DOI: 10.48011/asba.v2i1.1283

Uma nova abordagem do mapeamento de sensores para obtenção de modelos de sistemas eventos discretos sujeitos a falhas ⋆

Victor Hugo Cruz * **Lilian K. Carvalho** * **João Carlos Basilio** *

* *COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, 21.949-900, Rio de Janeiro, RJ, Brasil (e-mails: hugo.almeida@marinha.mil.br, lilian@dee.ufrj.br, basilio@dee.ufrj.br)*

Abstract: The first step in the design of fault diagnosis for industrial systems is to obtain a model that describes both normal and fault behaviors. In the context of fault diagnosis of discrete event systems (DES), Sensor Mapping (SM) is a tool used to obtain automaton models. However, its application, besides being limited to states associated with a single sensor reading, can also lead to incorrect estimation in real time of the states of the monitored system, since the information associated with commands and sensor readings are both grouped in a singe event. In this paper, a new method is proposed, the Sensor Mapping Based on Language Specification (SM-BLE) is proposed to obtain automata that model the behavior of the sensors of the system. The idea behind the proposed method is to change the SM by using rules usually adopted in the specification of the admissible language required in the supervisor control of DES.

Resumo: O primeiro estágio do projeto de diagnosticadores de falhas para sistemas industriais consiste na obtenção de um modelo que descreva tanto o comportamento normal como o de falha. No contexto de diagnose de falhas de sistemas a eventos discretos (SED), o Mapeamento de Sensores (MS) é uma ferramenta utilizada para a obtenção de modelos por autômatos. Porém a sua aplicação, além de ser limitada a estados associados a uma única leitura de sensores, pode também levar à incorreta estimação em tempo real dos estados do sistema monitorado, tendo em vista que as informações associadas a comandos e leituras de sensores são agrupadas em um mesmo evento. Neste artigo um novo método, o Mapeamento de Sensores Baseado na Especificação de Linguagens (MS-BEL), será proposto para a obtenção de autômatos que modelam o comportamento dos sensores do sistema. A ideia por trás do método proposto é alterar o MS utilizando-se regras usualmente adotadas na especificação da linguagem admissível necessária para o projeto de controle supervisório de SEDs.

Keywords: Discrete event systems, failure diagnosis, automata, sensor map.

Palavras-chaves: Sistemas a eventos discretos, diagnose de falhas, autômatos, mapa de sensores

1. INTRODUÇÃO

Embora sistemas industriais automatizados possuem componentes inteligentes com funções de controlar e monitorar a planta em tempo real, o aumento da complexidade desses sistemas e da exigência em relação aos requisitos de desempenho e segurança dos processos têm feito com que grandes esforços sejam realizados em pesquisas e desenvolvimentos de sistemas sofisticados capazes de diagnosticar falhas.

Falhas são ocorrências capazes de provocar distúrbios significativos ao sistema, alterando o seu comportamento para um ponto longe da sua faixa normal de operação, no qual os dispositivos de controle não conseguem regular eficientemente as variáveis. Se essa situação não for corrigida por uma ação adequada, a operação poderá entrar numa situação emergencial com consequências catastróficas. Para tanto, torna-se necessário, em primeiro lugar, diagnosticar a falha durante o funcionamento do sistema.

A implementação de um sistema para a diagnose de falhas está baseada em três pilares: a detecção, o isolamento e a identificação da falha. A detecção é uma funcionalidade que decide se o sistema está operando em condições normais ou se alguma falha ocorreu. Se determinada falha ocorreu, o isolamento localiza os componentes do sistema que falharam. Finalmente, a identificação está preocupada com a natureza da falha (Zaytoon and Lafortune, 2013).

Sampath et al. (1996) consideraram o problema do desenvolvimento de modelos a eventos discretos para a diagnose de falhas, onde foi apresentado um procedimento sistemático para gerar um modelo que captura a interação entre os componentes e incorpora nele as informações provenientes dos sensores. Esse método foi denominado de Mapeamento de Sensores (MS). O MS renomeia os eventos do sistema no processo de tal forma que os sinais de atuadores provenientes de controladores, fiquem agrupados com os sinais

⋆ Este trabalho foi, em parte, financiado pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), código de financiamento 001, pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processos número 309652/2017-0 e pelo Instituto de Pesquisas da Marinha (IPqM).

emitidos pelos sensores da planta, o que pode acarretar nos seguintes problemas: *(i)* diagnose prematura e errônea de falhas, conforme mostrado em Cruz (2020); *(ii)* impossibilidade da aplicação de resultados recentes do problema de escolha de sensores (Debouk et al., 2002; Basilio et al., 2012; Santoro et al., 2017), nos quais são supostos que os eventos gerados pelos sensores estão desacoplados dos eventos de comando. Torna-se, portanto, necessário modificar o MS com vistas a dissociar os eventos de comando dos eventos de leitura.

Neste artigo, é proposto o método de Mapeamento de Sensores Baseado na Especificação de Linguagens (MS-BEL). O MS-BEL está baseado nas seguintes premissas: (i) dissociação entre os sinais dos sensores e os comandos enviados para os atuadores da planta; (ii) simplicidade na implementação prática dos modelos obtidos. O MS-BEL combina o mapa de sensores do MS com as especificações de linguagens, usualmente utilizadas na determinação da linguagem admissível para controle supervisório de SEDs, para obter os autômatos que modelam o comportamento dos sensores e utiliza a operação de composição paralela para dissociar os eventos de comando dos eventos associados à leitura dos sensores dos componentes individuais.

Este artigo está estruturado da seguinte forma. Na seção 2 são revistos os principais conceitos de SEDs, diagnose de falhas e Mapeamento de Sensores. A apresentação do MS-BEL é realizada na seção 3, juntamente com a sua arquitetura, a descrição de seus módulos. Na seção 4, o método proposto será ilustrado utilizando um sistema de aquecimento. Finalmente, na seção 5, são apresentadas as conclusões deste artigo.

2. FUNDAMENTOS TEÓRICOS

Um autômato G é definido pela quíntupla $G = (X, \Sigma, f, \Gamma, x_0)$, em que, X é o conjunto de estados, Σ é o conjunto finito de eventos, $f : X \times \Sigma \to X$ é a função de transição de estados, $\Gamma : X \to 2^{\Sigma}$ é a função de eventos ativos e $x_0 \in X$ é o estado inicial. A função de transição de estados pode ser estendida para $f : X \times \Sigma^* \to X$, em que Σ^* representa o Fecho-de-Kleene de Σ, pela recursão $f(x, \varepsilon) = x$ e $f(x, s\sigma) = f(f(x, s), \sigma)$, $s \in \Sigma^*$ e $\sigma \in \Sigma$. A linguagem gerada por G é definida como $L(G) = \{s \in \Sigma^* : f(x_0, s)!\}$, em que $f(x_0, s)!$ significa que $f(x_0, s)$ é definida, i.e., existe $x \in X$ tal que $f(x_0, s) = x$. Dados dois autômatos G_1 e G_2, denota-se por $G_1 \| G_2$ a composição paralela ou síncrona entre G_1 e G_2 (Cassandras and Lafortune, 2008, p. 79)

