Opacidade em Controle Supervisório de Sistemas a Eventos Discretos em Rede

Jefté Marcelo Araujo Fontes * **Marcos Vinícius Silva Alves** *

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

Abstract: In this paper, the verification of opacity in discrete event systems is addressed by considering systems under the action of supervisors. It is assumed that an intruder is able to observe some event occurrences transmitted from the plant to the supervisor and some control actions transmitted from the supervisor to the plant. In order to limit the observations of control actions by the intruder, a sender device is included in the closed-loop structure. The estimation of the current plant state performed by the intruder is formally characterized, and a new concept of network opacity is proposed. In addition, a necessary and sufficient condition for network opacity is presented, and a method for its verification is developed.

Resumo: Neste artigo, a verificação da opacidade de sistemas a eventos discretos é abordada no contexto de sistemas sob a ação de supervisores. Supõe-se que um intruso tem a capacidade de observar tanto ocorrências de eventos transmitidas da planta para o supervisor, quanto ações de controle transmitidas do supervisor para a planta. Um dispositivo, denominado *Emissor*, é introduzido à arquitetura de controle supervisório com o objetivo de limitar a capacidade do intruso em observar ações de controle. A estimação de estados da planta feita do ponto de vista do intruso é caracterizada formalmente e um novo conceito de opacidade em rede é proposto. Por fim, propõe-se uma condição necessária e suficiente para um sistema em malha fechada ser opaco em rede e um método para verificá-la.

Keywords: opacity, supervisory control, automata, discrete event systems.

Palavras-chaves: opacidade, controle supervisório, autômatos, sistemas a eventos discretos.

1. INTRODUÇÃO

Diversos sistemas podem ser descritos, em um alto nível de abstração, por modelos com espaço de estados discreto e cuja dinâmica evolui em função de eventos assíncronos, sendo esses denominados Sistemas a Eventos Discretos (SED) (Cassandras e Lafortune, 2008). Utiliza-se frequentemente a denominação Sistema a Eventos Discretos em Rede (SEDR), para ressaltar a utilização das redes de comunicação nesse tipo de sistema (Lin, 2014; Alves et al., 2021; Alves e Basilio, 2023).

Como consequência do amplo uso de redes de comunicação, os sistemas de automação estão mais suscetíveis a vulnerabilidades de segurança (Cao et al., 2020; Carvalho et al., 2018; Alves et al., 2022). São denominados ataques passivos aqueles em que o intruso observa parte da informação que trafega pela rede de comunicação, porém, não realiza nenhuma alteração no conteúdo e no fluxo da informação que é transmitida. A confidencialidade de um sistema está relacionada a sua capacidade de impedir o acesso não autorizado a informações secretas. De maneira geral, a confidencialidade de um sistema pode ser garantida por meio de criptografia (Kurose e Ross, 2013; Lima et al., 2023). Contudo, sistemas de automação industriais operam, em muitos casos, em tempo real, gerando uma maior complexidade para a implementação de algoritmos de criptografia (Do et al., 2017). Como alternativa para garantir a segurança do sistema, uma abordagem frequentemente utilizada é garantir que a informação trafegada pelos canais vulneráveis não seja suficiente para o intruso acessar informações sensíveis (Wu et al., 2018).

Em SEDs, a confidenciabilidade tem sido estudada, em muitos casos, utilizando-se o conceito de opacidade (Jacob et al., 2016; Hadjicostis, 2020), no qual se supõe que o SED tem um segredo, definido em termos dos seus estados ou da sua linguagem. O conceito de opacidade tem sido estudado extensivamente nos últimos anos (Badouel et al., 2007; Lin, 2011; Wu e Lafortune, 2013; Ji et al., 2018; Keroglou e Hadjicostis, 2018; Barcelos e Basilio, 2021; Yang et al., 2021a), entretanto, os trabalhos citados anteriormente supõem que um intruso observa apenas eventos executados pela planta do sistema e não consideram a possibilidade do intruso ter acesso a informações oriundas de outros dispositivos que interagem com a planta, o que pode ser insuficiente para a análise da opacidade em sistemas nos quais existem supervisores que enviam ações de controle para a planta do sistema.

Neste artigo, a opacidade de SEDRs sob a ação de supervisores é investigada. Em SEDRs, o supervisor está normalmente posicionado distante da planta, a qual possui em geral uma arquitetura distribuída. Dessa forma, o sistema de controle supervisório de um SEDR utiliza dois conjuntos de canais para comunicação: os canais de

observação e os canais de controle. Os canais de observação (resp. controle) são utilizados para transmitir ocorrências de eventos (resp. ações de controle) da planta (resp. do supervisor) para o supervisor (resp. a planta).

Neste trabalho, a opacidade de SEDRs sob a ação de supervisores é analisada supondo a existência tanto de canais de observação vulneráveis quanto de canais de controle vulneráveis. Supõe-se também que o intruso conhece os modelos da planta e do supervisor. Propõe-se uma caracterização formal da estimação de estados da planta feita do ponto de vista do intruso, a qual leva em consideração a observação parcial de ambos, eventos ocorridos na planta e ações de controle emitidas pelo supervisor. Um algoritmo para a construção de um autômato observador usado para determinar a estimativa de estado da planta feita pelo intruso também é apresentado. Um novo conceito de opacidade, denominado opacidade em rede, é introduzido e apresenta-se uma condição necessária e suficiente para um SEDR ser opaco em rede.

O problema de controle supervisório de SEDRs com canais de controle vulneráveis foi abordado anteriormente por Zhu et al. (2019), Lima et al. (2019, 2022) e Yang et al. (2021b). Em Zhu et al. (2019) e Lima et al. (2019, 2022), consideram-se ataques capazes de alterar as ações de controle enviadas pelo supervisor e, portanto, o problema em questão é a síntese de supervisores robustos a alterações de ações de controle transmitidas por canais vulneráveis, o que difere do problema abordado neste artigo, no qual se investigar a segurança da informação na presença de ataques passivos. Em Yang et al. (2021b), são considerados ataques passivos supondo canais de controle e de observação vulneráveis. Contudo, Yang et al. (2021b) consideram uma arquitetura de controle supervisório convencional, na qual o supervisor sempre envia uma ação de controle para a planta mesmo que ela seja idêntica à última ação enviada, o que não seria necessário, uma vez que a nova ação não altera os conjuntos de eventos habilitados e desabilitados. Neste artigo, um novo dispositivo, denominado *Emissor*, será adicionado à arquitetura de controle supervisório, o qual não retransmite ações de controle idênticas, evitando o trafego desnecessário de informação por canais vulneráveis e, consequentemente, interferindo na estimativa de estado da planta realizada pelo intruso.

2. FUNDAMENTOS TEÓRICOS

Neste artigo, a dinâmica de um sistema a eventos discretos é representada por um autômato $G = (X, \Sigma, f, \Gamma, x_0, X_m)$, em que X é um conjunto finito de estados, Σ é um conjunto finito de eventos, $f : X \times \Sigma \to X$ é a função (parcial) de transição de estados, $\Gamma : X \to 2^{\Sigma}$ é a função de eventos ativos, x_0 é o estado inicial e X_m é o conjunto de estados marcados. A função de transição f pode ser estendida da forma usual para $f : X \times \Sigma^* \to X$, em que Σ^* denota o fecho de Kleene Σ. As linguagens gerada e marcada por G são definidas como $L(G) = \{s \in \Sigma^* : (\exists x \in X)[f(x_0, s) = x]\}$ e $L_m(G) = \{s \in L(G) : f(x_0, s) \in X_m\}$, respectivamente.

