

Controle Supervisório de Sistemas a Eventos Discretos com Múltiplos Canais de Comunicação *

Marcos Vinícius Silva Alves* Lilian Kawakami Carvalho**
João Carlos dos Santos Basilio**

* Departamento de Engenharia Elétrica, Universidade Federal de Sergipe, SE, (e-mail: marcosvsalves@academico.ufs.br).

** Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro, RJ (e-mails: lilian@dee.ufrj.br, basilio@dee.ufrj.br).

Abstract: In this paper, we address the problem of supervisory control of Networked Discrete Event Systems, where the transmission of event occurrences from the plant to the supervisor and the transmission of control decisions from the supervisor to the plant are carried out through a network with several communication channels subject to bounded delays, with different maximum delay upper bounds for each channel. Due to the multichannel structure, there may occur changes in the order of observations and there may also occur situations where the same control decision reaches different locations in the plant with different delays. To deal with such effects, we propose a new automaton model and, based on this model, we formulate a networked supervisory control problem, present a necessary and sufficient condition for the existence of a solution, and a test to verify the proposed existence condition.

Resumo: Neste artigo, abordamos o problema de controle supervisório de Sistemas de Eventos Discretos em Rede no qual a transmissão de ocorrências de eventos da planta para o supervisor e de ações de controle do supervisor para a planta são realizadas por meio de redes com vários canais de comunicação. Supõe-se que os canais de comunicação são sujeitos a atrasos limitados, sendo os atrasos máximos determinados individualmente para cada canal. Por se tratar de uma estrutura com múltiplos canais, podem ocorrer alterações na ordem das observações e também podem haver situações em que uma mesma ação de controle do supervisor chegue a diferentes locais da planta com diferentes atrasos. Para lidar com tais efeitos, propomos uma nova modelagem baseada em autômatos e, com base nesse modelo, formulamos um problema de controle supervisório em rede, apresentamos uma condição necessária e suficiente para a existência de uma solução e um teste para verificar a condição de existência apresentada.

Keywords: Discrete event systems; supervisory control; networked systems; communication delays; automata.

Palavras-chaves: Sistemas a eventos discretos; controle supervisório; sistemas em rede; atrasos de comunicação; autômatos.

1. INTRODUÇÃO

A inclusão de redes de comunicação em sistemas de controle supervisório torna necessário considerar a existência de atrasos entre a ocorrência de um evento na planta e a sua observação pelo supervisor e, também, atrasos na efetiva aplicação das ações de controle emitidas pelo supervisor. Consequentemente, as abordagens iniciais em controle supervisório (Ramadge e Wonham, 1987, 1989) tem aplicação limitada em sistemas em rede. Com o objetivo de lidar com atrasos de comunicação, uma nova abordagem teórica, chamada controle supervisório de sistemas a eventos discretos em rede (SEDR) (Lin, 2014; Alves et al., 2017, 2021), tem emergido na área de Sistemas a Eventos Discretos (SED). Lin (2014) considera redes com um único canal de comunicação. Por sua vez, redes de

comunicação com múltiplos canais foram abordadas em (Alves et al., 2017, 2021).

No contexto de SEDR, redes de comunicação com múltiplos canais foram consideradas em problemas de diagnose de falhas e estimação de estados por Nunes et al. (2018) e Alves e Basilio (2019), respectivamente, e também em controle supervisório por Alves et al. (2017, 2021).

Em uma estrutura diferente, Rashidinejad et al. (2018) abordam o problema de controle supervisório de SEDs temporizados, supondo que as ocorrências de eventos na planta são transmitidas para o supervisor por meio de um único canal de comunicação sujeito a atrasos fixos. Outros trabalhos anteriores em controle supervisório de SEDR (Park e Cho, 2007; Lin, 2014; Shu e Lin, 2017; Zhu et al., 2019) consideraram um único canal de comunicação *first-in first-out* (FIFO) sujeito a atrasos limitados. Em todos esses casos, o supervisor observa as ocorrências de eventos com atraso, mas não há mudanças na ordem de

* Este trabalho foi financiado em parte pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e pela Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ).

observação dos eventos, ao contrário do que ocorre nas redes de comunicação com múltiplos canais estudadas em (Alves et al., 2017, 2021).

Neste artigo, expande-se a metodologia desenvolvida em (Nunes et al., 2018; Alves et al., 2017, 2021), na qual se considerava atrasos apenas nas observações de eventos ocorridos na planta, para abranger arquiteturas de controle supervísório, nas quais as ações de controle do supervisor também são transmitidas para a planta por meio de redes com vários canais sujeitos a diferentes atrasos de comunicação. Para tanto, propomos um método para a construção de um novo autômato que modela os efeitos dos atrasos de comunicação na transmissão tanto de ocorrências de eventos na planta quanto de ações de controle do supervisor. Com base nesse modelo, formulamos um problema de controle supervísório em rede, apresentamos uma condição necessária e suficiente para a existência de um supervisor em rede, e uma forma sistemática de verificar a condição de existência apresentada. Em decorrência da limitação de espaço, as demonstrações dos resultados apresentados foram omitidas neste artigo.

2. FUNDAMENTOS TEÓRICOS

Um autômato de estados finitos determinístico é denotado pela quádrupla $G = (X, \Sigma, f, x_0)$, em que X é o conjunto finito de estados, Σ é o conjunto finito de eventos, $f : X \times \Sigma \rightarrow X$ é a função (parcial) de transição, e x_0 é o estado inicial. A função de transição f pode ser estendida, da forma usual, para $f : X \times \Sigma^* \rightarrow X$, em que Σ^* denota o fecho de Kleene Σ . A linguagem gerada por G é definida como $L(G) = \{s \in \Sigma^* : (\exists x \in X)[f(x_0, s) = x]\}$. O fecho do prefixo de uma linguagem L é definido como $\bar{L} = \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\}$. L é dita ser prefixo-fechada se $L = \bar{L}$. Dado um estado $x \in X$, o conjunto de eventos ativos no estado x é definido como $\Gamma_G(x) = \{\sigma \in \Sigma : f(x, \sigma) \text{ é definido}\}$.

O conjunto de eventos é particionado como $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc} = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_c (resp. Σ_o) e Σ_{uc} (resp. Σ_{uo}) são, respectivamente, os conjuntos de eventos controláveis (resp. observáveis) e não-controláveis (resp. não-observáveis). A projeção $P_o : \Sigma^* \rightarrow \Sigma_o^*$ é definida como (Ramadge e Wonham, 1989): (i) $P_o(\varepsilon) = \varepsilon$, em que ε denota a sequência vazia, (ii) $P_o(\sigma) = \sigma$, se $\sigma \in \Sigma_o$, ou $P_o(\sigma) = \varepsilon$, se $\sigma \in \Sigma_{uo}$, e (iii) $P_o(s\sigma) = P_o(s)P_o(\sigma)$, para $s \in \Sigma^*$ e $\sigma \in \Sigma$. A projeção inversa é definida como $P_o^{-1}(t) = \{s \in \Sigma^* : P_o(s) = t\}$. Ambas, projeção e projeção inversa, são estendidas para linguagem da forma usual.

No decorrer do texto, $Ac(G)$ denota a parte acessível de G . Para dois autômatos G_1 e G_2 , sua composição paralela (Cassandras e Lafontaine, 2008, sec. 2.3) é denotada por $G_1 \parallel G_2$. Adicionalmente, dado um vetor Q de dimensões $1 \times p$, com $p \in \mathbb{N}$, o y -ésimo componente do vetor Q é denotado por $[Q]_y$. Para um conjunto B (resp. sequência s), $|B|$ (resp. $\|s\|$) denota a sua cardinalidade (resp. comprimento).