O *fecho do prefixo* de uma sequência s será denotado por $\bar{s}$, sendo definido como $\bar{s} = \{u \in \Sigma^* : (\exists \upsilon \in \Sigma^*)[u\upsilon = s]\}$. A *pós-linguagem* de L após uma sequência s é definida por $L/s = \{t \in \Sigma^* : st \in L\}$. Uma linguagem L é dita viva se $\forall s \in L, \exists \sigma \in \Sigma : s\sigma \in L$. A *projeção natural*, ou simplesmente *projeção*, é definida sobre dois conjuntos de eventos Σ_a e Σ_b, em que $\Sigma_b \subseteq \Sigma_a$, da seguinte forma: $P : \Sigma_a^* \to \Sigma_b^*$, em que $P(\varepsilon) = \varepsilon$; $P(\sigma) = \sigma$ se $\sigma \in \Sigma_b$; $P(\sigma) = \varepsilon$ se $\sigma \in \Sigma_a \setminus \Sigma_b$, e; $P(s\sigma) = P(s)P(\sigma)$ para $s \in \Sigma_a^*$, $\sigma \in \Sigma_a$. A *projeção inversão* é definida por $P^{-1} : \Sigma_b^* \to 2^{\Sigma_a^*}$, em que $P^{-1}(t) = \{s \in \Sigma_a^* : P(s) = t\}$. O *fecho do prefixo*, a *projeção* e a *projeção inversa* são estendidas sobre uma linguagem L aplicando-as a todas as sequências de L.

Um autômato não determinístico G_{nd} é definido por uma quíntupla $G_{nd} = (X, \Sigma \cup \{\varepsilon\}, f_{nd}, \Gamma, X_0)$ em que todos os componentes são definidos como no autômato determinístico, exceto i) $f_{nd} : X \times \Sigma \cup \{\varepsilon\} \to 2^X$, ou seja, $f_{nd}(x, \sigma) \subseteq X$ quando for definida; ii) $X_0 \subseteq X$, isto é, o estado inicial pode ser um conjunto. Para obter a linguagem gerada e marcada por G_{nd} é necessário estender o domínio de f_{nd} para $X \times \Sigma^*$, sendo a função estendida denotada por f_{nd}^{ext}. Para tanto, é necessário definir o alcance ε, denotado por $\varepsilon R(x)$, como o conjunto de todos os estados alcançados a partir de x, incluindo x, por transições rotuladas por ε. Para um conjunto $B \in 2^X$, tem-se que $\varepsilon R(B) = \cup_{x \in B} \varepsilon R(x)$. Assim, pode-se definir a função estendida f_{nd}^{ext} como: i) $f_{nd}^{ext}(x, \varepsilon) = \varepsilon R(x)$; ii) $f_{nd}^{ext}(x, s\sigma) = \varepsilon R[\{z : z \in f_{nd}(y, \sigma)$ para algum estado $y \in f_{nd}^{ext}(x, s)\}]$, $s \in \Sigma^*$ e $\sigma \in \Sigma$.

Particionando-se o conjunto de eventos em $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_o e Σ_{uo} denotam, respectivamente, os conjuntos dos eventos observáveis e não-observáveis, e definindo-se o alcance não-observável como $UR(x, \Sigma_o) = \{x' \in X : (\exists s \in \Sigma_{uo}^*)[f(x, s) = x']\}$ e, se $Y \in 2^X$, então $UR(Y, \Sigma_o) = \cup_{x \in Y} UR(x, \Sigma_o)$, é possível definir o observador do autômato G da seguinte forma: $G_{obs} = Obs(G, \Sigma_o) = (X_{obs}, \Sigma_o, f_{obs}, \Gamma_{obs}, x_{0,obs})$, sendo $X_{obs} \in 2^X$, $f_{obs}(x_{obs}, \sigma) = \cup_{x \in x_{obs} \wedge f(x,\sigma)!} UR(f(x, \sigma), \Sigma_o)$, $\Gamma_{obs}(x_{obs}) = \cup_{x \in x_{obs}} \Gamma(x) \cap \Sigma_o$, $x_{0,obs} = UR(x_0, \Sigma_o)$. Note que, $L(G_{obs}) = P_o(L(G_{obs}))$, em que $P_o : \Sigma^* \to \Sigma_o^*$.

Seja $\Sigma_f \subset \Sigma_{uo} = \{\sigma_f\}$ o conjunto de eventos de falha. Por abuso de notação, usa-se $\sigma \in s$ para representar que s possui o evento σ. Nesse sentido, uma sequência s é dita de falha se $\sigma_f \in s$; em contrapartida, uma sequência s tal que $\sigma_f \notin s$ é denominada normal.

Uma linguagem L é dita ser diagnosticável se for possível detectar a ocorrência do evento de falha após um número finito de observações de eventos após a ocorrência da falha. A definição formal da diagnosticabilidade de uma linguagem é dada a seguir. Para tanto, seja $\Psi(\sigma_f)$ o conjunto composto por todas as sequências de $L(G)$ que terminam como evento σ_f.

Definição 1. (Diagnosticabilidade(Sampath et al., 1995)). Uma linguagem L, prefixo fechada e viva, será diagnosticável em relação à projeção $P_o : \Sigma^* \to \Sigma_o^*$ e ao conjunto de falhas Σ_f se:

$$(\exists z \in \mathbb{N})(\forall s \in \Psi(\Sigma_f))(\forall t \in L/s)$$
$$\left(||t|| \geq z \Rightarrow (\forall w \in P_o^{-1}(P_o(st)) \cap L)(\Sigma_f \in w)\right).$$

2.1 Mapeamento de Sensores

Na modelagem de sistemas dinâmicos utilizando modelos a eventos discretos é possível obter o modelo global (monolítico) a partir da composição paralela dos autômatos $G_i = (X_i, \Sigma_i, f_i, \Gamma_i, x_{0,i})$, $i = 1, ..., N$, que modelam os componentes individuais (módulos), sendo o comportamento integrado desses componentes modelado pelo autômato $G = \|_{i=1}^{N} G_i = (X, \Sigma, f, \Gamma, x_0)$, em que $\Sigma = \cup_{i=1}^{N} \Sigma_i$.

Do ponto de vista prático, os estados de um sistema podem ser caracterizados considerando-se as indicações de todos os sensores existentes no sistema. Para tanto, vamos supor que a planta possua M sensores, e que cada sensor possua p_j saídas possíveis. Seja $Y_j = \{y_{j1}, y_{j2}, \ldots, y_{jp_j}\}$,

$j = 1, \ldots, M$, conjuntos formados por todas as saídas possíveis do sensor-j, isto é, y_{jk}, $k = 1, \ldots, p_j$ representa a k-ésima saída possível do sensor j . Considere a seguinte função:

$$h_j : X \to Y_j, j = 1, ..., M,$$

que associa a cada estado $x_i \in X$, $i = 1, ..., |X|$, um valor $h_j(x_i) \in Y_j$. Definindo-se $Y = \times_{j=1}^{M} Y_j$, em que $\times$ denota produto cartesiano, é possível, então, introduzir uma nova função denominada Mapa de Sensores (MDS):

$$h : X \to Y,$$

que, a cada $x_i \in X$, associa-se uma tupla $h(x_i) = \big(h_1(x_i), h_2(x_i), \ldots, h_M(x_i))\big) \in Y$. Note que a função h_j define cada uma das células da tabela de MDS da figura 1, enquanto que a função h define cada uma das linhas da tabela.

