

**CONTROLE SUPERVISÓRIO DE SISTEMAS A EVENTOS DISCRETOS EM REDE
TEMPORIZADOS**

MARCOS V. S. ALVES*, LILIAN K. CARVALHO*, JOÃO C. BASILIO*

*COPPE - Programa de Engenharia Elétrica, Universidade Federal do Rio de Janeiro (UFRJ)
Rio de Janeiro-RJ, Brasil

Emails: mvalves@poli.ufrj.br, lilian@dee.ufrj.br, basilio@dee.ufrj.br

Abstract— In this paper, we study the supervisory control problem of Timed Networked Discrete Event Systems (TNDES) subject to delays and losses of observations. We assume that the plant is distributed, having, therefore, several communication channels for the transmission of the event observations to the supervisor; as a consequence communication delays may cause changes in the order of observations. Based on a model obtained by assuming *a priori* knowledge of the minimal transition activation time and the maximal time delay (possible packet loss also) of the communication channels, we formulate a networked supervisory control problem, and present a necessary and sufficient existence condition and a method for the synthesis of networked supervisors.

Keywords— Discrete event systems, supervisory control, automata.

Resumo— Neste artigo é tratado o problema de controle supervísório de sistemas a eventos discretos em rede temporizados (SEDRTs) sujeitos a atrasos e perdas de observações. Supõe-se que a planta é distribuída e, portanto, possui vários canais de comunicação para a transmissão das observações dos eventos. Consequentemente, atrasos de comunicação podem causar mudanças na ordem das observações recebidas pelo supervisor. Baseado em um modelo obtido supondo-se o conhecimento prévio dos tempos mínimos de ativação das transições da planta e dos atrasos máximos (e perdas) aos quais estão sujeitas as transmissões feitas por cada canal, formula-se um problema de controle supervísório em rede. Uma condição necessária e suficiente para a existência de supervisores em rede é apresentada e propõe-se, também, um método para projetar esses supervisores.

Palavras-chave— Sistemas a eventos discretos, controle supervísório, autômatos.

1 Introdução

Nos primeiros trabalhos em controle supervísório de Sistemas a Eventos Discretos (SEDs) (Ramadge and Wonham, 1989), supõe-se que o supervisor observa a ocorrência de um evento imediatamente após ele ser executado pela planta e que não ocorrem perdas de observação. Essas suposições são apropriadas somente se o SED estiver diretamente conectado ao supervisor, uma vez que, quando a planta e o supervisor estão distantes um do outro ou são interligados por meio de uma rede de comunicação complexa, atrasos e perdas de comunicação são inevitáveis e devem ser considerados. Esse problema é denominado Controle Supervísório de Sistemas em Rede (Lin, 2014; Shu and Lin, 2015; Shu and Lin, 2017).

Diversos trabalhos em SEDs abordam o problema de controle supervísório supondo atrasos na comunicação entre a planta e o supervisor (Balemi, 1994; Park and Cho, 2006; Park and Cho, 2007; Park and Cho, 2008; Lin, 2014; Shu and Lin, 2015; Shu and Lin, 2017). Todos esses trabalhos tem em comum a hipótese de que não ocorrem alterações na ordem das observações recebidas pelo supervisor, o que é equivalente a supor que todas as observações são transmitidas por meio de um único canal de comunicação modelado por uma fila *FIFO* (do inglês *first-in first-out*). No entanto, um único canal de comunicação normalmente é insuficiente, em decorrência da grande quantidade de dados transmitidos, ou do fato de

que, em sistemas distribuídos, sensores e atuadores estão geralmente distantes uns dos outros.

O problema de controle supervísório de SEDs supondo perdas intermitentes de observação foi considerado em Lin (2014) e Alves et al. (2014).