O conjunto de eventos Σ 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). Uma projeção $P_o : \Sigma^* \to \Sigma_o^*$ é definida como: *(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. Sejam $G_i = (X_i, \Sigma_i, f_i, \Gamma_i, x_{0_i}, X_{m_i})$, $i = 1, 2$ dois autômatos, o autômato resultante da composição paralela (Cassandras e Lafortune, 2008, sec. 2.3) entre G_1 e G_2 é denotado por $G_1 \| G_2$. Adicionalmente, para um conjunto B (resp. sequência s), $|B|$ (resp. $\|s\|$) denota sua cardinalidade (resp. seu comprimento).

Um autômato não-determinístico (Cassandras e Lafortune, 2008, sec. 2.2.4) é uma sêxtupla $G_{nd} = (X, \Sigma \cup \{\varepsilon\}, f_{nd}, \Gamma, X_0, X_m)$, em que os componentes X, Γ e X_m são definidos da mesma forma que nos autômatos determinísticos, por outro lado, além de transições rotuladas por eventos em Σ, podem existir transições rotuladas por ε, $f_{nd} : X \times (\Sigma \cup \{\varepsilon\}) \to 2^X$ é a função de transição não-determinística, parcialmente definida em seu domínio, e $X_0 \subseteq X$ é um conjunto de estados iniciais.

O autômato observador de um autômato determinístico G em relação a Σ_o é um autômato determinístico denotado por $Obs(G, \Sigma_o)$, cujas linguagens gerada e marcada são, respectivamente, $P_o(L(G))$ e $P_o(L_m(G))$. Por sua vez, o autômato observador de um autômato não-determinístico G_{nd}, denotado por $Obs(G_{nd})$, é um autômato determinístico equivalente a G_{nd} em termos de linguagem, ou seja, $L(Obs(G_{nd})) = L(G)$ e $L_m(Obs(G_{nd})) = L_m(G)$ (Cassandras e Lafortune, 2008).

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 devem pertencer ao conjunto $V := \{\gamma \in 2^{\Sigma} : \Sigma_{uc} \subseteq \gamma\}$. Dessa forma, um supervisor S pode ser definido como uma função $S : P_o(L(G)) \to V$. Por outro lado, um supervisor S pode ser representado por um autômato de estados finitos G_S em que a composição paralela entre G_S e a planta G descreve o comportamento do sistema em malha fechada S/G (lê-se S controlando G). Para isso, $G_S = (X_S, \Sigma, f_S, \Gamma_S, x_{0_S}, X_S)$ é construído de tal forma que, para todo $s_o \in P_o(L(S/G))$, $S(s_o) = \Gamma_S(f_S(x_{0_S}, s_o))$. Consequentemente, $L(G_S \| G) = L(S/G)$ e $L_m(G_S \| G) = L_m(S/G)$. O autômato G_S é denominado uma realização do supervisor S.

Um autômato $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ é dito ser opaco de estado atual (Saboori e Hadjicostis, 2007) em relação a uma projeção P_o e a um conjunto de estados secretos $X_{sec} \subset X$ se, para todo $s \in L(G)$, $\big(f(x_0, s) \in X_{sec}\big) \Rightarrow \big(\exists s' \in L(G), f(x_0, s') \in X \setminus X_{sec} \wedge P_o(s') = P_o(s)\big)$.

3. ESTRUTURA DE SEDS EM REDE

A estrutura de controle supervisório de SEDRs considerada nesse projeto é apresentada na Figura 1, em que um

supervisor S restringe o comportamento do SEDR, que por sua vez é composto por uma planta, modelada por um autômato $G = (X, \Sigma, f, \Gamma, x_0, X_m)$, um conjunto de locais de atuação AS_i, com $i \in I_n$ e $I_n = \{1, ..., n\}$, um conjunto de locais de medição MS_j, com $j \in I_m$ e $I_m = \{1, ..., m\}$, e uma rede de comunicação com múltiplos canais.

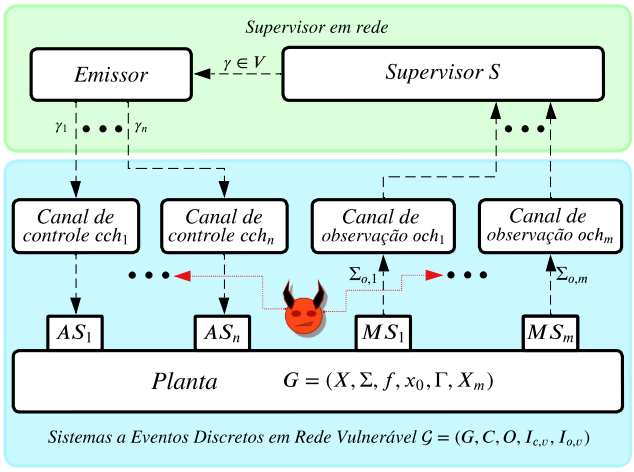

Figura 1. Estrutura de controle supervisório de um SEDR com canais vulneráveis.

Na estrutura mostrada na Figura 1, o local de atuação AS_i, $i \in I_n$, contém os atuadores responsáveis pela ocorrência dos eventos pertencentes ao subconjunto $\Sigma_{c,i} \subseteq \Sigma_c$. Por sua vez, o local de medição MS_j, $j \in I_m$, detecta ocorrências de eventos pertencentes ao subconjunto $\Sigma_{o,j} \subseteq \Sigma_o$. A rede de comunicação possui n canais de controle que conectam o supervisor a cada um do locais de atuação, sendo o canal que conecta o supervisor ao local de atuação AS_i denotado por cch_i. Do mesmo modo, a rede de comunicação também possui m canais de observação conectando cada local de medição MS_j com o supervisor, sendo esse canal denotado por och_j.

Supõe-se que os canais de controle e de observação podem ser classificados como vulneráveis ou seguros, de maneira que as informações trafegadas por um canal vulnerável podem ser acessadas por um intruso. Dessa classificação, surgem os conjuntos de índices a seguir:

$$I_{c,v} = \{i \in I_n : cch_i \text{ é vulnerável}\},$$
$$I_{o,v} = \{j \in I_m : och_j \text{ é vulnerável}\},$$

em que $I_{c,v}$ e $I_{o,v}$ são, respectivamente, o conjunto de índices dos canais de controle vulneráveis e o conjunto de índices dos canais de observação vulneráveis.

Sistemas a eventos discretos em rede com canais vulneráveis são definidos como se segue.

Definição 1. (SEDRV). Um sistema a eventos discretos em rede vulnerável (SEDRV) é uma quíntupla $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$, em que $G = (X, \Sigma, f, \Gamma, x_0, X_m)$ é uma autômato que modela a sua planta, C é o vetor de controle, de dimensão $1 \times n$ e cujo i-ésimo termo é $C_i = \Sigma_{c,i}$, O é o vetor de observação, de dimensão $1 \times m$ e cujo j-ésimo termo é $O_j = \Sigma_{o,j}$, $I_{c,v}$ é o conjunto de índices dos canais de controle vulneráveis e $I_{o,v}$ é o conjunto de índices dos canais de observação vulneráveis. □

Nesse artigo, considera-se um SEDRV sob a ação de um supervisore S previamente projetado, que restringe o comportamento do sistema em malha fechada de tal forma que $L(S/G)$ satisfaça um conjunto de especificações de controle. Adicionalmente, uma nova arquitetura em malha fechada é proposta, na qual é inserido um novo dispositivo, denominado *Emissor*. O *Emissor* será responsável por transmitir as ações de controle geradas pelo supervisor S para cada um dos locais de atuação. Quando uma ação de controle $\gamma \in V$ é gerada por S, o *Emissor* enviará $\gamma_i = \gamma \cap \Sigma_{c,i}$ para cada local de atuação $AS_i, i \in I_n$, pois AS_i atua apenas sobre os eventos em $\Sigma_{c,i}$. Note que, para um evento $\sigma \in \Sigma_{c,i}$ com $i \in I_n \setminus I_{c,v}$, isto é, um evento σ cuja ação de controle é transmitida por um canal de controle seguro, o intruso nunca saberá se σ está habilitado ou desabilitado.