Neste artigo, são considerados supervisores que atuam restringindo a ocorrência de eventos na planta G . Para isso, após cada nova observação de evento em Σ_o , o supervisor envia para a planta uma ação de controle $\gamma \in 2^\Sigma$, formada com os eventos que são permitidos de ocorrer até que

uma nova observação aconteça. O supervisor não é capaz de impedir a ocorrência de eventos não-controláveis e, portanto, as ações de controle pertencem ao conjunto $V := \{\gamma \in 2^\Sigma : \Sigma_{uc} \subseteq \gamma\}$. Dessa forma, um supervisor S pode ser definido como uma função $S : P_o(L(G)) \rightarrow V$. A linguagem gerada pelo sistema em malha fechada S/G (lê-se S controlando G) é definida recursivamente da forma a seguir: (i) $\varepsilon \in L(S/G)$, e (ii) $\forall s \in \Sigma^*$ e $\forall \sigma \in \Sigma$, $s\sigma \in L(S/G) \Leftrightarrow s \in L(S/G) \wedge s\sigma \in L(G) \wedge \sigma \in S(P_o(s))$.

Por fim, dada uma linguagem $K \subseteq L(G)$, K é dita ser controlável e.r.a $L(G)$ e Σ_{uc} se $\bar{K}\Sigma_{uc} \cap L(G) \subseteq \bar{K}$, e normal e.r.a $L(G)$ e P_o se $\bar{K} = L(G) \cap P_o^{-1}(P_o(\bar{K}))$.

3. ESTRUTURA DE CONTROLE SUPERVISÓRIO EM REDE

Como mostrado na Figura 1, um Sistema a Eventos Discretos com Múltiplos Canais de Observação (SED-MCO) é composto por uma planta, modelada por um autômato $G = (X, \Sigma, f, x_0)$, um conjunto de locais de medição MS_j , $j \in I_m$ com $I_m = \{1, \dots, m\}$, e uma rede de comunicação com múltiplos canais, denominados canais de observação, que são utilizados para transmitir os eventos detectados nos locais de medição para um agente observador, o qual pode ser, por exemplo, um supervisor ou um diagnosticador.

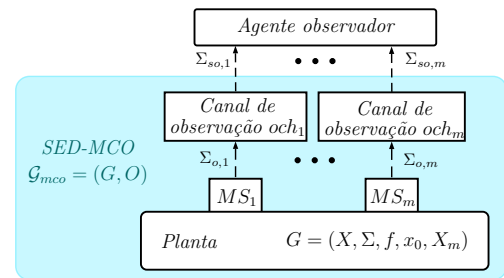


Figura 1. SED com múltiplos canais de observação.

O local de medição MS_j , $j \in I_m$, detecta as ocorrências dos eventos pertencentes ao subconjunto $\Sigma_{o,j} \subseteq \Sigma_o$. Em decorrência da natureza distribuída do sistema, os conjuntos $\Sigma_{o,j}$, $j \in I_m$ são disjuntos, isto é, $\Sigma_o = \dot{\cup}_{j \in I_m} \Sigma_{o,j}$.

Cada local de medição MS_j , $j \in I_m$, está conectado ao supervisor por meio de um canal de observação distinto, denotado por och_j . Os canais de observação och_j , $j \in I_m$, são modelado por filas *first-in first-out* (FIFO) e supõe-se que eles estão sujeitos a atrasos de comunicação, chamados de atrasos de observação. A modelagem de atrasos em redes comutadas por pacotes é geralmente realizada assumindo que a transmissão de cada nó de entrada é FIFO (Kurose e Ross, 2017, p. 319). Profibus (Tovar e Vasques, 1999), Controller Area Network (CAN) (Davis et al., 2011) e Ethernet (Jasperneite e Neumann, 2001) são exemplos de redes nas quais o enfileiramento FIFO é aplicado. Atrasos de observação são medidos por *passos* (Tripakis, 2004), isto é, pelo número de eventos que ocorrem na planta enquanto a ocorrência de evento está sendo transmitida. O atraso de observação máximo é determinado pelo canal de observação, ou seja, as ocorrências de eventos transmitidas pelo canal de observação och_j , $j \in I_m$, estão sujeitas a atrasos de no máximo $k_{o,j} \in \mathbb{Z}_+$ passos.

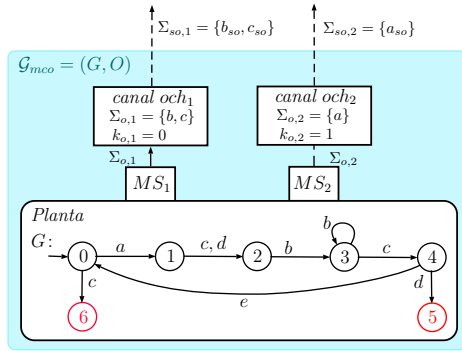


Figura 2. SED-MCO do Exemplo 1.

Um SED-MCO é formalmente definido a seguir.

Definição 1. (SED-MCO). Um sistema a eventos discretos com múltiplos canais de observação (SED-MCO) é um par $\mathcal{G}_{mco} = (G, O)$, em que $G = (X, \Sigma, f, x_0)$ é um autômato e O é o vetor de observação, o qual é um vetor $1 \times m$ cujo j -ésimo componente é $[O]_j = (\Sigma_{o,j}, k_{o,j})$. \square

Com o objetivo de representar o efeito dos atrasos de observação em um SED-MCO, Nunes et al. (2018) propuseram uma abordagem em que a ocorrência de um evento na planta e a sua efetiva observação pelo agente observador são representadas por eventos distintos. Nessa abordagem, define-se o conjunto de eventos $\Sigma_{so} = \{\sigma_{so} : \sigma \in \Sigma_o\}$, em que cada evento σ_{so} é obtido a partir de um evento observável $\sigma \in \Sigma_o$, adicionando-se o subscrito so . O evento σ_{so} representa a observação bem sucedida de σ . Dessa forma, quando um evento $\sigma \in \Sigma_o$ ocorrer na planta, o atraso na observação de σ (em passos) será igual ao número de ocorrências de eventos entre σ e σ_{so} . Como mostrado na Figura 1, o conjunto Σ_{so} é particionado como $\Sigma_{so} = \bigcup_{j \in I_m} \Sigma_{so,j}$, em que $\Sigma_{so,j} = \{\sigma_{so} : \sigma \in \Sigma_{o,j}\}$.

Nunes et al. (2018) e Alves e Basilio (2019) propuseram algoritmos com os quais pode-se obter um autômato $G_{mco} = (X_{mco}, \Sigma_{mco}, f_{mco}, x_{0mco})$, em que $\Sigma_{mco} = (\Sigma \cup \Sigma_{so})$, que descreve o comportamento e a observação de um SED-MCO. Sejam as projeções $P_{mco} : \Sigma_{mco}^* \rightarrow \Sigma^*$ e $P_{mco,so} : \Sigma_{mco}^* \rightarrow \Sigma_{so}^*$, uma sequência $w \in L(G_{mco})$ descreve o caso em que a sequência $s = P_{mco}(w) \in L(G)$ ocorreu na planta e a sequência $s_{so} = P_{mco,so}(w) \in \Sigma_{so}^*$ foi observada pelo agente observador.

Exemplo 1. Considere o SED com múltiplos canais de observação $\mathcal{G}_{mco} = (G, O)$ mostrado na Figura 2, em que $\Sigma = \{a, b, c, d, e\}$, $\Sigma_o = \{a, b, c\}$, $\Sigma_c = \{c, d\}$ e $\Sigma_{so} = \{a_{so}, b_{so}, c_{so}\}$. As ocorrências de eventos são transmitidas por dois canais de observação, descritos pelo vetor de observação $O = ((\{b, c\}, 0), (\{a\}, 1))$. De acordo com O , as ocorrências dos eventos b e c (resp. do evento a) são detectadas pelo local de medição MS_1 (resp. MS_2), transmitidas pelo canal de observação och_1 (resp. och_2) e alcançam o agente observador em até 0 (resp. 1) passo.