Neste artigo, são considerados Sistemas a Eventos Discretos em Rede Temporizados (SEDRTs) sujeitos a atrasos limitados de comunicação e perdas (intermitentes) de observações. Supõe-se que a comunicação entre a planta e o supervisor é feita por meio de uma rede que pode conter vários canais de comunicação e, portanto, atrasos na transmissão podem modificar a ordem das observações. Com o objetivo de propor um modelo não temporizado que represente todas as possíveis consequências decorrentes de atrasos e perdas de observações, supõe-se o conhecimento prévio dos tempos mínimos de ativação das transições da planta, dos atrasos máximos aos quais estão sujeitas as transmissões feitas por cada canal e do conjunto de eventos sujeitos a perdas de observações. Com base nesse novo modelo, um problema de controle supervísório em rede é proposto e novas definições de controlabilidade e observabilidade em rede são apresentadas e aplicadas para propor uma condição necessária e suficiente para a existência de um supervisor em rede.

2 Fundamentos teóricos

No decorrer deste artigo, $G = (X, \Sigma, f, \Gamma, x_0)$ denota um autômato de estados finitos, no qual,

X é o conjunto de estados, Σ é o conjunto de eventos, $f : X \times \Sigma^* \rightarrow X$ é a função de transição parcialmente definida sobre o seu domínio, em que Σ^* denota o fecho de Kleene de Σ , Γ é o conjunto de eventos ativos, ou seja, $\Gamma(x) = \{\sigma \in \Sigma : f(x, \sigma) \text{ é definida}\}$, e x_0 é o estado inicial. A linguagem gerada por G é definida como $L(G) = \{s \in \Sigma^* : f(x_0, s) \text{ é definida}\}$. O fecho do prefixo de uma linguagem L é definido como $\bar{L} = \{s \in \Sigma^* : (\exists t \in \Sigma^*)[st \in L]\}$, e L é dita ser prefixo-fechada se $L = \bar{L}$. A notação $|B|$ é usada para se referir à cardinalidade de um conjunto B .

O conjunto de eventos pode ser particionado como $\Sigma = \Sigma_c \dot{\cup} \Sigma_{uc}$, em que Σ_c e Σ_{uc} são, respectivamente, os conjuntos de eventos controláveis e não controláveis. Uma linguagem $K \subseteq L(G)$ é dita ser controlável em relação a $L(G)$ e Σ_{uc} se $\bar{K} \Sigma_{uc} \cap L(G) \subseteq \bar{K}$.

O conjunto de eventos também pode ser particionado como $\Sigma = \Sigma_o \dot{\cup} \Sigma_{uo}$, em que Σ_o e Σ_{uo} são, respectivamente, os conjuntos de eventos observáveis e não observáveis. A projeção natural é a função $P_o : \Sigma^* \rightarrow \Sigma_o^*$, definida da seguinte forma: (i) $P_o(\varepsilon) = \varepsilon$, em que ε denota a sequência vazia; (ii) $P_o(\sigma) = \sigma$, se $\sigma \in \Sigma_o$; (iii) $P_o(\sigma) = \varepsilon$, se $\sigma \in \Sigma_{uo}$; (iv) $P_o(s\sigma) = P_o(s)P_o(\sigma)$, para $s \in \Sigma^*$ e $\sigma \in \Sigma$. A projeção inversa $P_o^{-1} : \Sigma_o^* \rightarrow \Sigma^*$ é definida como $P_o^{-1}(t) = \{s \in \Sigma^* : P_o(s) = t\}$. As projeções natural e inversa são ambas estendidas para linguagens aplicando-as a todas as sequências pertencente à linguagem. Uma linguagem $K \subseteq L(G)$ é dita ser observável em relação a $L(G)$ e P_o se, para todo $s \in \bar{K}$ e $\sigma \in \Sigma$, $s\sigma \in L(G) \setminus \bar{K}$ implica que $P_o^{-1}(P_o(s))\{\sigma\} \cap \bar{K} = \emptyset$.