Para ilustrar a principal motivação por trás da inclusão do *Emissor* no sistema em malha fechada, considere que, após a emissão de uma ação de controle $\gamma \in V$, um novo evento $\sigma \in \Sigma_o$ é observado pelo supervisor, que gera uma nova ação de controle γ'. Com o intuito de diminuir o tráfego de informação que, inclusive, pode ser observada pelo intruso, o *Emissor* não retransmitirá uma ação de controle $\gamma'_i = \gamma' \cap \Sigma_{c,i}$, $i \in I_n$, se ela for igual a última ação $\gamma_i = \gamma \cap \Sigma_{c,i}$ enviada para o local de atuação AS_i. Dessa forma, podem ocorrer casos em que o supervisor gera uma nova ação de controle e nada é transmitido pelos canais vulneráveis. Vale ressaltar que, em decorrência da inclusão do *Emissor*, a abordagem proposta em Yang et al. (2021b) não pode ser aplicada a estrutura de SEDRV considerada neste artigo, uma vez que, em Yang et al. (2021b), supõe-se que uma ação de controle será sempre transmitida, mesmo quando ela for igual a última ação transmitida.

Exemplo 1. Considere o sistema em malha fechada apresentado na Figura 2, o qual é formado por um SEDRV $\mathcal{G}$, um supervisor S e um dispositivo *Emissor*.

O SEDRV é a quíntupla $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$, sendo sua planta modelada por $G = (X_g, \Sigma, f_g, \Gamma_g, x_0, X_m)$, em que $\Sigma = \{c_1, c_2, c_3, c_4, b_1, b_2\}$, $\Sigma_o = \{c_4, b_1, c_1, c_2, c_3\}$ e $\Sigma_c = \{c_1, c_2, c_3, c_4\}$. As ocorrências de eventos são transmitidas por meio de dois canais de observação e as ações de controle são transmitidas por meio de três canais de controle, sendo os vetores de observação e controle definidos, de acordo com a Figura 2, como $C = [\{c_1, c_2\}, \{c_3\}, \{c_4\}]$ e $O = [\{c_3, c_4\}, \{b_1, c_1, c_2\}]$. Os canais vulneráveis são indicados em vermelho na Figura 2 e, portanto, $I_{c,v} = \{1, 2\}$ e $I_{o,v} = \{1\}$.

A realização $G_S = (X_S, \Sigma, f_S, \Gamma_S, x_{0_S}, X_S)$ do supervisor S é mostrada na parte superior da Figura 2. Por meio da composição paralela $G \| G_S$, pode-se observar que o supervisor S sempre impede que a planta G execute as transições de estado $f(4, c_4) = 1$, $f(2, c_2) = 4$ e $f(3, c_2) = 3$, sendo as demais transições de G permitidas de ocorrer.

O *Emissor* se comporta da forma descrita a seguir. Uma vez que $S(\varepsilon) = \{b_1, b_2, c_1\}$, então o *Emissor* envia, inicialmente, $\gamma_1 = \{c_1\}$, $\gamma_2 = \emptyset$ e $\gamma_3 = \emptyset$ para os locais de atuação AS_1, AS_2 e AS_3, respectivamente. A cada nova ação de controle gerada pelo supervisor, o *Emissor* define o que será enviado para cada local de atuação. Suponha que o primeiro evento a ocorrer na planta seja b_1, então, como $S(b_1) = \{b_1, b_2, c_1, c_4\}$, o *Emissor* envia

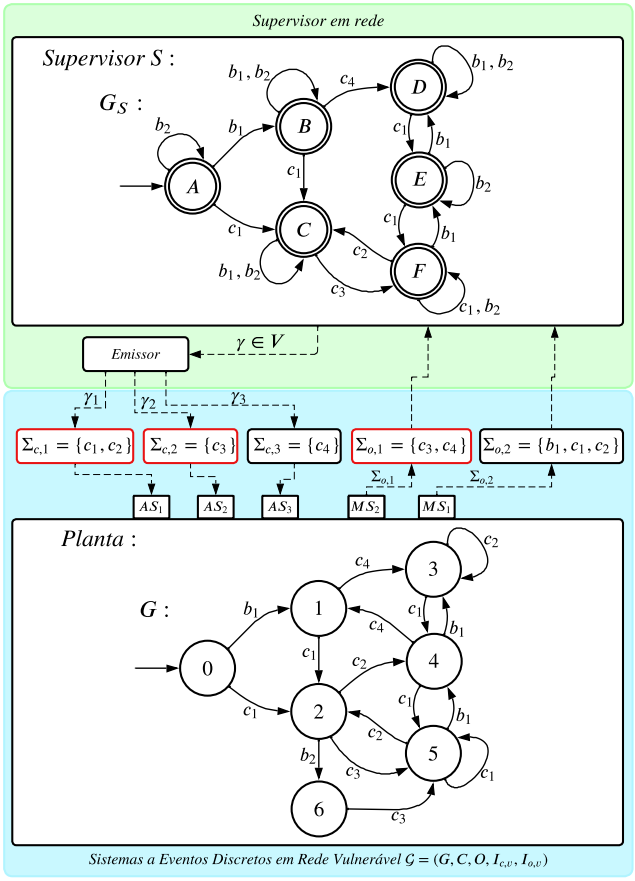

Figura 2. Exemplo de SEDRV sob a ação de um supervisor.

apenas $\gamma_3' = \{c_4\}$ para o local de atuação AS_3, pois as ações de controle em AS_1 e AS_2 devem permanecer as mesmas. Consequentemente, além do intruso não ser capaz de observar diretamente a ocorrência de b_1 na planta, pois ela é transmitida por um canal de observação seguro, ele também não é capaz de inferir a sua ocorrência a partir da transmissão da ação de controle $S(b_1)$, pois o único canal de controle utilizado na transmissão de $S(b_1)$ é o canal cch_3, o qual não é vulnerável.

Vale ressaltar que, ao se considerar a estrutura proposta em Yang et al. (2021b) após a ocorrência da sequência $s = b_1$, o supervisor transmitirá simultaneamente as ações de controle $S(b_1) \cap \Sigma_{c,i}$, $i = 1, 2, 3$, para seus respectivos locais de atuação AS_i, o que permitirá ao intruso inferir a ocorrência de b_1 na planta, ao identificar a transmissão de $S(b_1) \cap \Sigma_{c,1}$ e $S(b_1) \cap \Sigma_{c,2}$ pelos canais de controle vulneráveis cch_1 e cch_2, respectivamente. Esse fato ilustra a vantagem de se adicionar o dispositivo *Emissor* na arquitetura de controle supervisório em rede. □

4. MODELAGEM DA OBSERVAÇÃO DO SISTEMA PELO INTRUSO

4.1 Estimação de estado realizada pelo intruso

Conforme ilustrado na Figura 1, um intruso terá acesso a todos os canais vulneráveis och_j, $j \in I_{o,v}$, e cch_i, $i \in I_{c,v}$. Dessa forma, a estimação de estados do intruso não será feita apenas com base na observação de eventos da planta, sendo também melhorada com base na observação das ações de controle geradas pelo supervisor. Supõe-se que o intruso conhece a planta G do SEDRV e o supervisor S que atua sobre o sistema. Consequentemente, o intruso também possui conhecimento da linguagem do sistema em malha fechada $L(S/G)$ e do estado de G alcançado após a ocorrência de qualquer sequência em $L(S/G)$. Também se supõe que o intruso sabe da existência do dispositivo *Emissor*.