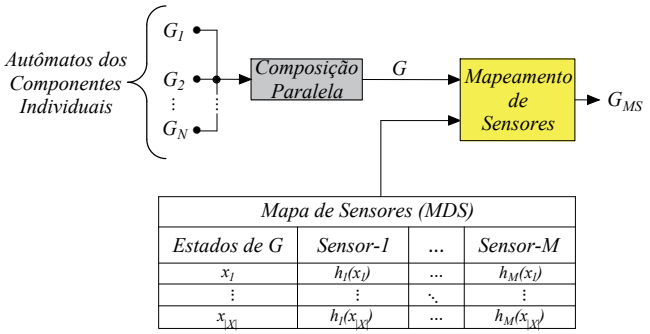


Figura 1. Arquitetura da modelagem de SED utilizando o MS.

O Mapeamento de Sensores (MS) é um procedimento sistemático que combina as informações provenientes dos sensores do processo com os dados de seus componentes individuais, resultando em um autômato que modela o comportamento completo do sistema (Sampath et al., 1996), denominado por G_{MS}. O MS modifica o procedimento de transformação em autômato de *Mealy* (Mealy, 1955) para considerar o caso em que o sistema possui eventos não observáveis. Assim, dada uma transição $x \xrightarrow{\sigma} y$ de G, o MS realiza as seguintes alterações (Sampath et al., 1996):

(1) Se o evento σ for observável (tipicamente um evento de comando), a respectiva transição deve ser renomeada em G_{MS} como $\langle \sigma, h(y) \rangle$, resultando na transição $x \xrightarrow{\langle \sigma, h(y) \rangle} y$. O novo evento $\langle \sigma, h(y) \rangle$ é observável em Σ_{MS};

(2) Se σ não for observável e se $h(x) \neq h(y)$, então adiciona-se um novo estado $\langle\langle x, \sigma, y \rangle\rangle$ ao conjunto X, e substitui-se a transição $x \xrightarrow{\sigma} y$ pelas seguintes transições:

$$x \xrightarrow{\sigma} \langle\langle x, \sigma, y \rangle\rangle \text{ e } \langle\langle x, \sigma, y \rangle\rangle \xrightarrow{\langle h(x) \to h(y) \rangle} y; e$$

(3) Se $\sigma \in \Sigma_{uo}$ e $h(x) = h(y)$, então nenhuma alteração deve ser realizada na transição $x \xrightarrow{\sigma} y$.

O procedimento do MS está descrito no algoritmo 1. Note que ao final do algoritmo, as informações dos M sensores estarão integradas aos eventos dos componentes individuais do sistema, resultando no autômato G_{MS}, conforme ilustrado na figura 1.

Algoritmo 1: Mapeamento de Sensores

Entradas:
$G = (X, \Sigma, f, \Gamma, x_0)$, em que $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$
MDS: mapa com as informações dos sensores
Saída:
$G_{MS} = (X_{MS}, \Sigma_{MS}, f_{MS}, \Gamma_{MS}, x_{0,MS})$
1: $G_{MS} \leftarrow G$
2: **para** cada transição $x \xrightarrow{\sigma} y \in G_{MS}$ **faça**
3: **se** $\sigma \in \Sigma_o$ **então**
4: $\sigma \leftarrow \langle \sigma, h(y) \rangle$
5: **se** $\sigma \notin \Sigma_o$ e $h(x) \neq h(y)$ **então**
6: $x_{new} \leftarrow \langle\langle x, \sigma, y \rangle\rangle$
7: $X_{MS} \leftarrow X_{MS} \cup \{x_{new}\}$
8: $f_{MS}(x, \sigma) \leftarrow x_{new}$
9: $f_{MS}(x_{new}, \langle h(x) \to h(y) \rangle) \leftarrow y$
10: **retorna** G_{MS}

Note que no autômato resultante da aplicação do MS, as informações de comando são agrupadas com os sinais dos sensores em um único evento. Isso pode provocar a diagnose prematura e errônea de falhas, conforme mostrado em Cruz (2020). Além disso, no problema de escolha de sensores (Debouk et al., 2002; Basilio et al., 2012; Santoro et al., 2017), é suposto que os eventos gerados pelos sensores estão desacoplados dos eventos de comando. Portanto, esses métodos não podem ser diretamente aplicadas a um autômato que tenha sido obtido utilizando-se o MS. Torna-se, portanto, necessário modificar o MS de tal forma a dissociar os eventos de comando dos eventos de leitura.

3. MAPEAMENTO DE SENSORES BASEADO NA ESPECIFICAÇÃO DE LINGUAGENS

O Mapeamento de Sensores Baseado na Especificação de Linguagens (MS-BEL), assim como o MS, considerado na seção 2.1, é um método que tem como objetivo acrescentar as informações dos sensores do processo à modelagem de SEDs. A ideia por trás do MS-BEL é construir autômatos que modelem o comportamento dos sensores para serem integrados ao modelo através de uma composição paralela, promovendo assim, a dissociação entre os eventos de comando e de leitura dos sensores dos componentes individuais.

Em diversos sistemas práticos, nem sempre é possível associar uma única saída do sensor para cada estado do sistema. Isso é consequência tanto da falta de informações sobre o comportamento dos componentes quanto do grau de abstração da modelagem realizada e do uso de sensores virtuais (Sampath, 2001). Para contornar esse problema, vamos definir a função

$$\eta_j : X \to 2^{Y_j}$$

que associa a cada estado x_i, às possíveis saídas para o sensor-j, $j = 1, ..., M$, ou seja, $\eta_j(x_i) = \{h_j^1(x_i), h_j^2(x_i), ..., h_j^k(x_i), ..., h_j^{p_{ij}}(x_i)\} \in 2^{Y_j}$, em que $h_j^k(x_i)$ corresponde à k-ésima saída possível para o sensor-j e p_{ij} é o número de mapeamentos possíveis do sensor-j para o estado x_i. Com base na função η_j, pode-se, então, definir o Mapa de Sensores Estendido (MDSE) como o conjunto $MDSE = \{(x_i, \eta_j(x_i)) : i = 1, \ldots, |X|, j = 1, \ldots, M\}$, que pode ser sintetizado da forma apresentada na figura 2.

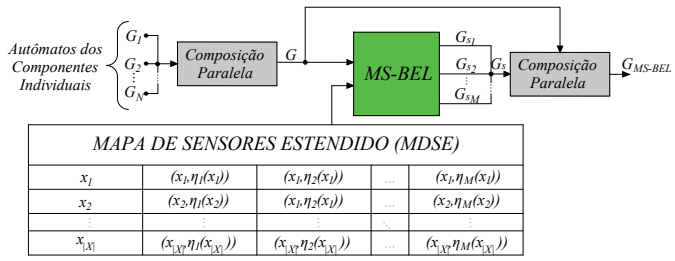


Figura 2. Arquitetura da modelagem modular utilizando o MS-BEL.

Exemplo 1. Considere o diagrama de um sistema composto por um tanque, uma válvula, um transdutor de nível e um sensor de vazão, ilustrado na figura 3. Suponha que a vazão de entrada F_i varie ao longo do tempo e que, quando a válvula V_1 estiver totalmente aberta, a vazão de saída F_o será maior do que F_i.

Suponha a seguinte abstração do sinal do transdutor de nível (sensor-1): *nível estável* (NE), *nível subindo* (NS) e *nível descendo* (ND), logo $Y_1 = \{NE, NS, ND\}$. Já o sensor de vazão (sensor-2) indica se há fluxo (F) ou não (NF) na tubulação; assim, $Y_2 = \{F, NF\}$.