O autômato G_{mco} que modela o SED-MCO \mathcal{G}_{mco} é apresentado na Figura 3. Cada estado desse autômato é um par, em que a primeira componente é o estado correspondente da planta G e a segunda componente é determinada pelo *status* dos canais de observação. Por simplicidade, as segundas componentes dos estados de G_{mco} foram renomeadas, porém leitores interessados podem encontrar como defini-las da forma original em (Alves e Basilio, 2019).

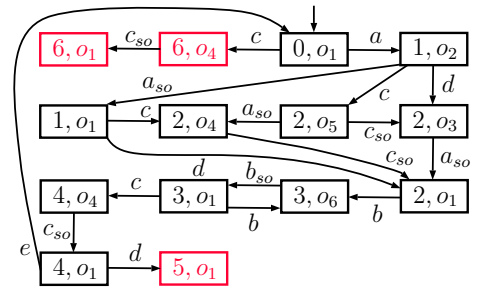


Figura 3. Autômato G_{mco} que modela o SED-MCO do Exemplo 1.

Por meio da Figura 3, pode-se notar que se os eventos a e c ocorrerem na planta nessa ordem, o evento a pode ser observado de uma das formas a seguir: (i) antes da ocorrência de c , (ii) entre a ocorrência e a observação c_{so} de c , ou (iii) após a observação c_{so} de c . Nesse último caso a e c são observados em uma ordem diferente daquela em que eles ocorreram na planta. \square

Como ilustrado no Exemplo 1, atrasos de observação podem fazer com que ocorrências de eventos transmitidas por canais diferentes sejam observadas em uma ordem diferente daquela em que elas ocorreram na planta, o que é capturado pelo modelo G_{mco} . Em alguns pontos deste artigo, um SED-MCO \mathcal{G}_{mco} será, por simplicidade, referido pelo autômato G_{mco} que o modela.

3.1 SED com Múltiplos Canais de Observação e Controle

Nesse artigo, considera-se a arquitetura de controle mostrada na Figura 4, na qual o agente observador foi substituído por um supervisor em rede, denotado por S_{net} , que atua restringindo o comportamento de um SED-MCO, sendo a ação de controle do supervisor transmitida para um conjunto de locais de atuação AS_i , $i \in I_n$ com $I_n = \{1, \dots, n\}$, situados na planta.

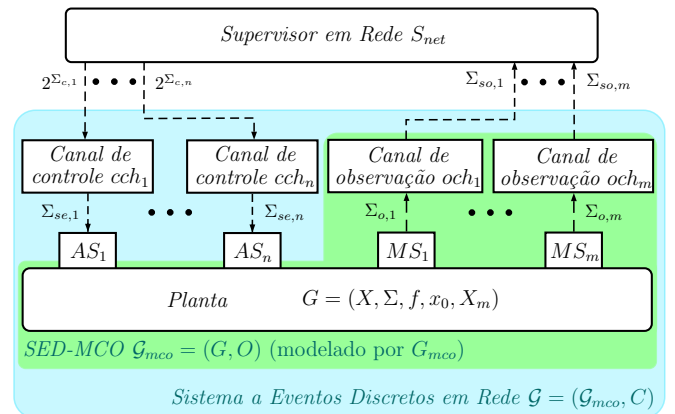


Figura 4. Estrutura de controle supervisorório de um SEDR.

Os locais de atuação AS_i , $i \in I_n$, possuem os atuadores responsáveis pelas ocorrências dos eventos controláveis pertencentes ao conjunto $\Sigma_{c,i} \subseteq \Sigma_c$, sendo os conjuntos $\Sigma_{c,i}$, $i \in I_n$, disjuntos. Portanto, $\Sigma_c = \bigcup_{i \in I_n} \Sigma_{c,i}$.

A ação de controle emitida pelo supervisor S_{net} é transmitida para cada local de atuação AS_i , $i \in I_n$, por meio de um canal de comunicação distinto, referido como canal de controle cch_i . Semelhante aos canais de observação, cada

canal de controle cch_i , $i \in I_n$, é modelado por uma fila FIFO e está sujeito a atrasos de comunicação limitados a $k_{c,i} \in \mathbb{Z}_+$ passos.

O sistema composto pelo SED-MCO, pelos locais de atuação e pelos canais de controle é referido como um SEDR, o qual é formalmente definido a seguir.

Definição 2. (SEDR). Um SEDR é um par $\mathcal{G} = (\mathcal{G}_{mco}, C)$, em que $\mathcal{G}_{mco} = (G, O)$ é um SED-MCO e C é o vetor de controle, o qual é um vetor $1 \times n$ cujo i -ésimo componente é $[C]_i = (\Sigma_{c,i}, k_{c,i})$. \square

A cada nova observação de um evento pertencente a Σ_{so} , o supervisor em rede S_{net} calcula uma nova ação de controle $\gamma \in V$. Em decorrência da distribuição dos locais de atuação, a ação de controle é particionada e cada parte é enviada separadamente para seu respectivo local de atuação, isto é, S_{net} envia as ações de controle $\gamma_i = \gamma \cap \Sigma_{c,i}$, $i = 1, \dots, n$, para os respectivos locais de atuação AS_i , por meio dos canais cch_i .

Com o objetivo de distinguir o ato de emitir uma ação de controle γ_i , $i \in I_n$, e o momento da efetiva aplicação dessa ação de controle em AS_i , define-se o novo evento $\gamma_{i,se}$, adicionando-se simplesmente o subscrito i, se a direita de γ_i . Por exemplo, a efetiva aplicação da ação de controle $\gamma_1 = \{a\}$ (a qual é um conjunto) é representada pelo evento $\gamma_{1,se} = \{a\}_{1,se}$. A partir da introdução desses novos eventos $\gamma_{i,se}$, definem-se os conjuntos $\Sigma_{se,i} = \{\gamma_{i,se} : \gamma_i \in 2^{\Sigma_{c,i}}\}$ e $\Sigma_{se} = \bigcup_{i \in I_n} \Sigma_{se,i}$. Por exemplo, supondo $\Sigma_{c,1} = \{a, b\}$ e $\Sigma_{c,2} = \{c\}$, têm-se $\Sigma_{se,1} = \{\emptyset_{1,se}, \{a\}_{1,se}, \{b\}_{1,se}, \{a, b\}_{1,se}\}$, $\Sigma_{se,2} = \{\emptyset_{2,se}, \{c\}_{2,se}\}$ e $\Sigma_{se} = \{\emptyset_{1,se}, \{a\}_{1,se}, \{b\}_{1,se}, \{a, b\}_{1,se}, \emptyset_{2,se}, \{c\}_{2,se}\}$.

Pode-se, então, introduzir o mapeamento $F_{ec} : \Sigma_{se}^* \rightarrow 2^{\Sigma_c}$ que retorna o conjunto de eventos controláveis habilitados após uma sequência de eventos pertencentes a Σ_{se}^* , definido recursivamente a seguir: (i) $F_{ec}(\varepsilon) = \emptyset$, e (ii) $\forall s_{se} \in \Sigma_{se}^*$ e $\forall \gamma_{i,se} \in \Sigma_{se,i}$ com $i \in I_n$, $F_{ec}(s_{se} \gamma_{i,se}) = (F_{ec}(s_{se}) \setminus \Sigma_{c,i}) \cup \gamma_i$, em que γ_i é a ação de controle cuja efetiva aplicação é representada pelo evento $\gamma_{i,se}$. Vale ressaltar que (i) é decorrente da hipótese de que a planta G é inicializada com todos os eventos controláveis desabilitados, isto é, um local de atuação só poderá executar um evento controlável após receber a primeira ação de controle emitida pelo supervisor. Por sua vez, (ii) é consequência do fato de que, dentre os eventos pertencentes a $\Sigma_{c,i}$, apenas aqueles que estão em γ_i estarão habilitados a ocorrer em AS_i após a ocorrência do evento $\gamma_{i,se}$.