Seja uma planta modelada por um autômato G . Um supervisor S atua com o objetivo de restringir o comportamento do sistema em malha fechada, denotado por S/G , a uma especificação dada em termos de uma linguagem prefixo-fechada $K \subset L(G)$, de tal forma que $L(S/G) = K$.

Sob observação parcial, isto é, $\Sigma_{uo} \neq \emptyset$, quando a planta gera uma sequência s , o supervisor decide quais eventos permanecem habilitados com base em $P_o(s)$ ao invés de s e, portanto, ele é definido como $S : P_o(L(G)) \rightarrow 2^\Sigma$, em que, para uma dada sequência $s \in L(G)$, $\Gamma[f(x_0, s)] \cap (S(P_o(s)) \cup \Sigma_{uc})$ são os eventos habilitados no estado de S/G correspondente ao estado $f(x_0, s)$ de G . A linguagem gerada pelo sistema em malha fechada é, portanto, definida recursivamente como se segue: (i) $\varepsilon \in L(S/G)$, e (ii) $s\sigma \in L(S/G) \Leftrightarrow s \in L(S/G) \wedge s\sigma \in L(G) \wedge \sigma \in (S(P_o(s)) \cup \Sigma_{uc})$.

Sejam G_1 e G_2 autômatos definidos sobre os conjuntos de eventos Σ_1 e Σ_2 , respectivamente. A composição paralela de G_1 e G_2 (Cassandras and Lafortune, 2008, p. 79) é denotada por $G_1 \parallel G_2$ e $L(G_1 \parallel G_2) = P_1^{-1}[L(G_1)] \cap P_2^{-1}[L(G_2)]$, em que $P_i : (\Sigma_1 \cup \Sigma_2)^* \rightarrow \Sigma_i^*$, para $i = 1, 2$.

3 Sistemas a eventos discretos em rede temporizados

3.1 Arquitetura de SEDRTs

Os sistemas considerados nesse trabalho possuem a estrutura ilustrada na figura 1, na qual a planta, modelada por um autômato $G = (X, \Sigma, f, \Gamma, x_0)$, se comunica com o controle supervisor por meio de uma rede de comunicação. As ocorrências de eventos observáveis são comunicadas ao supervisor por meio de m canais de comunicação (referidos aqui por canais de observação), e as ações de controle do supervisor são transmitidas para a planta por meio de n canais de comunicação (referidos aqui por canais de controle).

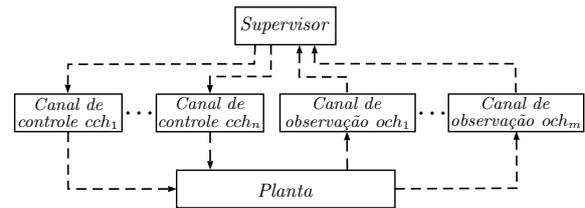


Figura 1: Arquitetura de controle supervisorio em rede.

Supõem-se as seguintes hipóteses:

A1. O conjunto de eventos observáveis é particionado como $\Sigma_o = \dot{\cup}_{i=1}^m \Sigma_{o,i}$, em que $\Sigma_{o,i}$, $i = 1, \dots, m$, é o conjunto de eventos cujas observações são transmitidas por meio do canal de observação och_i ;

A2. Cada canal de observação och_i , $i = 1, \dots, m$, é modelado por uma fila *FIFO* e está sujeito a atrasos de comunicação, cujo limite máximo é conhecido *a priori* e igual a T_i , em que T_i é um número real positivo diferente de zero;

A3. O conjunto de eventos observáveis é particionado, de acordo com perdas de observações, como $\Sigma_o = \Sigma_{lo} \dot{\cup} \Sigma_{nlo}$, em que Σ_{lo} (resp. Σ_{nlo}) é o conjunto de eventos sujeitos (resp. não sujeitos) a perdas de observações;

A4. O conjunto de eventos controláveis é particionado como $\Sigma_c = \dot{\cup}_{j=1}^n \Sigma_{c,j}$, em que $\Sigma_{c,j}$, $j \in \{1, \dots, n\}$, é o conjunto de eventos cuja ação de controle é transmitida por meio do canal de controle cch_j , o qual não está sujeito a perdas e atrasos de transmissão.