A válvula V_1 pode ser modelada pelo autômato G_{V_1} da figura 4, que possui três estados: *VPA* (válvula parcialmente aberta), *VF*(válvula fechada) e *VA* (válvula aberta); e três eventos: *ap* (abrir parcialmente), *fv* (fechar válvula completamente) e *av* (abrir válvula totalmente).

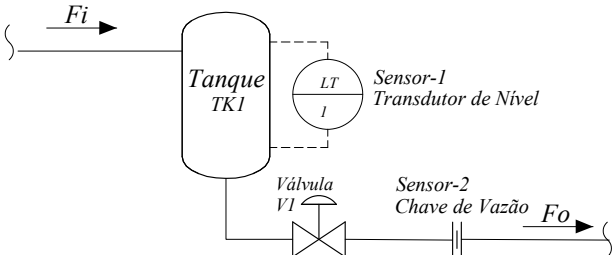


Figura 3. Diagrama do sistema do exemplo 1.

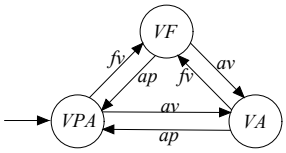


Figura 4. Autômato G_{V_1} que modela a válvula.

Note, neste exemplo, que $N = 1$, isto é, o sistema é composto por apenas um módulo e, portanto, $G = G_{V_1}$. Dessa forma, as saídas dos sensores serão mapeadas diretamente de acordo com os estados de G_{V_1}. Utilizando, inicialmente, o MDSE para o sensor de nível, obtém-se: $\eta_1(VA) = \{ND, NE\}$; $\eta_1(VF) = \{NS, NE\}$; $\eta_1(VPA) = \{ND, NS, NE\}$. Note que para os estados *VA*, *VF* e *VPA* não é possível definir unicamente o comportamento do nível; no primeiro caso, o nível do tanque pode descer ou permanecer estável (esse último, quando o nível atingir a cota zero e a vazão de entrada for igual a zero), no segundo caso, o nível do tanque pode subir ou ficar estável e no último caso, o sensor poderá fornecer a informação de que o nível está subindo, descendo ou estável. Já para o sensor de vazão, $\eta_2(VA) = \{F, NF\}$, $\eta_2(VF) = \{NF\}$, e $\eta_2(VPA) = \{F, NF\}$, conforme mostrado na tabela 1. O MDSE completo está representado na tabela 2. ■

Tabela 1. $\eta_j(x_i)$, $j = 1, 2$, $x_i \in \{VA, VF, VPA\}$ para o exemplo 1.

Estados de G_{V_1}	$\eta_1(x_i)$ (nível)	$\eta_2(x_i)$ (vazão)
VA	$\{ND, NE\}$	$\{F, NF\}$
VF	$\{NS, ND\}$	$\{NF\}$
VPA	$\{ND, NS, NE\}$	$\{F, NF\}$

Tabela 2. MDSE para o exemplo 1.

Estados de G_{V_1}	$(x_i, \eta_1(x_i))$	$(x_i, \eta_2(x_i))$
VA	$(VA, \{ND, NE\})$	$(VA, \{F, NF\})$
VF	$(VF, \{NS, ND\})$	$(VF, \{NF\})$
VPA	$(VPA, \{ND, NS, NE\})$	$(VPA, \{F, NF\})$

A arquitetura da modelagem modular de um SED utilizando o MS-BEL está ilustrada na figura 2. As entradas do MS-BEL são o autômato G e o MDSE. Para cada sensor-j, o MS-BEL retorna como saída o autômato G_{s_j} que modela o comportamento de G com as informações do sensor-j. O conjunto de todos os autômatos que possuem as informações dos sensores do sistema será denotado por: $G_s = \{G_{s_1}, G_{s_2}, ..., G_{s_M}\}$. O MS-BEL é dividido internamente em dois módulos, decomposição e criação de estados, que serão descritos a seguir.

3.1 Decomposição de estados

O módulo decomposição de estados tem a função de separar os estados de G mapeados com mais de um elemento do conjunto Y_j, ou seja, decompor o estado x_i para o qual $|\eta_j(x_i)| > 1$ em $|\eta_j(x_i)|$ estados, cada um com uma indicação diferente do sensor-j. Esse módulo é implementado pelo algoritmo 2, que possui como entradas, o autômato G, o mapa MDSE e o índice j do sensor, e como saídas, o autômato G_{db_j} cujos estados foram decompostos e o novo mapa MDS_{db_j} para o sensor-j.

Algoritmo 2 funciona da seguinte forma. Nos passos 1 e 2, G_{db_j} e MDS_{db_j} são inicializados com G e MDSE, respectivamente. Em seguida, no passo 3, inicia-se o laço para percorrer todos os estados de G. Se $|\eta_j(x_i)| > 1$, então será executada a decomposição do estado x_i. Nessa decomposição, para o k-ésimo elemento pertencente ao conjunto $\eta_j(x_i)$ é criado um novo estado $x_{new} = \langle\langle x_i, h_j^k(x_i)\rangle\rangle$ (passo 7), o qual é adicionado aos conjuntos X_{split} (passo 8) e ao X_{db_j} (passo 9). Em seguida, o mapa MDS_{db_j} recebe o novo par ordenado $(x_{new}, h_j^k(x_i))$ no passo 10. Entre passos 11 e 18, as transições do estado x_i do autômato G_{db_j} são copiadas para os estados x_{new} em G_{db_j}. Essa cópia das transições acontece em duas etapas: na primeira, quando a cópia das transições $x_i \xrightarrow{\sigma} y \in G_{db_j}$ é realizada, duas condições são observadas: (i) se $x_i = y$ (passo 12), será criado um autolaço no autômato G_{db_j} com a transição $x_{new} \xrightarrow{\sigma} x_{new}$ (passo 13); (ii) caso contrário, quando $x_i \neq y$ (passo 14), a transição $x_{new} \xrightarrow{\sigma} y$ será criada em G_{db_j} (passo 15); na segunda, para cada transição $y \xrightarrow{\sigma} x_i \in G_{db_j}$ em que $x_i \neq y$, será criada uma nova transição $y \xrightarrow{\sigma} x_{new} \in G_{db_j}$ (passo 18). No passo 19, é verificado se o estado x_i é o estado inicial do autômato G.

Algoritmo 2: Decomposição de Estados

Entradas:
$G = (X, \Sigma, f, \Gamma, x_0)$
MDSE: Mapa de Sensores Estendido
j: índice do sensor
Saídas:
$G_{db_j} = (X_{db_j}, \Sigma_{db_j}, f_{db_j}, \Gamma_{db_j}, x_{0,db_j})$
MDS_{db_j}