Exemplo 2. Considere o SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ mostrado na Figura 5, no qual \mathcal{G}_{mco} é o SED-MCO descrito no Exemplo 1 e $C = ((\{\beta, \mu\}, 0), (\{\lambda\}, 2))$ é o vetor de controle. Por fim, considere que os estados 5 e 6 da planta são proibidos, isto é, G nunca deve alcançá-los, o que é equivalente a dizer que o autômato G_{mco} não deve alcançar os estados $(5, o_1)$, $(6, o_1)$ e $(6, o_4)$.

Suponha que G_{mco} esteja no estado $(3, o_6)$ e d seja o único evento controlável habilitado na planta. Suponha agora que, após receber a observação b_{so} o supervisor S_{net} emita a ação de controle $\gamma = \{c\}$. Assim, $\gamma_1 = \{c\}$, $\gamma_2 = \emptyset$, $\gamma_{1,se} = \{c\}_{1,se}$, $\gamma_{2,se} = \emptyset_{2,se}$, e como $k_{c,2} > k_{c,1}$, pode acontecer de AS_1 receber γ_1 até dois passos antes de AS_2 receber γ_2 . Se isso ocorrer, então o conjunto de eventos

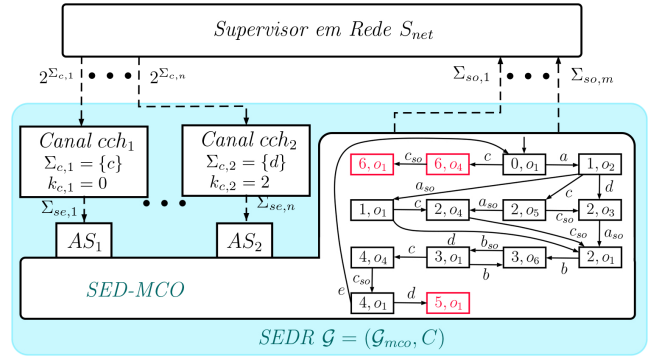


Figura 5. SEDR do Exemplo 2.

controláveis habilitados entre as efetivas aplicações das ações de controle $\gamma_{1,se}$ e $\gamma_{2,se}$ será $\{c, d\}$ ao invés de $\{c\}$, permitindo que G_{mco} execute c , c_{so} e d , alcançando o estado proibido $(5, o_1)$. \square

Como ilustrado no Exemplo 2, no problema estudado nessa artigo, tem-se a dificuldade adicional de lidar com a transmissão das ações de controle por meio de múltiplos canais sujeitos a diferentes atrasos de comunicação. Como consequência, ações de controle podem ser aplicadas com diferentes atrasos em cada local de atuação, o que não é considerado em (Park e Cho, 2007; Lin, 2014; Shu e Lin, 2017; Zhu et al., 2019; Rashidinejad et al., 2018).

3.2 Supervisor em Rede

O supervisor em rede S_{net} , como mostrado na Figura 6, é formado por três dispositivos: um receptor, um supervisor S e um emissor, sendo, portanto, uma tripla $S_{net} = (S, receptor, emissor)$.

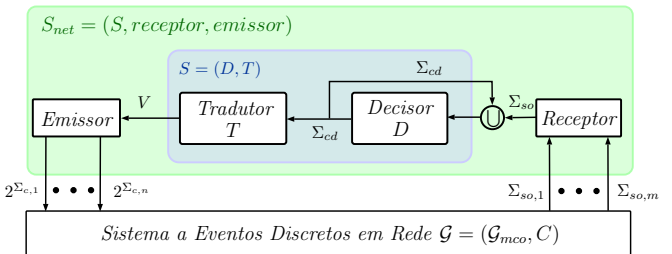


Figura 6. Supervisor em Rede S_{net} .

O receptor é responsável por captar as ocorrências de eventos transmitidas por meio dos canais de observação e, com isso, informar ao supervisor S as observações de eventos bem sucedidas representadas pelos eventos em Σ_{so} .

Toda vez que o supervisor S observa um novo evento em Σ_{so} , ele emite uma nova ação de controle $\gamma \in V$ que é transferida para o emissor, o qual transmite as ações de controle $\gamma_i = \gamma \cap \Sigma_{c,i}$, $i \in I_n$, para os locais de atuação AS_i por meio dos canais de comunicação cch_i .

O supervisor S é modelado como um par $S = (D, T)$, em que D e T são mapeamentos denominados decisor e tradutor, respectivamente. A estrutura de controle realimentado incluindo D e T é apresentada na Figura 6, em que $\Sigma_{cd} = \{\delta_{cd} : \delta \in 2^{\Sigma_c}\}$ é um conjunto de eventos formados adicionando-se o subscrito cd aos conjuntos que representam as possíveis decisões de controle. Por exemplo, se $\Sigma_c = \{c, d\}$, então as possíveis decisões de

controle são \emptyset , $\{c\}$, $\{d\}$ e $\{c, d\}$ (conjuntos), a assim, $\Sigma_{cd} = \{\emptyset_{cd}, \{c\}_{cd}, \{d\}_{cd}, \{c, d\}_{cd}\}$.

O *decisor* D é um mapeamento $D : \{\varepsilon\} \cup (\Sigma_{so} \cup \Sigma_{cd})^* \Sigma_{so} \rightarrow \Sigma_{cd}$ que emite uma decisão de controle $\delta_{cd} \in \Sigma_{cd}$ após cada nova observação de evento da planta, sendo essa decisão baseada na sequência formada pelos eventos em Σ_{so} que foram observados e pelas decisões de controle emitidas anteriormente por D . A inclusão da sequência vazia ε no domínio de D está associada a inicialização do sistema de controle supervisorio (decisão de controle inicial).

O *tradutor* T é o mapeamento $T : \Sigma_{cd} \rightarrow V$ definido como $T(\delta_{cd}) = \delta \cup \Sigma_{uc}$, em que $\delta \in 2^{\Sigma_c}$ é o conjunto formado pelos eventos controláveis habilitados pela decisão de controle $\delta_{cd} \in \Sigma_{cd}$. Por exemplo, se $\Sigma_{cd} = \{\emptyset_{cd}, \{c\}_{cd}, \{d\}_{cd}, \{c, d\}_{cd}\}$, então $T(\emptyset_{cd}) = \Sigma_{uc}$, $T(\{c\}_{cd}) = \{c\} \cup \Sigma_{uc}$, $T(\{d\}_{cd}) = \{d\} \cup \Sigma_{uc}$, e $T(\{c, d\}_{cd}) = \{c, d\} \cup \Sigma_{uc}$. O *tradutor* T mapeia cada saída do *decisor* D em uma ação de controle admissível pertencente a V .

O supervisor S pode ser visto como um mapeamento composto $S : \{\varepsilon\} \cup (\Sigma_{so} \cup \Sigma_{cd})^* \Sigma_{so} \rightarrow V$ em que $S(w) = T(D(w))$. Contudo, optou-se por particioná-lo como $S = (D, T)$ em decorrência do fato de que o projeto do supervisor em rede $S_{net} = ((D, T), \text{receptor}, \text{emissor})$ consiste unicamente no projeto do *decisor* D . Note que o *receptor* e o *emissor* são *hardware*, e o projeto do *tradutor* T é imediato, como explicado anteriormente.