Seja \mathbb{R}_+ o conjunto de números reais não negativos. Com o objetivo de considerar os efeitos da dinâmica de plantas reais, define-se a função parcial $t_{min} : X \times \Sigma \rightarrow \mathbb{R}_+$ para todo $(x, \sigma) \in X \times \Sigma$ tal que $\sigma \in \Gamma(x)$, a qual representa o tempo mínimo de ativação do evento σ no estado x , ou seja, σ pode ocorrer no estado x somente se o tempo decorrido desde a última transição for maior que $t_{min}(x, \sigma)$. A seguinte hipótese é imposta à dinâmica da planta:

A5. $t_{min}(x, \sigma) > 0$ para todo $(x, \sigma) \in X \times \Sigma$ tal que $\sigma \in \Gamma(x)$.

3.2 Exemplo de motivação

Considere o SEDRT cuja rede de comunicação é apresentada na figura 2(a). A planta é modelada pelo autômato $G = (X, \Sigma, f, \Gamma, x_0)$ ilustrado na figura 2(b), no qual $\Sigma = \{\alpha, \beta, \gamma, \mu, \eta\}$, $\Sigma_o = \{\alpha, \beta, \gamma\}$, $\Sigma_c = \{\alpha, \gamma, \mu, \eta\}$ e $\Sigma_{lo} = \emptyset$. Na figura 2(b), o valor anexado a cada transição $(x, \sigma, f(x, \sigma))$ de G representa o tempo mínimo de ativação $t_{min}(x, \sigma)$.

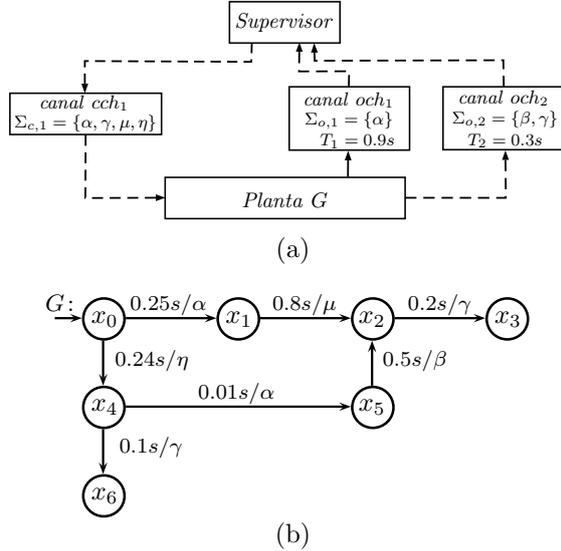


Figura 2: Sistema a eventos discretos temporizado em rede: (a) rede de comunicação e (b) autômato G que modela a planta.

Suponha que a planta G gerou a sequência $s_1 = \eta\alpha\beta\gamma$, na qual η é o único evento não observável. As ocorrências do evento α são transmitidas por meio do canal och_1 , cujo atraso de comunicação máximo é $T_1 = 0.9s$. Por sua vez, as ocorrências dos eventos β e γ são transmitidas por meio do canal och_2 , o qual está sujeito a um atraso máximo de comunicação igual a $T_2 = 0.3s$. Consequentemente, o evento α pode ser observado pelo supervisor das formas descritas a seguir: (i) antes da ocorrência de γ , quando o atraso de observação de α é menor que o tempo decorrido entre as ocorrências de α e γ ; (ii) entre a ocorrência e a observação do evento γ , dado que a observação de α pode atrasar até $0.9s$ e a planta pode gerar $\beta\gamma$ no mínimo $0.7s$ após a ocorrência do evento α ; (iii) após a observação do evento γ , dado que, como mostrado anteriormente, o evento α pode ser observado após a ocorrência de γ e as ocorrências de α e γ são transmitidas por canais de comunicação diferentes. O supervisor também pode observar o evento β após a ocorrência do evento γ , uma vez que $T_2 = 0.3s$ e o evento γ ocorre no mínimo $0.2s$ após a execução do evento β . No entanto, o supervisor não pode observar γ antes de observar β , uma vez que ambos são transmitidos por meio do mesmo canal, o qual, de acordo com a hipótese **A2**, é modelado por uma fila *FIFO*.