Seja o SEDRV $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$. O conjunto de eventos da planta que são observáveis do ponto de vista do intruso é dado por

$$\Sigma_{o,v} = \{\sigma \in \Sigma_o : (\exists j \in I_{o,v})[\sigma \in \Sigma_{o,j}]\}.$$

Define-se, então, a projeção $P_v : \Sigma^* \to \Sigma_{o,v}^*$ que modela a observação de ocorrências de eventos na planta pelo intruso.

Para modelar a observação de ações de controle transmitidas por canais vulneráveis, precisa-se levar em consideração o funcionamento do *Emissor*, o qual não retransmite uma ação de controle para um local de atuação AS_i, caso ela seja igual a última ação de controle enviada para AS_i. Dessa forma, a função que determina a ação de controle enviada por um canal de controle vulnerável cch_i, $i \in I_{c,v}$, após uma dada sequência de eventos observáveis é o mapeamento $T_i : \big(P_o(L(S/G)) \setminus \{\varepsilon\}\big) \to 2^{\Sigma_{c,i}} \cup \{\varepsilon\}$ definido como:

$$T_i(s\sigma) = \begin{cases} S(s\sigma) \cap \Sigma_{c,i}, \text{ se } S(s) \cap \Sigma_{c,i} \neq S(s\sigma) \cap \Sigma_{c,i}, \\ \varepsilon, \text{ caso contrário,} \end{cases}$$

para $s \in \Sigma_o^*$ e $\sigma \in \Sigma_o$.

Vale ressaltar que, $T_i(s\sigma) = \varepsilon$ indica que nenhuma nova ação de controle é transmitida para AS_i após o supervisor observar a sequência $s\sigma$; isso ocorre por consequência da ação controle anterior enviada para AS_i, $S(s) \cap \Sigma_{c,i}$, ser igual a ação de controle atual $S(s\sigma) \cap \Sigma_{c,i}$. Por outro lado, $T_i(s\sigma) = \emptyset$ indica que, após o supervisor observar $s\sigma$, uma nova ação de controle é enviada para AS_i desabilitando todos os eventos em $\Sigma_{c,i}$.

Com o objetivo de modelar a transmissão de informação em todos os canais de controle vulneráveis, define-se um vetor $v = [v_1, v_2, ..., v_{|I_{c,v}|}]$ de dimensão $1 \times |I_{c,v}|$, a partir do conjunto de índices dos canais de controle vulneráveis $I_{c,v}$, de tal forma que, $\forall i, j \in \{1, \ldots, |I_{c,v}|\}$, $v_i \in I_{c,v}$ e $v_i \neq v_j$ para $i \neq j$.

Com a intenção de unir a informação observada pelo intruso por meio dos canais de observação vulneráveis e dos canais de controle vulneráveis, será introduzida uma máscara de observação M que mapeia uma ocorrência de um evento observável $\sigma \in \Sigma_o$ em um símbolo, conforme descrito a seguir:

- Se $\sigma \in \Sigma_o \setminus \Sigma_{o,v}$ e, além disso, a ação de controle emitida por S em decorrência da observação de σ não gerar nenhuma nova transmissão pelos canais de controle vulneráveis, então σ será mapeado em ε;
- Se $\sigma \in \Sigma_{o,v}$ e, portanto, a sua ocorrência for transmitida por meio de um canal de observação vulnerável, porém a observação de σ pelo supervisor não gerar a transmissão de nenhuma ação de controle por canais vulneráveis, então σ será mapeado nele mesmo;
- Se uma nova ação de controle for transmitida por ao menos um dos canais de controle vulneráveis em

decorrência da observação de σ por S, então σ será mapeado em um par pertencente ao conjunto $\Delta_{oc} = \left((\Sigma_{o,v} \cup \{\varepsilon\}) \times \left(\bigtimes_{k=1}^{|I_{c,v}|} \left(2^{\Sigma_{c,v_k}} \cup \{\varepsilon\}\right)\right)\right)$, sendo o primeiro componente desse par igual a $P_v(\sigma)$, e o segundo componente igual a uma tupla formada pelas ações de controle transmitidas por cada um dos canais de controle vulneráveis.

Vale ressaltar que a máscara de observação M deve ser dinâmica em decorrência da natureza dinâmica do supervisor e do *Emissor*. A máscara $M : \Sigma_o^* \times \Sigma_o \to (\{\varepsilon\} \cup \Sigma_{o,v} \cup \Delta_{oc})$ é formalmente definida como:

$$M(s,\sigma) = \begin{cases} P_v(\sigma), \text{ se } (\forall k \in \{1,\ldots,|I_{c,v}|\}, T_{v_k}(s\sigma) = \varepsilon), \\ (P_v(\sigma), (T_{v_1}(s\sigma), \ldots, T_{v_{|I_{c,v}|}}(s\sigma)), \text{ se } (\exists k \in \\ \qquad \{1,\ldots,|I_{c,v}|\}, T_{v_k}(s\sigma) \neq \varepsilon), \end{cases}$$

se $s\sigma \in P_o(L(S/G))$, e não definida, caso contrário.

Note que $M(s,\sigma) = \varepsilon$ se o intruso não observar a ocorrência de σ, isto é, $P_v(\sigma) = \varepsilon$, e nada for transmitido pelos canais de controle vulneráveis, ou seja, o intruso não é capaz de observar nenhuma mudança no comportamento do sistema. Por outro lado, $M(s,\sigma) = \sigma$ se o intruso observar a ocorrência de σ na planta, porém nada for transmitido pelos canais de controle vulneráveis; dessa forma, o intruso consegue inferir, com base no conhecimento prévio da planta e do supervisor, que, além da ocorrência de σ na planta, a ação de controle emitida por S não gerou a transmissão de novas ações de controle pelos canais vulneráveis. Por fim, o caso em que $M(s,\sigma) \in \Delta_{oc}$ será ilustrado no exemplo a seguir.

Exemplo 2. Considere o sistema em malha fechada do Exemplo 1, o qual é mostrado na Figura 2, e o vetor $v = [1,2]$ formado a partir de $I_{c,v}$. Conforme descrito no Exemplo 1, como $S(\varepsilon) = \{c_1, b_1, b_2\}$, então o *Emissor* envia, inicialmente, $\gamma_1 = \{c_1\}$, $\gamma_2 = \emptyset$ e $\gamma_3 = \emptyset$ para os locais de atuação AS_1, AS_2 e AS_3, respectivamente.