4. MODELAGEM POR AUTÔMATO

Nesta seção, um algoritmo para a construção de um autômato G_{net} associado a um dado SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ é proposto. O autômato G_{net} será usado na verificação da existência de supervisores em rede. Essa modelagem estende a proposta por Nunes et al. (2018) nos seguintes aspectos: (i) G_{net} caracteriza os efeitos de ambos os tipos de atraso, de observação e de controle, e (ii) G_{net} leva em conta a ação do supervisor em rede sobre a planta, considerando as restrições impostas aos eventos que podem ser gerados pela planta.

Conforme descrito na Seção 3.2, o projeto do supervisor em rede consiste unicamente em determinar o *decisor* D . Com isso em mente, o autômato G_{net} modela o comportamento de um dado SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ considerando os efeitos de todas as possíveis decisões D , isto é, os estados de G_{net} alcançados após uma nova observação bem sucedida possuem transições com todos os eventos em Σ_{cd} . Para isso, o conjunto de eventos de G_{net} é definido como

$$\Sigma_{net} = \Sigma \cup \Sigma_{so} \cup \Sigma_{se} \cup \Sigma_{cd}. \quad (1)$$

Definem-se, também, as projeções a seguir:

$$\begin{aligned} P_{net} &: \Sigma_{net}^* \rightarrow \Sigma^* \\ P_{net,mco} &: \Sigma_{net}^* \rightarrow (\Sigma \cup \Sigma_{so})^* \\ P_{net,se} &: \Sigma_{net}^* \rightarrow \Sigma_{se}^* \\ P_{net,o} &: \Sigma_{net}^* \rightarrow \Sigma_{net,o}^* \end{aligned}$$

em que $\Sigma_{net,o} = \Sigma_{so} \cup \Sigma_{cd}$ é o conjunto de eventos observáveis do ponto de vista do *decisor* D . Uma sequência $s_{net} \in L(G_{net}) \subset \Sigma_{net}^*$ modela o caso em que a sequência $s = P_{net}(s_{net})$ ocorreu na planta, a sequência $P_{net,o}(s_{net})$ foi observada pelo *decisor* D , e o conjunto de eventos controláveis de fato habilitados é $F_{ec}(P_{net,se}(s_{net}))$. Além disso,

a sequência s_{net} corresponde a sequência $P_{net,mco}(s_{net})$ no autômato G_{mco} , no qual se modela apenas os atrasos de observação.

Os estados de G_{net} são quádruplas $x_{net} = (x_{mco}, Q_c, E_c, cd)$ em que: (i) $x_{mco} \in X_{mco}$ é o estado atual do autômato G_{mco} que modela o SED-MCO \mathcal{G}_{mco} , (ii) Q_c é um vetor $1 \times n$ cuja i -ésima componente $[Q_c]_i$ é uma sequência que caracteriza o *status* atual do canal de controle ch_i , (iii) $E_c \in 2^{\Sigma_c}$ é o conjunto formado pelos eventos controláveis habilitados na planta durante o estado x_{net} , e (iv) $cd \in \{0, 1\}$ é um variável booleana que é igual a um se o *decisor* D estiver calculando uma nova decisão de controle e igual a zero, caso contrário.

Um ponto chave na construção do autômato G_{net} é definir Q_c de tal forma que essa componente do estado caracterize o *status* atual dos canais de controle. Com isso em mente, define-se o conjunto $Q_{c_i} = \{0\} \cup \{q = \delta_1 k_1 \dots \delta_p k_p : (l \in \{1, \dots, p\})[\delta_l \in 2^{\Sigma_{c,i}} \wedge k_l \in \mathbb{Z}_+]\}$. Assim, o componente Q_c estará no espaço $Q_c^n = Q_{c_1} \times \dots \times Q_{c_n}$.

O cálculo do componente Q_c é executado com o auxílio de quatro operações, quais sejam: (i) operação *add*, que é usada para adicionar 1 a todos os indicadores de atraso do vetor $Q_c \in Q_c^n$, (ii) operação *link_{cd}*, que é usada para inserir ações de controle emitidas pelo supervisor ao vetor $Q_c \in Q_c^n$, (iii) operação *rem*, que é usada para remover ações de controle de $Q_c \in Q_c^n$ que foram transmitidas com sucesso, e (iv) operação *D_{max}*, que retorna 1 quando existe uma ação de controle em Q_c cujo indicador de atraso já alcançou o limite máximo de atraso. Essas operações são formalmente definidas a seguir.

Definição 3.

• A operação *add* é o mapeamento $add : Q_c^n \rightarrow Q_c^n$ em que $add(Q) = \tilde{Q}$ tal que, $\forall l \in I_n$,

$$[\tilde{Q}]_l = \begin{cases} q_1(k_1 + 1) \dots q_p(k_p + 1), & \text{se } [Q]_l = q_1 k_1 \dots q_p k_p, \\ 0, & \text{se } [Q]_l = 0. \end{cases}$$

• a operação *link_{cd}* é o mapeamento $link_{cd} : Q_c^n \times V \times V \rightarrow Q_c^n$ em que, dado que uma nova ação de controle γ foi emitida e γ_{pre} foi a ação de controle anterior emitida por S , $link_{cd}(Q_c, \gamma_{pre}, \gamma) = \tilde{Q}_c$ tal que, $\forall i \in I_n$,

$$[\tilde{Q}_c]_i = \begin{cases} [Q_c]_i (\gamma \cap \Sigma_{c,i}) 0, & \text{se } (\gamma \cap \Sigma_{c,i} \neq \gamma_{pre} \cap \Sigma_{c,i}) \\ & \wedge ([Q_c]_i \neq 0), \\ (\gamma \cap \Sigma_{c,i}) 0, & \text{se } (\gamma \cap \Sigma_{c,i} \neq \gamma_{pre} \cap \Sigma_{c,i}) \\ & \wedge ([Q_c]_i = 0), \\ [Q_c]_i, & \text{caso contrário,} \end{cases}$$

• A operação *rem* é o mapeamento $rem : Q_c^n \times \mathbb{Z}_+ \rightarrow Q_c^n$ em que $rem(Q, ch) = \tilde{Q}$ tal que, $\forall i \in I_n$,

$$[\tilde{Q}]_l = \begin{cases} q_2 k_2 \dots q_{p_l} k_{p_l}, & \text{se } l = ch \wedge [Q]_l = q_1 k_1 q_2 k_2 \dots \\ & q_{p_l} k_{p_l} \text{ com } p_l > 1, \\ [Q]_l, & \text{se } l \neq ch, \\ 0, & \text{caso contrário.} \end{cases}$$

• A operação de verificação é o mapeamento $D_{max} : Q_c^n \rightarrow \{0, 1\}$ em que

$$D_{max}(Q_c) = \begin{cases} 1, & \text{se } (\exists i \in I_n) [[Q_c]_i = \delta_1 k_1 \dots \\ & \text{com } k_1 \geq k_{c,i}], \\ 0, & \text{caso contrário.} \end{cases}$$

□

O autômato $G_{net} = (X_{net}, \Sigma_{net}, f_{net}, x_{0_{net}})$ é construído de acordo com o Algoritmo 1.

Algoritmo 1. (Construção do autômato G_{net}).

Entradas: $\mathcal{G} = (\mathcal{G}_{mco}, C)$.

Saída: $G_{net} = (X_{net}, \Sigma_{net}, f_{net}, x_{0_{net}})$.

Passo 1 (Inicialização):

1.1: Calcule $G_{mco} = (X_{mco}, (\Sigma_{so} \cup \Sigma), f_{mco}, x_{0_{mco}})$ a partir de $\mathcal{G}_{mco} = (G, O)$ conforme proposto em (Nunes et al., 2018) ou (Alves e Basilio, 2019).