Suponha agora que a planta G gerou a sequên-

cia $s_2 = \alpha\mu\gamma$. Nesse caso, embora o evento α possa ser observado após a ocorrência do evento μ (o qual não é observável), o evento α não pode ser observado após a ocorrência de γ , uma vez que a planta só pode gerar a sequência $\mu\gamma$ no mínimo $1s$ após a execução do evento α , cuja observação atrasa no máximo $0.9s$.

Portanto, embora ambas sequências $s_1 = \eta\alpha\beta\gamma$ e $s_2 = \alpha\mu\gamma$ tenham um evento entre α e γ , o evento α pode ser observado pelo supervisor após a ocorrência (ou observação) de γ quando a planta gera a sequência s_1 , mas ele não pode ser observado após a ocorrência (ou observação) de γ quando a planta gera a sequência s_2 .

Com base nesse exemplo, pode-se concluir que, quando vários canais de comunicação são usados para transmitir as ocorrências dos eventos para o supervisor, possíveis mudanças na ordem das observações recebidas devem ser consideradas na etapa de projeto do supervisor, o que não é feito nos trabalhos anteriores (Balemi, 1994; Park and Cho, 2006; Park and Cho, 2007; Park and Cho, 2008; Lin, 2014; Shu and Lin, 2015; Shu and Lin, 2017).

Sistemas a eventos discretos em rede sujeitos a mudanças na ordem das observações são estudados por Nunes et al. (2016) no contexto de diagnosticabilidade de SEDs. Nesse trabalho, propõe-se um modelo baseado no conceito de *step*, ou seja, um atraso de comunicação limitado por n *steps* representa o caso no qual n eventos podem ser gerados pela planta, enquanto uma informação está sendo transmitida. No entanto, o conceito de *step* não pode ser usado para modelar corretamente o sistema considerado nesse exemplo. Essa limitação é decorrente do fato de que a sequência $s_1 = \eta\alpha\beta\gamma$ requer dois *steps* para modelar todos os possíveis atrasos de observação do evento α , enquanto a sequência $s_2 = \alpha\mu\gamma$ requer a suposição de que a observação de α está sujeita a atrasos de somente um *step*.

Na próxima seção, propõe-se um novo modelo para SEDRTs que permite representar sistemas distribuídos, com comportamento dinâmico heterogêneo e uma rede de comunicação composta por vários canais. Esse novo modelo é baseado no modelo proposto anteriormente em Nunes et al. (2016). No entanto, as informações de tempo da dinâmica da planta e da rede de comunicação são diretamente consideradas na modelagem, sem a utilização do conceito de *step*. A modelagem de perdas de observação é similar a abordagem utilizada em Carvalho et al. (2012) e Alves et al. (2014).