Suponha que o primeiro evento a ocorrer na planta seja b_1. Então, uma vez que $b_1 \in \Sigma_o \setminus \Sigma_{o,v}$, o intruso não será capaz de observar diretamente a sua ocorrência. Além disso, $S(b_1) \cap \Sigma_{c,1} = \gamma_1$, $S(b_1) \cap \Sigma_{c,2} = \gamma_2$ e $S(b_1) \cap \Sigma_{c,3} = \{c_4\} \neq \gamma_3$ e, consequentemente, a transmissão da ação de controle $S(b_1)$ envolverá apenas o canal cch_3, o qual não é vulnerável. Isso é capturado pela função T_i, $i = 1,2$, isto é, $T_1(b_1) = T_2(b_1) = \varepsilon$, o que indica que nada é transmitido pelos canais vulneráveis cch_1 e cch_2 após a observação de b_1 pelo supervisor. Como consequência, a máscara M mapeia essa ocorrência de b_1 em $M(\varepsilon, b_1) = \varepsilon$, indicando que nada é observado pelo intruso em decorrência da transmissão de b_1 da planta para o supervisor e da transmissão de $S(b_1)$ do supervisor para planta.

Considere que, após a ocorrência de b_1, o evento $c_4 \in \Sigma_{o,v}$ ocorre na planta. Então, o *Emissor* enviará apenas $S(b_1c_4) \cap \Sigma_{c,3} = \emptyset$ pelo canal de controle seguro cch_3, desabilitando o evento controlável c_4 no local de atuação AS_3. As ações de controle aplicadas aos demais locais de atuação não são alteradas e, portanto, não são reenviadas. Consequentemente, $M(b_1, c_4) = c_4$ indicando que o intruso observa a ocorrência de c_4, porém não observa nenhuma alteração nas ações de controle em decorrência da observação de c_4 pelo supervisor.

Considere agora que, após a ocorrência de b_1, o evento $c_1 \in \Sigma_o \setminus \Sigma_{o,v}$ ocorre na planta ao invés de c_4. Nesse caso, o *Emissor* enviará $S(b_1c_1) \cap \Sigma_{c,1} = \emptyset$ e $S(b_1c_1) \cap \Sigma_{c,2} = \{c_3\}$ pelos canais de controle vulneráveis cch_1 e cch_2, respectivamente, desabilitando c_1 e c_2 em AS_1 e habilitando c_3 em AS_2. Consequentemente, $M(b_1, c_1) = (\varepsilon, (\emptyset, \{c_3\}))$ indicando que, embora o intruso não observe a ocorrência de c_1, ele é capaz de inferir que um evento pertencente a Σ_o ocorreu, gerando a transmissão das ações de controle $\emptyset$ e $\{c_3\}$ por cch_1 e cch_2.

Considere que, após a ocorrência de b_1c_1, o evento $c_3 \in \Sigma_{o,v}$ ocorre na planta. Nesse caso, tem-se $M(b_1c_1, c_3) = (c_3, (\{c_1, c_2\}, \emptyset))$ indicando que além de observar diretamente a ocorrência de c_3, o intruso observa que essa ocorrência fez com que o supervisor alterasse as ações de controle aplicadas aos locais de atuação AS_1 e AS_2 para $\{c_1, c_2\}$ e $\emptyset$, respectivamente.

Por fim, considere que, após a sequência $b_1c_1c_3$, o evento b_1 ocorre novamente na planta. Então, $M(b_1c_1c_3, b_1) = (\varepsilon, (\{c_1, \varepsilon\}))$, em que o segundo componente $(\{c_1\}, \varepsilon)$ indica que o intruso observa a transmissão da ação de controle $\{c_1\}$ pelo canal vulnerável cch_1, porém nada foi transmitido pelo canal vulnerável cch_2. □

Com base na máscara de observação dinâmica M, define-se uma função de observação do intruso, denotada por I_v, que mapeia uma sequência $s \in L(S/G)$ em uma sequência observável que representa a observação do intruso enriquecida pela informação das ações de controle transmitidas pelos canais vulneráveis.

Definição 2. A função de observação do intruso é o mapeamento $I_v : L(S/G) \to (\Sigma_{o,v} \cup \Delta_{oc})^*$, definido recursivamente como se segue:

(i) $I_v(\varepsilon) = \left(\varepsilon, (S(\varepsilon) \cap \Sigma_{c,v_1}, \ldots, S(\varepsilon) \cap \Sigma_{c,v_{|I_{c,v}|}})\right)$

(ii) Para todo $s\sigma \in L(S/G)$ com $s \in \Sigma^*$ e $\sigma \in \Sigma$,

$$I_v(s\sigma) = \begin{cases} I_v(s), \text{ se } \sigma \in \Sigma_{uo}, \\ I_v(s)M(P_o(s), \sigma), \text{ se } \sigma \in \Sigma_o. \end{cases}$$

O domínio de I_v é estendido para $2^{L(S/G)}$ da forma $I_v(L) = \bigcup_{w \in L} I_v(w)$. □

Exemplo 3. Considere novamente o sistema em malha fechada do Exemplo 1. Suponha que a sequência $w_1 = b_1c_1b_2c_3b_1 \in L(S/G)$ ocorreu na planta. Então, a sequência observada pelo intruso após a ocorrência de w_1 é $I_v(w_1) = I_v(b_1c_1b_2c_3b_1) = I_v(\varepsilon)M(\varepsilon, b_1)M(b_1, c_1)$ $M(b_1c_1, c_3)M(b_1c_1c_3, b_1)$. Note que, como $b_2 \in \Sigma_{uo}$, o fato desse evento ocorrer em w_1 não afeta a observação do intruso. Por sua vez $I_v(\varepsilon) = (\varepsilon, (\{c_1\}, \emptyset))$ e, conforme apresentado no Exemplo 2, $M(\varepsilon, b_1) = \varepsilon$, $M(b_1, c_1) = (\varepsilon, (\emptyset, \{c_3\}))$, $M(b_1c_1, c_3) = (c_3, (\{c_1, c_2\}, \emptyset))$ e $M(b_1c_1c_3, b_1) = (\varepsilon, (\{c_1\}, \{\varepsilon\})$. Portanto, $I_v(w_1) = (\varepsilon, (\{c_1\}, \emptyset))\,(\varepsilon, (\emptyset, \{c_3\}))(c_3, (\{c_1, c_2\}, \emptyset))\,(\varepsilon, (\{c_1\}, \{\varepsilon\}))$. □

Após observar uma sequência pertencente a $I_v(L(S/G))$, o intruso estima o estado que a planta se encontra, tendo como base as ocorrência de eventos e as ações de controle contidas nessa sequência. Formalmente, após $t \in I_v(L(S/G))$, a estimação de estados do ponto de vista do intruso é descrita por

$$SE(t) = \{x \in X : (\exists s \in L(S/G))[f(x_0, s) = x \wedge t = I_v(s)]\}.$$

4.2 Autômato observador do intruso

Nesta seção, será proposto um algoritmo (Algoritmo 1) para a construção de um autômato determinístico G_v que será utilizado para calcular a estimação do estado da planta do ponto de vista do intruso. Com esse objetivo, serão considerados um SEDRV $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$, em que $G = (X, \Sigma, f, \Gamma, x_0, X_m)$, e um supervisor S cuja realização é $G_S = (X_S, \Sigma, f_S, \Gamma_S, x_{0_S}, X_S)$. Além disso, será considerado um vetor $v = [v_1, v_2, ..., v_{|I_{c,v}|}]$ construído a partir dos índices dos canais de controle vulneráveis em $I_{c,v}$, conforme descrito na Seção 4.1.