1: $G_{db_j} \leftarrow G$
2: $MDS_{db_j} \leftarrow$ MDSE
3: **para** cada estado $x_i \in X$ **faça**
4: **se** $|\eta_j(x_i)| > 1$ **então**
5: $X_{split} \leftarrow \emptyset$
6: **para** cada $h_j^k(x_i) \in \eta_j(x_i)$ **faça**
7: $x_{new} \leftarrow \langle\langle x_i, h_j^k(x_i)\rangle\rangle$
8: $X_{split} \leftarrow X_{split} \cup \{x_{new}\}$
9: $X_{db_j} \leftarrow X_{db_j} \cup \{x_{new}\}$
10: $MDS_{db_j} \leftarrow MDS_{db_j} \cup \{(x_{new}, h_j^k(x_i))\}$
11: **para** cada transição $x_i \xrightarrow{\sigma} y \in G_{db_j}$ **faça**
12: **se** $x_i = y$ **então**
13: $f_{db_j}(x_{new}, \sigma) \leftarrow x_{new}$
14: **se não**
15: $f_{db_j}(x_{new}, \sigma) \leftarrow y$
16: **para** cada transição $y \xrightarrow{\sigma} x_i \in G_{db_j}$ **faça**
17: **se** $x_i \neq y$ **então**
18: $f_{db_j}(y, \sigma) \leftarrow x_{new}$
19: **se** $x_i = x_0$ **então**
20: $X_{db_j} \leftarrow X_{db_j} \cup \{x_{i_{new}}\}$
21: $x_{0_{db_j}} \leftarrow x_{i_{new}}$
22: $MDS_{db_j} \leftarrow MDS_{db_j} \cup \{(x_{i_{new}}, \text{SL})\}$
23: **para** cada $q \in X_{split}$ **faça**
24: $f_{db_j}(x_{i_{new}}, \varepsilon) \leftarrow q$
25: **para** cada $v \in X_{split}$ **faça**
26: **para** cada $w \in X_{split}$ **faça**
27: **se** $v \neq w$ **então**
28: $f_{db_j}(v, \varepsilon) \leftarrow w$
29: $MDS_{db_j} \leftarrow MDS_{db_j} \setminus \{x_i, \eta_j(x_i)\}$
30: $X_{db_j} \leftarrow X_{db_j} \setminus \{x_i\}$
31: **retorna** $[G_{db_j}, MDS_{db_j}]$

Em caso afirmativo, será criado um novo estado inicial em G_{db_j}, denominado $x_{i_{new}}$, o qual é adicionado ao conjunto X_{db_j} (passo 20) e ao mapa MDS_{db_j} (passo 22) com rótulo SL (Sem Leitura), significando que nenhuma leitura do sensor foi obtida nesse estado inicial. Além disso, são criadas transições rotuladas por ε do estado $x_{i_{new}}$ para cado estado $\langle\langle x_i, h_j^k(x_i)\rangle\rangle \in X_{split}$ (passo 24).

A próxima etapa do algoritmo 2 é criar as transições rotuladas por ε ligando dois estados diferentes que foram criados devido à decomposição do passo 4 ao passo 18 e que foram armazenados no conjunto X_{split}. A última etapa do algoritmo 2 é excluir do MDS_{db_j} o mapeamento referente ao estado x_i (passo 29) e excluir do autômato G_{db_j} o estado x_i, juntamente com as suas transições no passo 30).

Exemplo 2. Considere novamente o sistema da figura 3, que foi utilizado no exemplo 1. Para ilustrar o algoritmo 2, iremos aplicar o módulo de decomposição de estados para o sensor-2, que representa a chave de vazão. Assim, a entradas do algoritmo 2 serão o autômato G_{V_1}, representado na figura 4, o *MDSE*, descrito na tabela 2, e o índice do sensor ($j = 2$). Nos passos 1 e 2 são criados o autômato $G_{db_2} = G_{V_1}$ e o mapa de sensores $MDS_{db_2} =$ MDSE, respectivamente. Iniciando o laço do passo 3 pelo estado inicial *VPA*, verifica-se no passo 4 que $|\eta_2(VPA)| > 1$, logo ele deve ser decomposto. Note que $h_2^1(VPA) = F$ e $h_2^2(VPA) = NF$. Assim, para a saída $h_1^1(VPA)$ será criado o estado $\langle\langle VPA,F\rangle\rangle$ (passo 7) que é adicionado aos conjuntos X_{split} (passo 8) e X_{db_2} (passo 9). No passo 10, o MDS_{db_2} recebe o par ordenado $(\langle\langle VPA,F\rangle\rangle, \{F\})$. Em seguida, nos passos de 11 a 18 as transições do estado *VPA* são copiadas para o estado $\langle\langle VPA,F\rangle\rangle$, conforme destaque em verde na figura 5. Para a outra saída $h_2^2(VPA) = NF$, são executados os passos de 7 a 18, durante os quais será criado o estado $\langle\langle VPA,NF\rangle\rangle$ e serão copiadas as transições do estado *VPA* para o estado $\langle\langle VPA,NF\rangle\rangle$, conforme está destacado em azul na figura 5.

No passo 19 verifica-se que o estado *VPA*, que foi decomposto, é o estado inicial do autômato G_{V_1}, e, então, deve-se criar um novo estado inicial G_{db_2}, VPA_{new}, nos passos 20 e 21, sendo adicionado ao MDS_{db_2} no passo 22, em que $h_2(VPA_{new})$ =SL. No passo 24 são criadas transições rotuladas por ε a partir do estado VPA_{new} para o estados $\langle\langle VPA,F\rangle\rangle$ e $\langle\langle VPA,NF\rangle\rangle$, como destacado em vermelho na figura 5. Do passo 25 ao passo 28 são criadas transições rotuladas por ε entre os estados $\langle\langle VPA,F\rangle\rangle$ e $\langle\langle VPA,NF\rangle\rangle$, destacadas em laranja na figura 5, uma vez que esses estados foram armazenados no conjunto X_{split}.

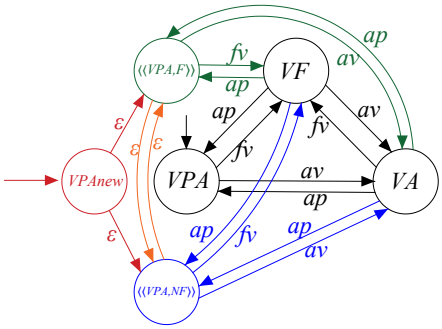


Figura 5. Autômato G_{db_2} ao final do passo 28.

Para finalizar a decomposição do estado $x_i = VPA$, exclui-se o par ordenado $(VPA, \{F, NF\})$ do mapa MDS_{db_j} no passo 29. Em seguida, no passo 30, remove-se, do autômato G_{db_2}, o estado *VPA* junto com a correspondentes transições, resultando no autômato da figura 6.

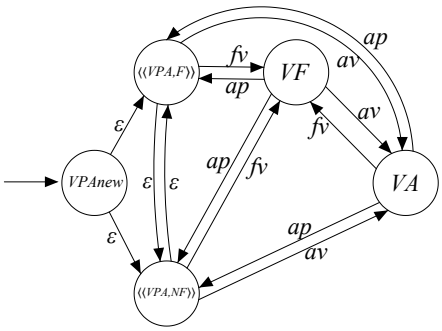


Figura 6. G_{db_2} após decomposição do estado *VAP*.

Para os estados VF e $VA \in G_{V_1}$, devem ser novamente realizados os passos 7–30 do algoritmo 2. Note que $\eta_2(VF) = \{NF\}$; logo $|\eta_2(VF)| = 1$, o que não satisfaz a condição do passo 4, e, portanto, nenhuma ação deve ser realizada em relação a VF. Para o estado VA, $|\eta_2(VA)| > 1$, e, portanto, ele deve ser decomposto nos estados $\langle\langle VA,F\rangle\rangle$ e $\langle\langle VA,NF\rangle\rangle$, conforme foi realizado para o estado VPA, resultando no autômato G_{db_2} da figura 7, cujo mapa MDS_{db_2} está descrito na tabela 3. Note que cada estado de G_{db_2} possui apenas uma saída do sensor-2. ■

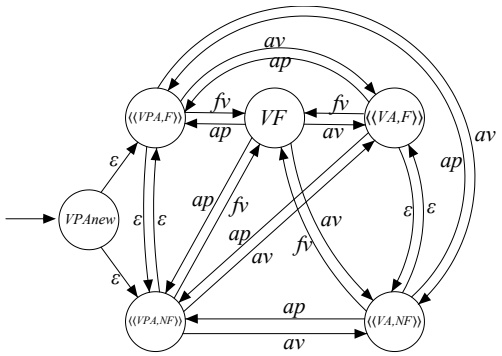


Figura 7. Autômato G_{db_2} ao final do algoritmo 2.