4 Um novo modelo para SEDRTs

A dinâmica de um SEDRT será descrita por um autômato $G_e = (X_e, \Sigma_e, f_e, \Gamma_e, x_{0_e})$ cuja linguagem gerada é definida sob o conjunto estendido

de eventos

$$\Sigma_e := \Sigma \cup \Sigma_o^s \cup \Sigma_o^l, \quad (1)$$

no qual os conjuntos de eventos adicionados Σ_o^s e Σ_o^l são utilizados para modelar, respectivamente, o sucesso e a perda da observação de um evento em Σ_o , sendo definidos da seguinte forma:

$$\Sigma_o^s := \{\sigma_s : \sigma \in \Sigma_o\} \text{ e } \Sigma_o^l := \{\sigma_l : \sigma \in \Sigma_{lo}\}. \quad (2)$$

Como mostrado no exemplo de motivação apresentado na seção 3.2, uma dada sequência executada pela planta pode gerar um conjunto de diferentes observações. Como consequência desse fato, toda sequência $s \in L(G)$ pode ser representada por um conjunto de sequências estendidas pertencentes a Σ_e^* , definido de acordo com $\Sigma_{o,i}$ e T_i , para todo $i \in \{1, \dots, m\}$, t_{min} e Σ_{lo} , como ilustrado no exemplo a seguir.

Exemplo 1 Considere o SEDRT apresentado nas figuras 2(a,b), no qual $\Sigma = \{\alpha, \beta, \gamma, \mu, \eta\}$ e $\Sigma_o = \{\alpha, \beta, \gamma\}$. Considere também que o evento β está sujeito a perdas de observações, ou seja, $\Sigma_{lo} = \{\beta\}$. Suponha que a planta execute a sequência $s = \eta\alpha\beta$. Então, em decorrência de atrasos e perdas de observações, as sequências estendidas a seguir podem ocorrer:

- A sequência $s_{e,1} = \eta\alpha\alpha_s\beta\beta_s$ que modela o caso em que o atraso da observação de α é menor que o intervalo de tempo entre a ocorrência de α e β (α é observado durante o intervalo 1 da figura 3);
- A sequência $s_{e,2} = \eta\alpha\beta\alpha_s\beta_s$ que modela o caso no qual o supervisor observa α entre a ocorrência e a observação de β (durante o intervalo 2 da figura 3);
- A sequência $s_{e,3} = \eta\alpha\beta\beta_s\alpha_s$ que modela o caso em que o supervisor observa β e em seguida observa α (durante o intervalo 3 da figura 3);
- As sequências $\eta\alpha\alpha_s\beta\beta_l$, $\eta\alpha\beta\alpha_s\beta_l$ e $\eta\alpha\beta\beta_l\alpha_s$ que modelam os casos em que a observação de β é perdida. Note que essas sequências são obtidas a partir das sequências anteriores por meio da substituição de β_s por β_l ;
- As sequências $\eta\alpha\alpha_s\beta$, $\eta\alpha\beta$, $\eta\alpha\beta\alpha_s$, $\eta\alpha\beta\beta_s$ e $\eta\alpha\beta\beta_l$ modelam casos nos quais algumas ocorrências ainda estão sendo transmitidas para o supervisor e, portanto, são prefixos das outras sequências citadas acima.

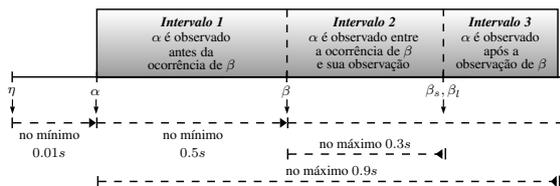


Figura 3: Possíveis observações de $s = \eta\alpha\beta$ modeladas por sequências estendidas em Σ_e^* .

Com o objetivo de associar sequências geradas pela planta G às sequências estendidas em G_e , as seguintes funções e projeções são definidas:

- $\psi : \Sigma_o^* \rightarrow \Sigma_o^{s*}$, em que, $\psi(\varepsilon) = \varepsilon$, $\psi(\sigma) = \sigma_s$, para todo $\sigma \in \Sigma_o$, e $\psi(s\sigma) = \psi(s)\psi(\sigma)$, para toda sequência $s \in \Sigma_o^*$ e $\sigma \in \Sigma_o$. A função ψ é estendida para linguagens aplicando-a a todas as sequências pertencentes à linguagem.
- $\psi^{-1} : \Sigma_o^{s*} \rightarrow \Sigma_o^*$, a qual é a função inversa de ψ .
- Projeções $P_e : \Sigma_e^* \rightarrow \Sigma^*$ e $P_{e,s} : \Sigma_e^* \rightarrow \Sigma_o^{s*}$.