No *Passo 1* do Algoritmo 1, constrói-se o autômato $G_{S/G}$ que modela o comportamento do sistema em malha fechada. A ideia por trás do Algoritmo 1 é, inicialmente, construir um autômato não-determinístico a partir $G_{S/G}$ que descreverá as observações do intruso decorrentes do disparo de cada uma das transições de $G_{S/G}$. Com isso em mente, no *Passo 2* do Algoritmo 1, o autômato não-determinístico G_{nd} é obtido modificando-se os rótulos de cada transição de $G_{S/G}$ por um evento que representa o que será observado pelo intruso se aquela transição ocorrer. Com esse objetivo, duas funções são definidas a seguir.

Definição 3. (funções aplicadas no Algoritmo 1).
(i) Para todo $k \in \{1, \ldots, |I_{c,v}|\}$, a função de envio de ações de controle é o mapeamento $\mathcal{T}_{v_k} : X_S \times \Sigma \to \left(2^{\Sigma_{c,v_k}} \cup \{\varepsilon\}\right)$ definido como:

$$\mathcal{T}_{v_k}(x_s, \sigma) = \begin{cases} \varepsilon, \text{ se } \Gamma_s(x_s) \cap \Sigma_{c,v_k} = \Gamma_s(f_s(x_s, \sigma)) \cap \Sigma_{c,v_k}, \\ \Gamma_s(f_s(x_s, \sigma)) \cap \Sigma_{c,v_k}, \text{ caso contrário.} \end{cases}$$

(ii) A função de renomeação de transições é o mapeamento $R : X_S \times \Sigma \to (\{\varepsilon\} \cup \Sigma_{o,v} \cup \Delta_{oc})$ definido como:

$$R(x_s, \sigma) = \begin{cases} P_v(\sigma), \text{ se } (\forall k \in \{1, \ldots, |I_{c,v}|\}, \mathcal{T}_{v_k}(x_s, \sigma) = \varepsilon), \\ (P_v(\sigma), (\mathcal{T}_{v_1}(x_s, \sigma), \ldots, \mathcal{T}_{v_{|I_{c,v}|}}(x_s, \sigma)), \text{ caso contrário.} \end{cases}$$

□

Note que, de acordo com a Definição 3, a função de envio de ações de controle $\mathcal{T}_{v_k}$ é semelhante a função T_i, $i \in v$, introduzida na Seção 4.1. A diferença entre essas funções concerne aos seus domínios, a função T_i é definida em termos de sequências pertencentes a $P_o(L(S/G))$ e, por sua vez, a função $\mathcal{T}_{v_k}$ é definida em função das transições do autômato G_S. A relação entre essas duas funções é estabelecida no lema a seguir.

Lema *4.* Para todo $s\sigma \in L(S/G)$ e $k \in \{1, \ldots, |I_{c,v}|\}$: *(i)* $\sigma \in \Sigma_o \Rightarrow \mathcal{T}_{v_k}(f_S(x_{0_S}, s), \sigma) = T_{v_k}(P_o(s\sigma))$, e *(ii)* $\sigma \in \Sigma_{uo} \Rightarrow \mathcal{T}_{v_k}(f_S(x_{0_S}, s), \sigma) = \varepsilon$. □

A função de renomeação de transições R, assim como $\mathcal{T}_{v_k}$, é definida em relação às transições do autômato G_S. A renomeação executada pela função R é feita de acordo com a máscara de observação dinâmica M, conforme mostrado no lema a seguir.

Lema *5.* Para todo $s\sigma \in L(S/G)$: *(i)* $\sigma \in \Sigma_o \Rightarrow R(f_S(x_{0_S}, s), \sigma) = M(P_o(s), \sigma)$, e *(ii)* $\sigma \in \Sigma_{uo} \Rightarrow R(f_S(x_{0_S}, s), \sigma) = \varepsilon$. □

Com base no Lema 5, pode-se inferir que $R(x_s, \sigma)$ descreve a observação resultante da ocorrência do evento σ enquanto G_S está no estado x_s, levando em consideração tanto a informação transmitida pelos canais de controle vulneráveis quanto a informação transmitida pelos canais de observação vulneráveis.

Algoritmo 1. (Construção do autômato G_v).

Entradas:
• SEDRV $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$ em que $G = (X, \Sigma, f, \Gamma, x_0, X_m)$
• Realização $G_S = (X_S, \Sigma, f_S, \Gamma_S, x_{0_S}, X_{m_S})$ do supervisor S

Saída: Autômato $G_v = (X_v, \Sigma_{o,v} \cup \Delta_{co}, f_v, x_{0_v}, \Gamma_v, X_{m_v})$ que modela S/G do ponto de vista do intruso

Passo 1: Construa o autômato $G_{S/G} := G \| G_S = (X_{S/G}, \Sigma, f_{S/G}, \Gamma_{S/G}, x_{0_{S/G}}, X_{m_{S/G}})$.

Passo 2: A partir de $G_{S/G}$, construa o autômato não-determinístico $G_{nd} = (X_{S/G}, \Sigma' \cup \{\varepsilon\}, f_{nd}, \Gamma_{nd}, \{x_{0_{S/G}}\}, X_{m_{S/G}})$, em que:

(i) $\Sigma' = \Sigma_{o,v} \cup \Delta_{oc}$,

(ii) Para todo $x_{nd} = (x, x_s) \in X_{S/G}$:

$$\Gamma_{nd}(x_{nd}) = \{R(x_s, \sigma) : \sigma \in \Gamma_{S/G}(x_{nd})\},$$

(iii) Para todo $x_{nd} = (x, x_s) \in X_{S/G}$ e $e \in \Gamma_{nd}(x_{nd})$:

$$f_{nd}(x_{nd}, e) = \{y \in X_{S/G} : (\exists \sigma \in \Sigma)[y = f_{S/G}(x_{nd}, \sigma) \wedge e = R(x_s, \sigma)]\}.$$

Passo 3: Baseado no autômato G_{nd}, construa o autômato não-determinístico $G'_{nd} = (X_{S/G} \cup \{init\}, \Sigma' \cup \{\varepsilon\}, f'_{nd}, \Gamma'_{nd}, \{init\}, X_{m_{S/G}})$, em que:

(i) Para todo $x_{nd} \in X_{S/G}$, $\Gamma'_{nd}(x_{nd}) = \Gamma_{nd}(x_{nd})$, e $\Gamma'_{nd}(init) = \{I_v(\varepsilon)\}$,

(ii) Para todo $x_{nd} \in X_{S/G}$ e $e \in \Gamma_{nd}(x_{nd})$, $f'_{nd}(x_{nd}, e) = f_{nd}(x_{nd}, e)$ e $f_{nd}(init, I_v(\varepsilon)) = \{x_{0_{S/G}}\}$.

Passo 4: Calcule o autômato $G_v = Obs(G'_{nd})$.

No *Passo 2* do Algoritmo 1, o autômato não-determinístico G_{nd} é construído a partir de $G_{S/G}$, usando a função de renomeação de transições R para renomear cada transição de $G_{S/G}$. Note que a estrutura de $G_{S/G}$ se conserva em G_{nd}, mas os eventos que rotulam as transições de estado são substituídos de acordo com a máscara de observação dinâmica M. No *Passo 3*, o autômato não-determinístico G'_{nd} é construído a partir de G_{nd} adicionando-se um novo estado a G_{nd}, denotado por *init*, o qual representa a inicialização do sistema, sendo a única transição a partir desse estado rotulada pela observação da ação de controle inicial $I_v(\varepsilon)$. Por fim, no *Passo 4*, calcula-se o autômato observador $G_v = Obs(G'_{nd})$.