Tabela 3. MDS_{db_2} para o sensor de vazão.

Estados de G_{V_1}	$(x_i, \eta_2(x_i))$
VPA_{new}	$(VPA_{new}, \{SL\})$
$\langle\langle VPA,F\rangle\rangle$	$(\langle\langle VPA,F\rangle\rangle, \{F\})$
$\langle\langle VPA,NF\rangle\rangle$	$(\langle\langle VPA,NF\rangle\rangle, \{NF\})$
VF	$(VF, \{NF\})$
$\langle\langle VA,F\rangle\rangle$	$(\langle\langle VA,F\rangle\rangle, \{F\})$
$\langle\langle VA,NF\rangle\rangle$	$(\langle\langle VA,NF\rangle\rangle, \{NF\})$

Com base no algoritmo 2, o seguinte resultado pode ser enunciado.

Proposição 1. $L(G) = L(Obs(G_{db_j}, \Sigma_{db_j} \backslash \{\varepsilon\}))$.

Prova. Por construção, quando o mapeamento de um sensor j para um estado x_i do autômato G possuir mais de uma saída, ou seja, $\eta_j(x_i) = \{h_j^1(x_i), h_j^2(x_i), ..., h_j^{p_{ij}}(x_i)\}$, o estado x_i será decomposto em $\langle\langle x_i, h_j^1(x_i)\rangle\rangle, \langle\langle x_i, h_j^2(x_i)\rangle\rangle, ..., \langle\langle x_i, h^{p_{ij}}(x_i)\rangle\rangle$, com a palavra nula ε rotulando todas as transições entre esses estados. Como todas as transições do estado x_i são copiadas para os respectivos estados decompostos, então, ao calcular $Obs(G_{db_j}, \Sigma_{db_j} \backslash \{\varepsilon\})$, os estados decompostos serão reagrupados por pertencerem ao alcance não observável em relação ao conjunto $\Sigma_{db_j} \backslash \{\varepsilon\}$. ■

3.2 Criação de Estados

O módulo de criação de estados do MS-BEL é responsável por criar o autômato G_{s_j}, $j = 1 \ldots M$, a partir do autômato G_{db_j} e do mapa MDS_{db_j} obtidos no algoritmo 2. Para tanto, o algoritmo 3 (criação de estados) insere as informações que estão mapeadas em MDS_{db_j} no autômato G_{db_j} através da criação de estados e transições com os eventos de leituras dos sensores. Deve ser ressaltado que o algoritmo de criação de estados leva também em conta o comportamento de falha do autômato G_{db_j} a fim de não alterá-lo durante a construção do autômato G_{s_j}.

As entradas do algoritmo 3 são o autômato G_{db_j}, o mapa MDS_{db_j} e o índice j do sensor, e a saída é o

Algoritmo 3: Criação de Estados

Entradas:
$G_{db_j} = (X_{db_j}, \Sigma_{db_j}, f_{db_j}, \Gamma_{db_j}, x_{db_j})$
MDS_{db_j}
j: índice do sensor
Saída:
$G_{s_j} = (X_{s_j}, \Sigma_{s_j}, f_{s_j}, \Gamma_{s_j}, x_{0,s_j})$
1: $G_{s_j} \leftarrow G_{db_j}$
2: **para** cada transição $x \xrightarrow{\sigma} y \in G_{db_j}$ **faça**
3: **se** $\eta_j(x) \neq \eta_j(y)$ **então**
4: $x_{new} \leftarrow \langle\langle x, \sigma, y\rangle\rangle$
5: $X_{s_j} \leftarrow X_{s_j} \cup \{x_{new}\}$
6: $f_{s_j}(x, \sigma) \leftarrow x_{new}$
7: $f_{s_j}(x_{new}, \eta_j(y)) \leftarrow y$
8: **para** cada transição $y \xrightarrow{\alpha} z \in G_{db_j}$ **faça**
9: **se** $(\alpha \in \Sigma_{f,db_j})\&(\alpha \neq \sigma)$ **então**
10: **se** $\eta_j(x) \neq \eta_j(z)$ **então**
11: $x'_{new} \leftarrow \langle\langle x_{new}, \alpha, z\rangle\rangle$
12: $X_{s_j} \leftarrow X_{s_j} \cup \{x'_{new}\}$
13: $f_{s_j}(x_{new}, \alpha) \leftarrow x'_{new}$
14: $f_{s_j}(x'_{new}, \eta_j(z)) \leftarrow z$
15: **se não**
16: $f_{s_j}(x_{new}, \alpha) \leftarrow z$
17: $G_{s_j} \leftarrow Obs(G_{s_j}, \Sigma_{s_j} \backslash \varepsilon)$
18: **retorna** G_{s_j}

autômato G_{s_j}. Inicialmente o autômato G_{s_j} é atribuído a G_{db_j} no passo 1. Em seguida, cada transição $x \xrightarrow{\sigma} y$ de G_{db_j} é percorrida, e, se $\eta_j(x) \neq \eta_j(y)$, então um estado $x_{new} = \langle\langle x, \sigma, y\rangle\rangle$ (passo 4) é adicionado a G_{s_j} (passo 5) e são criadas duas novas transições: (i) $x \xrightarrow{\sigma} x_{new}$ (passo 6); (ii) $x_{new} \xrightarrow{\eta_j(y)} y$ (passo 7). Dos passo 8–16 são construídas as transições com eventos de falhas a partir do novo estado x_{new} que foi adicionado ao autômato G_{s_j}. Assim, no mesmo laço da transição $x \xrightarrow{\sigma} y \in G_{db_j}$, é verificado na transição $y \xrightarrow{\alpha} z \in G_{db_j}$ se $\alpha \in \Sigma_{f,db_j}$ e se $\alpha \neq \sigma$ (passo 9). Dois casos devem ser considerados. No primeiro caso, se $\eta_j(x) \neq \eta_j(z)$ (passo 10), então cria-se um novo estado $x'_{new} = \langle\langle x_{new}, \alpha, z\rangle\rangle$ (passo 11) que é adicionado a G_{s_j} (passo 12) e na sequência são criadas duas novas transições: (i) $x_{new} \xrightarrow{\alpha} x'_{new}$ (passo 13); (ii) $x'_{new} \xrightarrow{h_j(z)} z$ (passo 14). No segundo caso, se $\eta_j(x) = \eta_j(z)$ (passo 15), então a transição a ser inserida em G_{s_j} será $x_{new} \xrightarrow{\alpha} z$ (passo 16). Após percorrer todas as transições do autômato G_{db_j} e inserir estados e transições em G_{s_j} de acordo com o MDS_{db_j}, no passo 17 é construído um autômato determinístico $Obs(G_{s_j}, \Sigma_{db_j} \backslash \{\varepsilon\})$. Finalmente, o comportamento completo do sistema é obtido através da composição paralela entre os módulos G_{s_j}, $j = 1, \ldots, M$, obtendo-se $G_{MS\text{-}BEL} = \|_{j=1}^{M} G_{s_j}$.