1.2: Defina $\Sigma_{net} = \Sigma \cup \Sigma_{so} \cup \Sigma_{se} \cup \Sigma_{cd}$.

1.3: Defina $x_{0_{net}} = (x_{0_{mco}}, Q_{c0}, \emptyset, 1)$, em que $Q_{c0} \in \mathcal{Q}_c^n$ é um vetor com todos os componentes iguais a zero.

1.4: Defina $X_{net} = \{x_{0_{net}}\}$, crie um fila vazia F e adicione $x_{0_{net}}$ a F .

Passo 2: Enquanto $F \neq \emptyset$ faça:

2.1: Faça x_{net} igual ao primeiro estado em F , remova-o de F e defina $(x_{mco}, Q_c, E_c, cd) = x_{net}$.

2.2: Se $cd = 1$, então:

2.2.1: Calcule a última ação de controle emitida $\gamma_{pre} \in V$, da forma a seguir:

(a) Para cada $i \in I_n$ defina:

$$\gamma_c^i = \begin{cases} E_c \cap \Sigma_{c,i}, & \text{se } [Q_c]_i = 0, \\ \delta_{p_i}, & \text{se } [Q_c]_i = \delta_1 k_1 \dots \delta_{p_i} k_{p_i}, \delta_l \in 2^{\Sigma_{c,i}} \\ & \text{e } k_l \in \mathbb{Z}_+, \forall l \in \{1, \dots, p_i\}. \end{cases}$$

(b) $\gamma_{pre} = (\bigcup_{i \in I_n} \gamma_c^i) \cup \Sigma_{uc}$.

2.2.2: Para cada $\delta_{cd} \in \Sigma_{cd}$, faça:

(a) Calcule $\tilde{Q}_c = \text{link}_{cd}(Q_c, \gamma_{pre}, T(\delta_{cd}))$.

(b) Defina $\tilde{x} = f_{net}(x_{net}, \delta_{cd}) = (x_{mco}, \tilde{Q}_c, E_c, 0)$,

(c) Se $\tilde{x} \notin X_{net}$, então adicione \tilde{x} a F e faça $X_{net} = X_{net} \cup \{\tilde{x}\}$.

2.3: Se $cd = 0 \wedge D_{max}(Q_c) = 0$, então:

2.3.1: Para todo $\sigma \in [(\Sigma_{uc} \cup E_c) \cap \Gamma_{G_{mco}}(x_{mco})]$:

(a) Calcule $\tilde{Q}_c = \text{add}(Q_c)$.

(b) Defina $\tilde{x} = f_{net}(x_{net}, \sigma) = (x_{mco}, \tilde{Q}_c, \tilde{E}_c, 0)$.

(c) Se $\tilde{x} \notin X_{net}$, então adicione \tilde{x} a F e faça $X_{net} = X_{net} \cup \{\tilde{x}\}$.

2.4: Para cada canal de controle cch_i , $i \in I_n$, faça:

2.4.1: Se $[Q_c]_i \neq 0$, então:

(a) Seja $[Q_c]_i = \delta_1 k_1 \dots$, faça $\gamma_i = \delta_1$, e calcule $\tilde{Q}_c = \text{rem}(Q_c, i)$ e $\tilde{E}_c = (E_c \setminus \Sigma_{c,i}) \cap \gamma_i$.

(b) Defina $\tilde{x} = f_{net}(x_{net}, \gamma_{i,se}) = (x_{mco}, \tilde{Q}_c, \tilde{E}_c, cd)$, em que $\gamma_{i,se} \in \Sigma_{se}$ é o evento que representa a efetiva aplicação da ação de controle γ_i .

(c) Se $\tilde{x} \notin X_{net}$, então adicione \tilde{x} a F e faça $X_{net} = X_{net} \cup \{\tilde{x}\}$.

2.5: Para cada $\sigma_{so} \in (\Sigma_{so} \cap \Gamma_{G_{mco}}(x_{mco}))$, faça:

(a) Defina $\tilde{x} = f_{net}(x_{net}, \sigma_{so}) = (x_{mco}, \sigma_{so}, Q_c, E_c, 1)$.

(b) Se $\tilde{x} \notin X_{net}$, então adicione \tilde{x} a F e faça $X_{net} = X_{net} \cup \{\tilde{x}\}$.

Os resultados a seguir tratam da corretude do Algoritmo 1.

Lema 4. Para toda sequência $w \in L(G_{net})$, o estado $(x_{mco}, Q_c, E_c, cd) = f_{net}(x_{0_{net}}, w)$, em que $x_{mco} \in X_{mco}$, $Q_c \in \mathcal{Q}_c$, $E_c \in 2^{\Sigma_c}$ e $cd \in \{0, 1\}$, é tal que:

(i) $x_{mco} = f_{mco}(x_{0_{mco}}, P_{net,mco}(w))$;

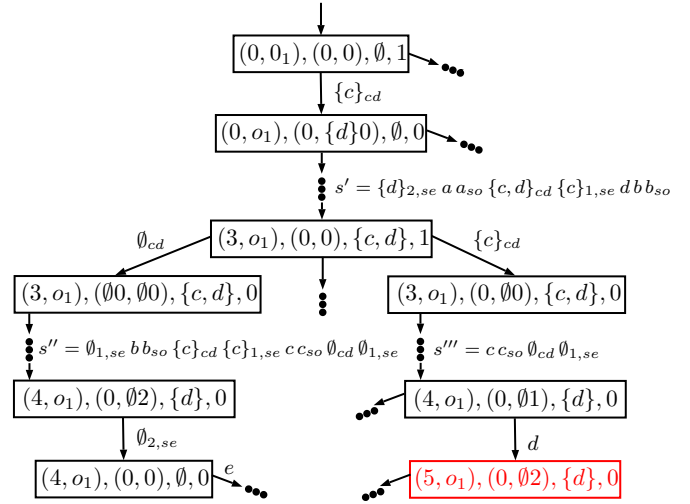


Figura 7. Parte do autômato G_{net} do Exemplo 3.

(ii) Para todo $i \in I_n$:

- (a) Se nenhuma ação de controle está sendo transmitida pelo canal cch_i após w , então $[Q_c]_i = 0$;
- (b) Se as ações de controle $\delta_1, \dots, \delta_{p_i} \in 2^{\Sigma_{c,i}}$ foram enviadas nessa ordem e estão sendo transmitidas por cch_i após w , então $[Q_c]_i = \delta_1 k_1 \dots \delta_{p_i} k_{p_i}$, em que $k_t \in \mathbb{Z}_+$, $t = 1, \dots, p_i$, é igual ao número de ocorrências de eventos na planta desde o início da transmissão de δ_t ;

(iii) $E_c = F_{ec}(P_{net,se}(w))$;

(iv) $cd = 1 \Leftrightarrow P_{net,o}(w) \in (\{\varepsilon\} \cup \Sigma_{net,o}^* \Sigma_{so})$. \square

O item (i) do Lema 4 mostra que uma sequência de $L(G_{net})$ pode ser obtida a partir de uma sequência de $L(G_{mco})$ inserindo-se apenas eventos pertencentes a $\Sigma_{se} \cup \Sigma_{cd}$. Por sua vez, como demonstrado em (Alves e Basilio, 2019), as sequências em $L(G_{mco})$ podem ser obtidas a partir de sequências em $L(G)$ inserindo-se apenas eventos pertencentes a Σ_{so} . Como consequência, pode-se concluir que o autômato G_{net} preserva o comportamento da planta G e sua linguagem. O item (ii) mostra que a segunda componente dos estados de G_{net} caracteriza corretamente o *status* do canal de controle. Adicionalmente, de acordo com o item (iii) do Lema 4, a terceira componente dos estados de G_{net} armazena o conjunto de eventos controláveis efetivamente habilitados na planta. Por fim, de acordo com o item (iv), a quarta componente é igual a um (1) no estado inicial de G_{net} e nos seus estados alcançados após a ocorrência de um evento de observação bem sucedida pertencente a Σ_{so} , o que também indica que o *decisor* D pode computar uma nova decisão de controle.