Exemplo 4. Considere o sistema em malha fechada apresentado no Exemplo 1 (Figura 2). Com o objetivo de construir o autômato G_v que modela a observação do sistema do ponto de vista do intruso, o autômato $G_{S/G} = G \| G_S$, mostrado na Figura 3(a), é calculado no Passo 1 do Algoritmo 1. Em seguida, no Passo 2 do Algoritmo 1, a função de renomeação de transições R é usada para obter

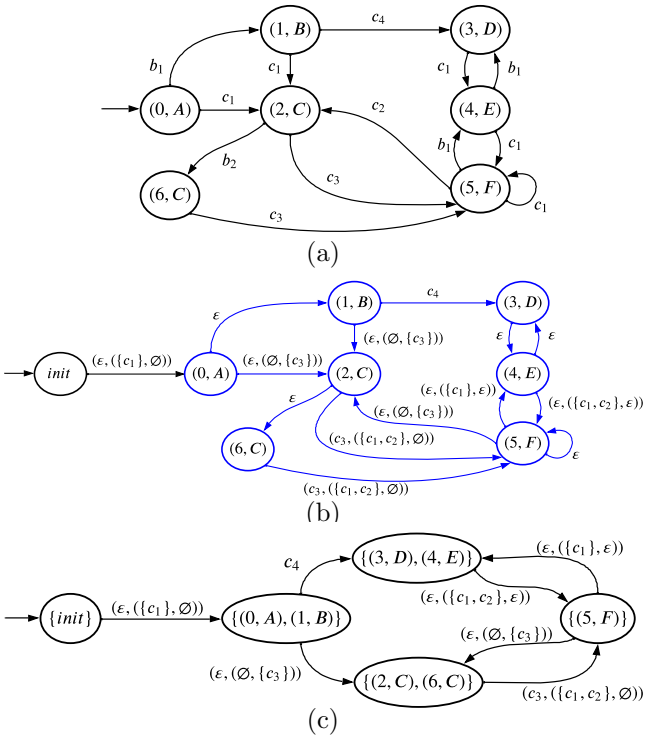

Figura 3. Autômato construídos no Exemplo 4: (a) $G_{S/G}$, (b) G'_{nd} e (c) G_v.

o autômato G_{nd} cujo diagrama de transição de estados corresponde a parte em azul da Figura 3(b). No Passo 3, o autômato G'_{nd} mostrado na Figura 3(b) é obtido adicionando-se o estado *init* a G_{nd} e criando-se uma transição de *init* para $(0, A)$, rotulada por $I_v(\varepsilon) = (\varepsilon, (\{c_1\}, \emptyset))$. O autômato $G_v = Obs(G'_{nd})$ obtido no Passo 4 do Algoritmo 1 é mostrado na Figura 3(c).

Note que o estado inicial do autômato G_v representa um período de inicialização do sistema, no qual a ação de controle $S(\varepsilon)$ é enviada para a planta e, portanto, a única transição possível a partir desse estado é rotulada pela observação da ação de controle inicial do ponto de vista do intruso $I_v(\varepsilon) = (\{c_1\}, \{\emptyset\})$. □

A linguagem gerada pelo autômato G_v é determinada pelo lema a seguir.

Lema *6.* $L(G_v) = I_v(L(S/G))$. □

Com o objetivo de determinar a estimação do estado da planta $SE(t)$ após a observação de uma sequência $t \in I_v(L(S/G))$ pelo intruso, introduz-se o mapeamento de estados $\hat{X} : X_v \to 2^X$ definido, para todo $x_v \in X_v$, como se segue:

$$\hat{X}(x_v) = \{x \in X : (\exists x_s \in X_s)[(x, x_s) \in x_v]\}.$$

Note que $\hat{X}(x_v)$ captura a primeira componente dos estados de $X_{S/G} \subseteq X \times X_S$ que aparecem em x_v. Pode-se, então, apresentar o resultado a seguir.

Teorema *7.* Para todo $t \in I_v(L(S/G))$,

$$SE(t) = \hat{X}(f_v(x_{0_v}, t)).$$

Exemplo 5. Aplicando-se o Teorema 7 em conjunto com o autômato G_v obtido no Exemplo 4, pode-se verificar que, após a observação da sequência $t_1 = (\varepsilon, (\{c_1\}, \emptyset))\, c_4$, a estimação do estado da planta feita pelo intruso será $SE(t_1) = \hat{X}(f(x_{0_v}, t_1)) = \hat{X}(\{(3, D), (4, E)\}) = \{3, 4\}$. A dúvida do intruso entre os estados 3 e 4 após a observação de t_1 é decorrente do fato do intruso não ser capaz de identificar a ocorrência do evento c_1 enquanto o supervisor está no estado D, pois $c_1 \notin \Sigma_{o,v}$ e o *Emissor* não envia novas ações de controle para a planta quando o evento c_1 ocorre enquanto o supervisor está no estado D.

Por sua vez, após a observação de $t_2 = t_1\,(\varepsilon, (\{c_1, c_2\}, \varepsilon)))$, a estimação do estado da planta será $SE(t_2) = \hat{X}(f(x_{0_v}, t_2)) = \hat{X}(\{(5, F)\}) = \{5\}$. Note que a diferença entre as sequências observáveis t_1 e t_2 é apenas o último evento em t_2, $(\varepsilon, (\{c_1, c_2\}, \varepsilon))$, o qual corresponde a ocorrência de c_1 no estado 4 da planta, que não é diretamente observada pelo intruso mas é identificada devido a transmissão da ação de controle $\{c_1, c_2\}$ pelo canal de controle vulnerável cch_1. Portanto, pode-se concluir que, nesse caso, a observação da ação de controle emitida pelo supervisor foi suficiente para o intruso atualizar a estimação do estado atual da planta de $\{3, 4\}$ para $\{5\}$. □

5. OPACIDADE EM REDE

Com o objetivo de garantir que um sistema em malhada fechada seja seguro em relação às informações contidas nele, o conceito de opacidade de estado atual será generalizado para SEDRV. Com isso em mente, a definição de opacidade de estado atual será adaptada para considerar as ações de controle emitidas pelo supervisor por meio de canais vulneráveis, utilizando a função de observação do intruso I_v. Esse novo conceito de opacidade será chamado opacidade (de estado atual) em rede.

Definição 8. (Opacidade em rede). Sejam $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$ um SEDRV, em que $G = (X, \Sigma, f, \Gamma, x_0, X_m)$, e S um supervisor projetado para controlar G. O sistema em malha fechada S/G é dito ser opaco em rede em relação aos canais vulneráveis determinados por $I_{c,v}$ e $I_{o,v}$ e a um conjunto de estados secretos $X_{sec} \subseteq X$ se

$$\big(\forall s \in L(S/G)\big)\big(f(x_0, s) \in X_{sec} \Rightarrow (\exists s' \in L(S/G), \\ f(x_0, s') \in X \setminus X_{sec} \wedge I_v(s') = I_v(s)\big).$$

Em palavras, S/G é opaco em rede em relação a $I_{c,v}$, $I_{o,v}$ e X_{sec} se, para toda sequência $s \in L(S/G)$ que conduz G para um estado secreto, existir uma sequência $s' \in L(S/G)$ que conduza G para um estado não-secreto e que seja indistinguível de s do ponto de vista do intruso.

O teorema a seguir apresenta uma condição necessária e suficiente para a opacidade em rede.

Teorema *9.* Sejam $\mathcal{G} = (G, C, O, I_{c,v}, I_{o,v})$ um SEDRV e S um supervisor. Considere o autômato G_v, obtido de acordo com o Algoritmo 1, e um dado conjunto de estados secretos X_{sec}. O sistema em malha fechada S/G será opaco em rede em relação a $I_{c,v}$, $I_{o,v}$ e X_{sec} se, e somente se, para todo estado $x_v \in X_v$ em G_v, $\hat{X}(x_v) \setminus X_{sec} \neq \emptyset$. □

De acordo com o Teorema 9, a opacidade em rede pode ser verificada por meio de uma busca nos estados de G_v.