Proposição 2. $L(G) = L(Obs(G_{s_j}, \Sigma_{s_j} \backslash Y_j))$.

Prova 1. Por construção, os estados resultantes do módulo de decomposição de estados executado pelo MS- BEL, são conectados entre si por transições rotuladas com a palavra nula $\{\varepsilon\}$. Logo $Obs(G_{s_j}, \Sigma_{s_j} \backslash Y_j)$ irá reagrupar esses estados por conta do alcance não observável em estados com as mesmas transições de origem e destino dos estados que originaram o decomposição. Além disso,

como os estados x_{new} e x'_{new}, que são inseridos pelo MS-BEL durante a execução do módulo de criação de estados possuem como evento ativo apenas um elemento do conjunto Y_j e/ou eventos α, tal que $\alpha \in \Sigma_f$, então $Obs(G_{s_j}, \Sigma_{s_j} \backslash Y_j)$ irá reagrupar esses estados no estado de destino da sua respectiva transição. ■

De acordo com a proposição 2, pode-se concluir que: (i) as modificações realizadas pelo MS-BEL não alteram o comportamento modelado pelo autômato G; (ii) ao final da construção do G_{s_j}, são inseridos em G os eventos dos sensores mapeados considerando-se os comportamentos normal e de falha de G.

4. APLICAÇÃO DO MS-BEL A UM SISTEMA DE AQUECIMETO

Considere um sistema de aquecimento composto por uma bomba, uma válvula, um sensor de pressão e um sensor de vazão, conforme mostrado na figura 8. A bomba possui dois estados de funcionamento, bomba desligada ($BOFF$) e bomba ligada (BON), e a válvula possui três estados, válvula fechada (VF), válvula aberta (VA) e válvula travada fechada (VTF). O controlador é responsável por coordenar a sequência de comandos para a bomba e para a válvula. O sensor de pressão fornece sinal de saída discreta com a informação sobre o estado de pressurização da rede. Quando a bomba, está ligada a saída do sensor de pressão é P, caso contrário é SP; logo $Y_1 = \{P,SP\}$. O sensor de vazão indica se há ou não fluxo na rede. Quando a bomba está ligada e a válvula está aberta, o sensor de vazão tem como saída o sinal V, caso contrário sua saída é SV; assim $Y_2 = \{V,SV\}$.

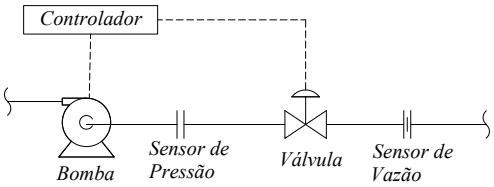
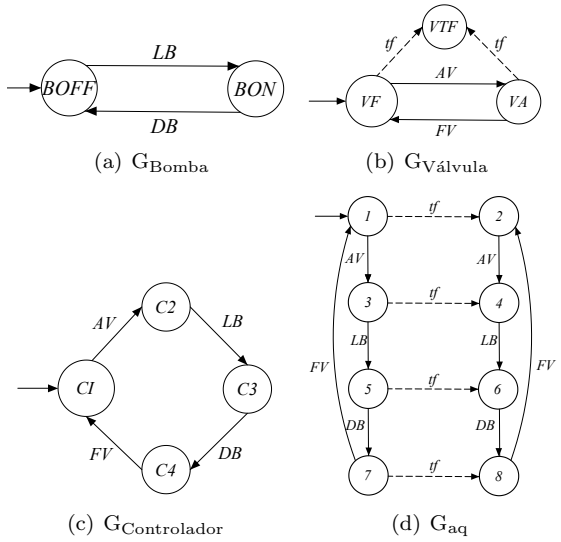


Figura 8. Arranjo dos componentes do sistema.



(a) G_{Bomba} (b) $G_{\text{Válvula}}$

(c) $G_{\text{Controlador}}$ (d) G_{aq}

Figura 9. Autômatos que modelam os componentes ativos.

Tabela 4. Descrição dos estados e eventos dos componentes do sistema de aquecimento.

Estado	Descrição	Evento	Descrição
$BOFF$	Bomba desligada	LB	Ligar a bomba
BON	Bomba ligada	DB	Desligar a bomba
VF	Válvula fechada	AV	Abrir a válvula
VA	Válvula aberta	FV	Fechar a válvula
VTF	Válvula travada fechada	tf	Travar fechada

Na figura 9 (a)–(c) estão modelados os autômatos individuais da bomba, da válvula e do controlador, respectivamente, e, na tabela 4, estão descritos os estados e eventos da bomba e da válvula. O comportamento integrado dos componentes do sistema é modelado pelo autômato G_{aq}=$G_{\text{Bomba}} \| G_{\text{Válvula}} \| G_{\text{Controlador}}$, representado na figura 9 (d), em que $G_{aq} = (X_{aq}, \Sigma_{aq}, f_{aq}, \Gamma_{aq}, x_{0,aq})$. Note que o autômato G_{aq} não possui informações sobre o comportamento dos sensores de pressão e de vazão.

A fim de se aplicar o MS-BEL, deve-se construir o MDSE que relaciona os estados do autômato G_{aq} com as saídas dos sensores do sistema de acordo com a tabela 5; por exemplo, para o estado 1, $\eta_1(\{BOFF,C1,VF\}) = \{SP\}$ e $\eta_2(\{BOFF,C1,VF\}) = \{SV\}$.

Tabela 5. MDSE para o autômato G_{aq}.

Número do Estado	Nome do Estado	$(x_i, \eta_1(x_i))$	$(x_i, \eta_2(x_i))$
1	(BOFF,C1,VF)	(1,{SP})	(1,{SV})
2	(BOFF,C1,VTF)	(2,{SP})	(2,{SV})
3	(BOFF,C2,VA)	(3,{SP})	(3,{SV})
4	(BOFF,C2,VTF)	(4,{SP})	(4,{SV})
5	(BON,C3,VA)	(5,{P})	(5,{V})
6	(BON,C3,VTF)	(6,{P})	(6,{SV})
7	(BOFF,C4,VA)	(7,{SP})	(7,{SV})
8	(BOFF,C4,VTF)	(8,{SP})	(8,{SV})

Aplicando o MS-BEL ao sistema de aquecimento descrito, é possível notar no MDSE, tabela 5, que há somente um elemento de saída possível do conjunto Y_j para cada estado de G_{aq}, ou seja, $|\eta_j(x_i)| = 1, \forall i, j$. Como consequência, ao aplicar o MS-BEL, o algoritmo 2, do módulo de decomposição de estados, não realizará nenhuma alteração por conta da condição do passo 4. Logo, G_{db_j} e MDS_{db_j} serão iguais a G_{aq} e a $MDSE$, respectivamente, $j = 1, 2$.

Aplicando o algoritmo 3 de criação de estados para $j = 1$ (sensor de pressão), obtém-se o autômato G_{s_1} representado na figura 10. Note que o módulo de criação de estados inseriu seis novos estados e doze transições em virtude das mudanças de leituras do sensor de pressão identificadas no mapa da tabela 5 e para manter o comportamento do sistema em função da ocorrência da falha.