Exemplo 3. Considere, novamente, o SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ apresentado no Exemplo 2. Por restrições de espaço, foi possível mostrar apenas parte do autômato G_{net} desse SEDR na Figura 7, na qual pode-se observar que, no caminho da esquerda, as decisões de controle foram capazes de impedir que G_{mco} alcançasse o estado proibido $(5, 0_1)$, uma vez que d se encontra desabilitado no estado $((4, 0_1), (0, 0), \emptyset, 0)$. Por outro lado, as decisões de controle tomadas no decorrer da sequência da direita, falharam em impedir o alcance desse estado proibido. O êxito na sequência da esquerda é decorrente da postergação da habilitação

de c até a ocorrência de uma nova observação b_{so} , ao invés de emití-la já no estado $((3, o_1), (0, 0), \{c, d\}, 1)$. \square

Lema 5. Dado um SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ e seu autômato G_{net} correspondente, obtido de acordo com o Algoritmo 1. Para todo $w \in L(G_{net})$ e $\sigma \in \Sigma_{net}$, $w\sigma \in L(G_{net})$ se, e somente se, σ pode ser gerado em $\mathcal{G} = (\mathcal{G}_{mco}, C)$ após a ocorrência de w . \square

Teorema 6. Dado um SEDR $\mathcal{G} = (\mathcal{G}_{mco}, C)$ e seu autômato G_{net} correspondente, obtido de acordo com o Algoritmo 1. Então, $L(G_{net})$ é a linguagem sobre Σ_{net}^* que caracteriza o comportamento de \mathcal{G} na presença de atrasos de observação e controle. \square

5. PROBLEMA DE CONTROLE

Nesta seção, formula-se o problema de controle supervisorio em rede de SEDR sujeitos a atrasos de comunicação, usando para isso o modelo G_{net} apresentado na Seção 4. Note que, em G_{net} , quando o supervisor em rede deseja prevenir a ocorrência de um determinado evento $\sigma \in \Sigma_c$ na planta, ele não pode desabilitar σ diretamente, mas pode escolher uma decisão de controle (representada por um evento em Σ_{cd}) que não inclua σ , e enviá-la para a planta. Dessa forma, o conjunto de eventos de G_{net} é particionado quanto a controlabilidade como $\Sigma_{net} = \Sigma_{net,c} \dot{\cup} \Sigma_{net,uc}$, em que $\Sigma_{net,c} = \Sigma_{cd}$ (resp. $\Sigma_{net,uc} = \Sigma \cup \Sigma_{so} \cup \Sigma_{se}$) é o conjunto de eventos controláveis (resp. não-controláveis) de G_{net} . Por outro lado, conforme estabelecido na Seção 4, Σ_{net} é particionado em relação a observabilidade como $\Sigma_{net} = \Sigma_{net,o} \dot{\cup} \Sigma_{net,uo}$, em que $\Sigma_{net,o} = \Sigma_{cd} \cup \Sigma_{so}$ (resp. $\Sigma_{net,uo} = \Sigma \cup \Sigma_{se}$) é o conjunto de eventos observáveis (resp. não-observáveis) de G_{net} .

A linguagem gerada por um SEDR sob a ação de um supervisor em rede S_{net} é apresentada a seguir.

Definição 7. ($L(S_{net}/G_{net})$). Dado um SEDR com a estrutura mostrada na Figura 4, seu autômato G_{net} correspondente e um supervisor em rede $S_{net} = ((D, T), receptor, emissor)$, a linguagem do sistema em malha fechada é recursivamente definida da forma a seguir:

(i) $\varepsilon \in L(S_{net}/G_{net})$,

(ii) Para todo $w \in \Sigma_{net}^*$ e $\sigma \in \Sigma_{net}$, $w\sigma \in L(S_{net}/G_{net}) \Leftrightarrow (w \in L(S_{net}/G_{net})) \wedge (w\sigma \in L(G_{net})) \wedge [(\sigma \in \Sigma_{net,uc}) \vee (P_{net,o}(w) \in (\{\varepsilon\} \cup \Sigma_{net,o}^* \Sigma_{so})) \wedge \sigma = D(P_{net,o}(w))]$. \square

Note que, de acordo com a Definição 7, o supervisor em rede S_{net} não desabilita eventos não-controláveis em $\Sigma_{net,uc}$. Por outro lado, na inicialização do sistema e após cada nova observação bem sucedida, isto é, quando $P_{net,o}(w) \in \{\varepsilon\} \cup \Sigma_{net,o}^* \Sigma_{so}$, S_{net} habilita apenas uma decisão de controle Σ_{cd} , a qual é determinada por $D(P_{net,o}(w))$.

Pode-se, então, formular o seguinte problema de controle.

Problema 1. Dado um SEDR com a estrutura mostrada na Figura 4, seu autômato G_{net} correspondente e uma linguagem prefixo-fechada $K_{net} \subseteq L(G_{net})$, encontrar um supervisor em rede S_{net} que satisfaça os seguintes requerimentos:

R1. $L(S_{net}/G_{net}) \subseteq K_{net}$.

R2. Não existe um supervisor em rede S'_{net} que satisfaça **R1** e $L(S_{net}/G_{net}) \subset L(S'_{net}/G_{net})$. \square

O requerimento **R1** exige que o supervisor S_{net} seja seguro, e requerimento **R2** demanda um supervisor em rede S_{net} que seja maximamente permissivo.

6. EXISTÊNCIA DE SUPERVISORES EM REDE

Com o objetivo de apresentar um condição necessária e suficiente para a existência de um supervisor em rede para o Problema 1, propõe-se, nesta seção, uma nova propriedade de linguagem, denominada factibilidade, que em conjunto com a controlabilidade e a normalidade, garante a existência de um supervisor em rede. A ideia por trás da factibilidade é que, na inicialização do sistema e toda vez que uma nova observação bem sucedida ocorrer, deve existir ao menos uma decisão de controle em Σ_{cd} que seja capaz de manter o sistema dentro da linguagem de especificação.

Definição 8. Uma linguagem $K_f \subseteq L(G_{net})$ é factível em relação a $\Sigma_{net,o}$, Σ_{so} , Σ_{cd} e $P_{net,o}$ se, $\forall w \in \overline{K_f}$, $(P_{net,o}(w) \in (\{\varepsilon\} \cup \Sigma_{net,o}^* \Sigma_{so}) \Rightarrow (\exists \delta_{cd} \in \Sigma_{cd}, w\delta_{cd} \in \overline{K_f}))$. \square

Seja $\mathcal{C}_f(K_{net}) = \{K_f \subseteq K_{net} : K_f \text{ é factível e.r.a } \Sigma_{net,o}, \Sigma_{so}, \Sigma_{cd} \text{ e } P_{net,o}\}$ a classe formada por todas as sublinguagens factíveis de K_{net} . Pode-se facilmente verificar, a partir da Definição 8, que $\mathcal{C}_f(K_{net})$ é fechada sobre a união de linguagens e, adicionalmente, a linguagem vazia sempre pertence a $\mathcal{C}_f(K_{net})$. Sejam $\mathcal{C}_c(K_{net})$ e $\mathcal{C}_n(K_{net})$ as classes de sublinguagens controláveis e sublinguagens normais de K_{net} , respectivamente. Sabe-se que a linguagem vazia pertence a ambas $\mathcal{C}_c(K_{net})$ e $\mathcal{C}_n(K_{net})$, e que ambas são também fechadas sobre a união (Cassandras e Lafortune, 2008). Consequentemente, pode-se concluir que a sublinguagem controlável, normal e factível suprema de K_{net} , denotada por $supCNF(K_{net})$, sempre existe e, no pior caso, é igual a linguagem vazia.