Exemplo 6. Considere, novamente, o sistema em malha fechada do Exemplo 1 (Figura 2), em que $I_{c,v} = \{1, 2\}$ e $I_{o,v} = \{1\}$, e cujo o autômato G_v obtido no Exemplo 4 é mostrado na Figura 3(c).

Considere o conjunto de estados secretos $X_{sec} = \{2, 4\}$. Então, pode-se observar, por meio da Figura 3(c), que, para todo estado $x_v \in X_v$ em G_v, $\hat{X}(x_v) \setminus X_{sec} \neq \emptyset$. Portanto, de acordo com o Teorema 6, o sistema em malha fechada S/G é opaco em rede em relação a $I_{c,v} = \{1, 2\}$, $I_{o,v} = \{1\}$ e $X_{sec} = \{2, 4\}$. Vale ressaltar que caso fosse utilizada a arquitetura de controle supervisório considerada por Yang et al. (2021b), na qual não se tem um dispositivo *Emissor* que impeça a retransmissão de ações de controle repetidas, o sistema em malha fechada não seria opaco, pois o intruso seria capaz de identificar quando planta estivesse no estado 4.

Considere, agora, o conjunto de estados secretos $X_{sec} = \{5\}$. Pode-se observar, com ajuda da Figura 2, que G alcança o estado 5 após a execução da sequência $w = b_1c_4c_1c_1 \in L(S/G)$. Além disso, $I_v(w) = (\varepsilon, (\{c_1\}, \emptyset))$ $c_4\,(\varepsilon, (\{c_1, c_4\}, \varepsilon))$ que é igual a sequência observável t_2 considerada no Exemplo 5. A estimação de estados após a observação de t_2 é $SE(t_2) = \{5\}$, o que já permite concluir que o sistema não é opaco. Consequentemente, quando se verifica a condição apresentada no Teorema 6 para estado alcançado por G_v após t_2 obtêm-se que $\hat{X}(f_v(x_{0_v}, t_2)) \setminus X_{sec} = \hat{X}(\{(5, F)\}) \setminus \{5\} = \emptyset$. Portanto, de acordo com o Teorema 6, S/G não é opaco em rede em relação a $I_{c,v} = \{1, 2\}$, $I_{o,v} = \{1\}$ e $X_{sec} = \{5\}$. □

6. CONCLUSÃO

Neste artigo, foi proposta uma estrutura de controle supervisório em rede, na qual o dispositivo *Emissor* é utilizado para limitar a capacidade do intruso de observar as ações de controle emitidas pelo supervisor. Nesse contexto, foi proposta uma nova propriedade de opacidade em rede e, também, um método para verificá-la. Como etapa futura desse trabalho, pretende-se buscar um método que permita modificar o supervisor de um sistema não opaco em rede para que, se possível, o sistema se torne opaco em rede.

REFERÊNCIAS

Alves, M.V.S., Barcelos, R.J., Carvalho, L.K., e Basilio, J.C. (2022). Robust decentralized diagnosability of networked discrete event systems against DoS and deception attacks. *Nonlinear Analysis: Hybrid Systems*, 44, 101162.

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.

Alves, M.V.S. e Basilio, J.C. (2023). State estimation and detectability of networked discrete event systems with multi-channel communication networks. *IEEE Transactions on Automation Science and Engineering*. DOI: 10.1109/TASE.2023.3265846.

Badouel, E., Bednarczyk, M., Borzyszkowski, A., Caillaud, B., e Darondeau, P. (2007). Concurrent secrets. *Discrete Event Dyn Syst*, 17, 425–446.

Barcelos, R.J. e Basilio, J.C. (2021). Enforcing current-state opacity through shuffle and deletions of event observations. *Automatica*, 133.

Cao, L., Jiang, X., Zhao, Y., Wang, S., You, D., e Xu, X. (2020). A survey of network attacks on cyber-physical systems. *IEEE Access*, 8, 44219–44227.

Carvalho, L.K., Wu, Y., Kwong, R., e Lafortune, S. (2018). Detection and mitigation of classes of attacks in supervisory control systems. *Automatica*, 97, 121 – 133.

Cassandras, C.G. e Lafortune, S. (2008). *Introduction to discrete event systems*. Springer, 2ª edição.

Do, V.L., Fillatre, L., Nikiforov, I., e Willett, P. (2017). Security of scada systems against cyber–physical attacks. *IEEE Aerospace and Electronic Systems Magazine*, 32(5), 28–45.

Hadjicostis, C.N. (2020). *Estimation and inference in discrete event systems*. Springer, Switzerland, 1ª edição.

Jacob, R., Lesage, J.J., e Faure, J.M. (2016). Overview of discrete event systems opacity: Models, validation, and quantification. *Annual Reviews in Control*, 41, 135–146.

Ji, Y., Wu, Y.C., e Lafortune, S. (2018). Enforcement of opacity by public and private insertion functions. *Automatica*, 93, 369–378.

Keroglou, C. e Hadjicostis, C. (2018). Probabilistic system opacity in discrete event systems. *Discrete Event Dyn Syst*, 28, 289–314.

Kurose, J.F. e Ross, K.W. (2013). *Computer Networking: A Top-Down Approach*. Pearson, Upper Saddle River, NJ, USA, 6ª edição.

Lima, P.M., Alves, M.V.S., Carvalho, L.K., e Moreira, M.V. (2019). Security against communication network attacks of cyber-physical systems. *Journal of Control, Automation and Electrical Systems*, 30(1), 125–135.

Lima, P.M., Alves, M.V.S., Carvalho, L.K., e Moreira, M.V. (2022). Security of cyber-physical systems: Design of a security supervisor to thwart attacks. *IEEE Transactions on Automation Science and Engineering*, 19(3), 2030–2041.

Lima, P.M., Carvalho, L.K., e Moreira, M.V. (2023). Ensuring confidentiality of cyber-physical systems using event-based cryptography. *Information Sciences*, 621, 119–135.

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

Lin, F. (2011). Opacity of discrete event systems and its applications. *Automatica*, 47(3), 496–503.

Saboori, A. e Hadjicostis, C.N. (2007). Notions of security and opacity in discrete event systems. In *46th IEEE Conference on Decision and Control*, 5056–5061. New Orleans, LA, USA.

Wu, Y.C. e Lafortune, S. (2013). Comparative analysis of related notions of opacity in centralized and coordinated architectures. *Discrete Event Dynamic Systems*, 23(3), 307–339.

Wu, Y.C., Raman, V., Rawlings, B.C., Lafortune, S., e Seshia, S.A. (2018). Synthesis of obfuscation policies to ensure privacy and utility. *Journal of Automated Reasoning*, 60, 107–131.

Yang, J., Deng, W., Qiu, D., e Jiang, C. (2021a). Opacity of networked discrete event systems. *Information Sciences*, 543, 328–344.

Yang, S., Hou, J., Yin, X., e Li, S. (2021b). Opacity of networked supervisory control systems over insecure communication channels. *IEEE Transactions on Control of Network Systems*, 8(2), 884–896.

Zhu, Y., Lin, L., e Su, R. (2019). Supervisor obfuscation against actuator enablement attack. In *2019 18th European Control Conference (ECC)*, 1760–1765.