Uma sequência estendida $s_e \in \Sigma_e^*$, como pode ser visto por meio das sequências apresentadas no exemplo 1, modela o caso em que a planta gera a sequência $P_e(s_e)$ e o supervisor observa a sequência $\psi^{-1}(P_{e,s}(s_e))$.

Com o objetivo de construir o autômato G_e iterativamente, os estados desse autômato são nomeados com duas componentes. A primeira componente de um estado de G_e é definida de acordo com o estado x de G correspondente. A segunda componente armazena duas informações: os eventos observáveis que foram gerados por G até o estado x ter sido alcançado e cujas ocorrências ainda estão sendo transmitidas, e o tempo mínimo decorrido entre a ocorrência desses eventos. Por exemplo, um estado de G_e nomeado como $(x_3, \alpha 0.5\beta 0.2\gamma)$ representa o caso em que a planta alcançou o estado x_3 após executar uma sequência $s \in L(G)$ que contém, nesta ordem, os eventos observáveis α , β e γ , cujas observações estão sendo transmitidas e, adicionalmente, o tempo decorrido entre as ocorrências de α e β (resp. β e γ) foi, no mínimo, igual a 0.5 (resp. 0.2) unidades de tempo. Vale ressaltar que a sequência s pode ter outros eventos observáveis cujas transmissões foram finalizadas antes do estado $(x_3, \alpha 0.5\beta 0.2\gamma)$ ser alcançado.

A definição da segunda componente de um estado de G_e é feita por meio da manipulação de eventos observáveis e números reais não-negativos. Portanto, define-se o conjunto $\mathcal{Q} := \{q = q_1q_2 \dots q_l : \forall i \in \{1, 2, \dots, l\}, (q_i \in \Sigma_o) \vee (q_i \in \mathbb{R}_+)\}$. A seguir, algumas operações sobre \mathcal{Q} são apresentadas, juntamente com uma função que associa um evento observável com o canal de observação utilizado para transmiti-lo.

Definição 1

◦ A operação de concatenação é a função $\text{conc} : \mathcal{Q} \times \mathcal{Q} \rightarrow \mathcal{Q}$ em que, para quaisquer sequências $q = q_1 \dots q_l \in \mathcal{Q}$ e $p = p_1 \dots p_k \in \mathcal{Q}$,

$$\text{conc}(q, p) = \begin{cases} q_1 \dots q_{l-1} (q_l + p_1) p_2 \dots p_k, & \text{se} \\ & q_l, p_1 \in \mathbb{R}_+ \\ q_1 \dots q_l p_1 \dots p_k, & \text{caso contrário.} \end{cases}$$

◦ A operação de eliminação é a função $\text{cut} : \mathcal{Q} \rightarrow \mathcal{Q}$ em que, para todo $q = q_1q_2 \dots q_l \in \mathcal{Q}$,

$$\text{cut}(q) = \begin{cases} q_i q_{i+1} \dots q_l, & \text{se } (\exists i \leq l)[(q_i \in \Sigma_o) \wedge \\ & \wedge (q_j \in \mathbb{R}_+, \forall j \in \{1, \dots, i-1\})] \\ 0, & \text{se } q_i \in \mathbb{R}_+, \forall i \in \{1, 2, \dots, l\}. \end{cases}$$