Apresenta-se, a seguir, uma condição necessária e suficiente para a existência de um supervisor em rede que resolve o Problema 1.

Teorema 9. Dado um SEDR com a estrutura mostrada na Figura 4, seu autômato G_{net} correspondente e uma linguagem prefixo-fechada $K_{net} \subseteq L(G_{net})$. Então, existe S_{net} que satisfaz os requerimentos **R1** e **R2** do Problema 1 se, e somente se, $supCNF(K_{net}) \neq \emptyset$. \square

A verificação da condição de existência proposta no Teorema 9 pode ser feita construindo-se um autômato H_{cnf} que gera a linguagem $supCNF(K_{net})$, conforme descrito no Algoritmo 2.

Algoritmo 2. (Construção de H_{cnf}).

Entradas: G_{net} , e H_{net} tal que $L(H_{net}) = K_{net}$.

Output: $H_{cnf} = (X_{cnf}, \Sigma_{net}, f_{cnf}, x_{0_{cnf}})$.

Passo 1: Defina $H'_{net} = H_{net}$.

Passo 2: Calcule o autômato H_{net}^{CN} que gera a sublinguagem controlável e normal suprema de $L(H'_{net})$.¹

¹ Algoritmos para o cálculo da sublinguagem controlável e normal suprema podem ser encontrados em (Brandt et al., 1990; Hashtrudi Zad et al., 2005).

Passo 3: Construa $R = (X_r, \Sigma_{net}, f_r, x_{0_r}) = H_{net}^{CN} \parallel G_{net}$.

Passo 4: Faça $X_{del} = \emptyset$.

Passo 5: Para cada estado $(x_{cn}, x_{net}) \in X_r$ faça:

5.1: Se $\Gamma_{G_{net}}(x_{net}) \cap \Sigma_{cd} \neq \emptyset \wedge \Gamma_R((x_{cn}, x_{net})) \cap \Sigma_{cd} = \emptyset$, então faça $X_{del} = X_{del} \cup \{x_{cn}\}$.

Passo 6: Se $X_{del} \neq \emptyset$, então:

6.1: Para cada estado $x_{cn} \in X_{del}$, remova x_{cn} de H_{net}^{CN} .

6.2: Faça $H'_{net} = Ac(H_{net}^{CN})$ e retorne para o *Passo 2*.

Passo 7: Defina $H_{cnf} = H_{net}^{CN}$.

Por fim, pode-se enunciar o resultado a seguir que trata da corretude do Algoritmo 2.

Proposição 10. $L(H_{cnf}) = supCNF(K_{net})$. □

Exemplo 4. Considere o SEDR do Exemplo 2. Pode-se, então, verificar que existe um supervisor S_{net} capaz de evitar que os estados proibidos $(5, o_1)$, $(6, o_1)$ e $(6, o_4)$ de G_{mco} sejam alcançados, executando-se o Algoritmo 2, usando como entradas o autômato G_{net} obtido com o Algoritmo 1 e o autômato H_{net} obtido a partir de G_{net} removendo-se os estados cuja primeira componente é um dos estados proibidos de G_{mco} . De fato, o supervisor S_{net} apresentado na Figura 8 é uma solução. □

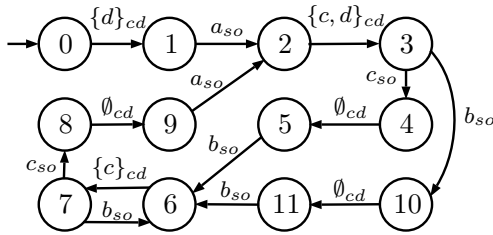


Figura 8. Supervisor em rede S_{net} (Exemplo 4).

7. CONSIDERAÇÕES FINAIS

Nesse artigo, foi proposto um método para construir um modelo de autômato para SEDR que leva em conta todos os efeitos possíveis de atrasos de controle e observação. Com base neste modelo, formulou-se um problema de controle supervisorório em rede e apresentou-se uma forma sistemática para a verificação da existência de uma solução para esse problema. Como continuação desse trabalho, pretende-se investigar como projetar o supervisor em rede S_{net} a partir do autômato H_{cnf} obtido por meio do Algoritmo 2.

REFERÊNCIAS

Alves, M.V.S. e Basilio, J.C. (2019). State estimation and detectability of networked discrete event systems with multi-channel communication networks. In *2019 American Control Conference (ACC)*, 5602–5607. Philadelphia, PA, USA.

Alves, M.V.S., Carvalho, L.K., e Basilio, J.C. (2017). Supervisory control of timed networked discrete event systems. In *56th Conference on Decision and Control*, 4859–4865. Melbourne, Australia.

Alves, M.V.S., Carvalho, L.K., e Basilio, J.C. (2021). Supervisory control of networked discrete event systems with timing structure. *IEEE Transactions on Automatic Control*, 66(5), 2206–2218.

Brandt, R., Garg, V., Kumar, R., Lin, F., Marcus, S., e Wonham, W. (1990). Formulas for calculating supremal controllable and normal sublanguages. *Systems & Control Letters*, 15(2), 111–117.

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

Davis, R., Kollmann, S., Pollex, V., e Slomka, F. (2011). Controller area network (can) schedulability analysis with fifo queues. In *2011 23rd Euromicro Conference on Real-Time Systems*, 45–56.

Hashtrudi Zad, S., Moosaei, M., e Wonham, W. (2005). On computation of supremal controllable, normal sublanguages. *Systems & Control Letters*, 54(9), 871–876.

Jasperneite, E. e Neumann, P. (2001). Switched ethernet for factory communication. In *8th International Conference on Emerging Technologies and Factory Automation*, 205–212.

Kurose, J. e Ross, K. (2017). *Computer Networking: A Top-Down Approach*. Pearson, Upper Saddle River, NJ, USA, 7ª edição.

Lin, F. (2014). Control of networked discrete event systems: Dealing with communication delays and losses. *SIAM J. Control Optim.*, 52(2), 1276–1298.

Nunes, C.E.V., Moreira, M.V., Alves, M.V.S., Carvalho, L.K., e 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.

Park, S.J. e Cho, K.H. (2007). Supervisory control of discrete event systems with communication delays and partial observations. *Systems & Control Letters*, 56(2), 106–112.

Ramadge, P.J.G. e Wonham, W.M. (1987). Supervisory control of a class of discrete event process. *SIAM Journal of Control Optimization*, 25(1), 206–230.

Ramadge, P.J.G. e Wonham, W.M. (1989). The control of discrete event systems. *Proc. of the IEEE*, 77(1), 81–98.

Rashidinejad, A., Reniers, M., e Feng, L. (2018). Supervisory control of timed discrete-event systems subject to communication delays and non-FIFO observations. In *14th IFAC Workshop on Discrete Event Systems WODES*, volume 51, 456 – 463. Naples, Italy.

Shu, S. e Lin, F. (2017). Deterministic networked control of discrete event systems with nondeterministic communication delays. *IEEE Transactions on Automatic Control*, 62(1), 190–205.

Tovar, E. e Vasques, F. (1999). Ieee transactions on industrial electronics. *IEEE Transactions on Automatic Control*, 46, 1241–1251.

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

Zhu, Y., Lin, L., Ware, S., e Su, R. (2019). Supervisor synthesis for networked discrete event systems with communication delays and lossy channels. In *2019 IEEE 58th Conference on Decision and Control (CDC)*, 6730–6735. Nice, France.