*, 5955–5960. Los Angeles, CA, USA.

Keroglou, C. and Hadjicostis, C.N. (2018). Distributed Fault Diagnosis in Discrete Event Systems via Set Intersection Refinements. *IEEE Transactions on Automation Science and Engineering*, 63(10), 3601–3607.

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.

Nunes, C.E., Moreira, M.V., Alves, M.V., Carvalho, L.K., and Basilio, J.C. (2018). Codiagnosability of networked discrete event systems subject to communication delays and intermittent loss of observation. *Discrete Event Dynamic Systems*, 28(2), 215–246.

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.

Qiu, W. and Kumar, R. (2008). Distributed diagnosis under bounded-delay communication of immediately forwarded local observations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 38(3), 628–643.

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.

Schmidt, K.W. (2013). Verification of modular diagnosability with local specifications for discrete-event systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 43(5), 1130–1140.

Tripakis, S. (2004). Decentralized control of discrete-event systems with bounded or unbounded delay communication. *IEEE Transactions on Automatic Control*, 49(9), 1489–1501.

Veras, M.Z.M., Cabral, F.G., and Moreira, M.V. (2018). Distributed synchronous diagnosability of discrete-event systems. *14th IFAC Workshop on Discrete Event Systems*, 51(7), 88–93.

Wang, Y., Yoo, T.S., and Lafortune, S. (2007). Diagnosis of discrete event systems using decentralized architectures. *Discrete Event Dynamic Systems: Theory And Applications*, 17, 233–263.