As seguintes alterações foram realizadas para obter G_{s_1}:

- Criação do estado $\langle 3, LB, 5\rangle$ e das transições $3 \xrightarrow{LB} \langle 3, LB, 5\rangle$ e $\langle 3, LB, 5\rangle \xrightarrow{P} 5$, uma vez que $\eta_1(3) = SP \neq \eta_1(5) = P$ (condição do passo 3). Como há uma transição de falha $5 \xrightarrow{tf} 6$ e $\eta_1(3) \neq \eta_1(6)$ (condições dos passos 9 e 10, respectivamente), deve-se adicionar o estado $\langle\langle 3, LB, 5\rangle, tf, 6\rangle$ e as transições $\langle 3, LB, 5\rangle \xrightarrow{tf} \langle\langle 3, LB, 5\rangle, tf, 6\rangle$ e $\langle\langle 3, LB, 5\rangle, tf, 6\rangle \xrightarrow{P} 6$. Analogamente, para os estados $\langle 5, DB, 7\rangle$ e $\langle\langle 5, DB, 7\rangle, tf, 8\rangle$ e para as

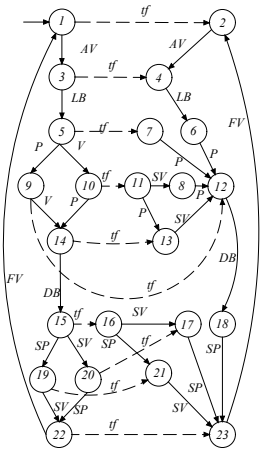
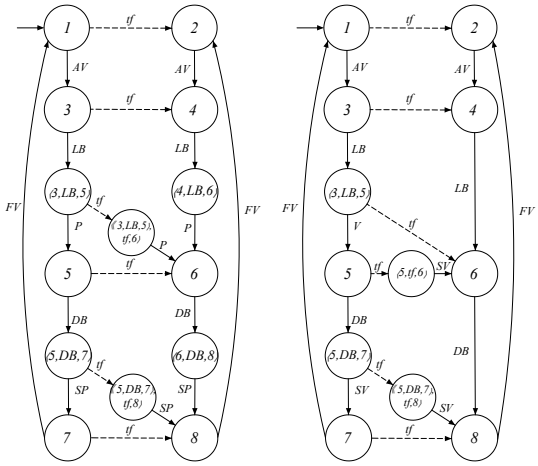


Figura 11. $G_{MS\text{-}BEL} = G_{aq}||G_{s_1}||G_{s_2}$.



(a) G_{s_1} (b) G_{s_2}

Figura 10. Autômatos que modelam o comportamento dos sensores de pressão (a) e vazão (b).

transições transições $5 \overset{DB}{\rightarrow} \langle 5, DB, 7\rangle$, $\langle 5, DB, 7\rangle \overset{SP}{\rightarrow} 7$, $\langle 5, DB, 7\rangle \overset{tf}{\rightarrow} \langle\langle 5, DB, 7\rangle, tf, 8\rangle$ e $\langle\langle 5, DB, 7\rangle, tf, 8\rangle \overset{SP}{\rightarrow} 8$;

- Criação do estado $\langle 4, LB, 6\rangle$ e das transições $4 \overset{LB}{\rightarrow} \langle 4, LB, 6\rangle$ e $\langle 4, LB, 6\rangle \overset{P}{\rightarrow} 6$, pois $\eta_1(4) = SP \neq \eta_1(6) = P$ (condição do passo 3). Analogamente, para os estados $\langle 6, DB, 8\rangle$ e transições $6 \overset{DB}{\rightarrow} \langle 6, DB, 8\rangle$ e $\langle 6, DB, 8\rangle \overset{SP}{\rightarrow} 8$.

Aplicando-se o módulo de criação de estados para $j = 2$, obtém-se o autômato G_{s_2}, que modela o comportamento do sensor de vazão representado na figura 10(b). Note que, assim como para G_{s_1}, estados e transições foram inseridas quando houve alteração nas leituras dos estados de origem e destino de acordo com a tabela 5. O autômato $G_{MS\text{-}BEL} = G_{aq}||G_{s_1}||G_{s_2}$ está representado na figura 11. Note que os estados foram renomeados para facilitar a visualização.

Uma importante consequência da aplicação do MS-BEL é possibilidade de se analisar a contribuição de cada sensor para a diagnosticabilidade de determinada falha. Por exemplo, no autômato G_{s_1} é possível observar que existe uma sequência normal, $s_n = (AV, LB, P, DB, SP, FV)$, e uma sequência de falha, $s_f = (tf, AV, LB, P, DB, SP, FV)$, que possuem a mesma projeção sobre $\Sigma^*_{o,s1}$, ou seja, $P_o(s_n) = P_o(s_f) = (AV, LB, P, DB, SP, FV)$, evidenciando que a introdução do sensor de pressão no modelo não contribuiu para a diagnosticabilidade da falha tf. Já, para o autômato G_{s_2}, nota-se que a sequência normal, $s_n = (AV, LB, V, DB, SV, FV)$, e a sequência após o evento de falha $s_f = (tf, AV, LB, DB, FV)$, possuem projeções diferentes, $P_o(s_n) = (AV, LB, V, DB, SV, FV)$ e $P_o(s_f) = (AV, LB, DB, FV)$, o que evidencia que a introdução do sensor de vazão no modelo do sistema pode levar à diagnosticabilidade da falha tf (Basilio et al., 2012). De fato, a análise completa da diagnosticabilidade do $G_{MS\text{-}BEL}$ (Cruz, 2020) utilizando Viana and Basilio (2019) revela que $L(G_{MS\text{-}BEL})$ é diagnosticável em relação a P_o e a falha tf.

5. CONCLUSÃO

Neste artigo, o mapa de sensores foi revisitado, e um novo método para a modelagem de SED foi proposta. A eficácia do novo método foi ilustrado utilizando o mesmo exemplo anteriormente utilizado em Sampath et al. (1996) para ilustrar o mapa de sensores.

REFERÊNCIAS

Basilio, J.C., Lima, S.T.S., Lafortune, S., and Moreira, M.V. (2012). Computation of minimal event bases that ensure diagnosability. *Discrete Event Dynamic Systems*, 22(3), 249–292.

Cassandras, C.G. and Lafortune, S. (2008). *Introduction to Discrete Event Systems*. Springer, New York, 2 edition.

Cruz, V.H.A. (2020). *Mapeamento de sensores para a diagnose de falhas de sistemas a eventos discretos: uma abordagem inspirada na especificação de linguagens*. Dissertação de mestrado, Universidade Federal do Rio de Janeiro - COPPE, Rio de Janeiro, RJ, Brasil.

Debouk, R., Lafortune, S., and Teneketzis, D. (2002). On an optimization problem in sensor selection. *Discrete Event Dynamic Systems*, 12(4), 417–445.

Mealy, G.H. (1955). A method for synthesizing sequential circuits. *The Bell System Technical Journal*, 34(5), 1045–1079.

Sampath, M. (2001). A hybrid approach to failure diagnosis of industrial systems. In *Proceedings of the 2001 American Control Conference*, 2077–2082. Arlington, Virginia, USA.

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.

Sampath, M., Sengupta, R., Lafortune, S., Sinnamohideen, K., and Teneketzis, D.C. (1996). Failure diagnosis using discrete-event models. *IEEE Transactions on Control Systems Technology*, 4(2), 105–124.

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.

Viana, G.S. and Basilio, J.C. (2019). Codiagnosability of discrete event systems revisited: A new necessary and sufficient condition and its applications. *Automatica*, 101, 354–364.

Zaytoon, J. and Lafortune, S. (2013). Overview of fault diagnosis methods for discrete event systems. *Annual Reviews in Control*, 37(2), 308–320.