◦ A operação de adição é o mapeamento $add : \mathcal{Q} \times X \times \Sigma \rightarrow \mathcal{Q}$ em que, para todo $q = q_1 q_2 \cdots q_l \in \mathcal{Q}$, $x \in X$ e $\sigma \in \Sigma$,

$$add(q, x, \sigma) = \begin{cases} cut(conc(q, t_{min}(x, \sigma)\sigma)), & \text{se} \\ & (\sigma \in \Sigma_o) \wedge (\sigma \in \Gamma(x)) \\ cut(conc(q, t_{min}(x, \sigma))), & \text{se} \\ & (\sigma \in \Sigma_{uo}) \wedge (\sigma \in \Gamma(x)) \\ \text{indefinido, caso contrário.} \end{cases}$$

◦ A operação de remoção é o mapeamento $rem : \mathcal{Q} \times \mathbb{N} \rightarrow \mathcal{Q}$ em que, para todo $q = q_1 q_2 \cdots q_l \in \mathcal{Q}$,

$$rem(q, i) = \begin{cases} cut(q_2 \cdots q_l), & \text{se } (i = 1) \\ conc(q_1 \cdots q_{i-1}, q_{i+1} \cdots q_l), & \text{se} \\ & (1 < i < l) \\ cut(q_1 \cdots q_{l-1}), & \text{se } (i = l) \\ \text{indefinido, caso contrário.} \end{cases}$$

◦ A função de índice do canal de observação $och : \Sigma_o \rightarrow \{1, \dots, m\}$ é definida como $och(\sigma) = i$, para todo $\sigma \in \Sigma_{o,i}$ e $i \in \{1, \dots, n\}$.

O algoritmo usado para calcular G_e é apresentado a seguir. O exemplo 2 pode ser utilizado para facilitar o entendimento desse algoritmo.

Algoritmo 1 (Construção do autômato G_e)

Entradas:

- $G = (X, \Sigma, f, \Gamma, x_0)$: autômato que modela a planta do SEDRT;
- $\Sigma_{o,i}$ e T_i , para todo $i \in \{1, \dots, m\}$: conjunto de eventos transmitidos e atraso máximo referentes a cada canal de observação;
- Σ_{lo} : conjunto de eventos sujeitos a perdas de observações;
- $t_{min}(x, \sigma)$, para todo $(x, \sigma) \in X \times \Sigma$ tal que $\sigma \in \Gamma(x)$.

Saída:

- $G_e = (X_e, \Sigma_e, f_e, \Gamma_e, x_{0_e})$.

Passo 1: Defina o estado inicial $x_{0_e} = (x_0, 0)$ e $X_e = \emptyset$;

Passo 2: Defina Σ_e , Σ_o^s e Σ_o^l de acordo com as equações (1) e (2);

Passo 3: Construa uma fila FIFO $F = [x_{0_e}]$;

Passo 4: Enquanto $F \neq \emptyset$ faça:

- 4.1: Defina $(x, q) = dequeue(F)$.
- 4.2: $X_e \leftarrow X_e \cup \{(x, q)\}$;
- 4.3: Seja $q = q_1 q_2 \cdots q_l$. Defina o conjunto de índices $I_o = \{j \in \{1, \dots, l\} : q_j \in \Sigma_o\}$;
- 4.4: Para todo $y \in I_o$, defina $min_{et}(y)$ como a soma dos números reais a direita de q_y em q ;
- 4.5: Para todo $\sigma \in \Gamma(x)$:
 - (a) $\tilde{x}_e = f_e((x, q), \sigma) =$

$$\begin{cases} (f(x, \sigma), add(q, x, \sigma)), & \text{se } (min_{et}(y) + \\ & + t_{min}(x, \sigma)) < T_{och(q_y)}, \forall y \in I_o; \\ \text{indefinido, caso contrário.} \end{cases}$$

(b) Se $(\tilde{x}_e \notin X_e) \wedge (\tilde{x}_e \notin F)$, então faça $F = enqueue(F, \tilde{x}_e)$;

◦ 4.6: Para cada canal de observação och_i , $i = 1, \dots, m$:

(a) $y = min\{i \in I_o : \sigma_i \in \Sigma_{o,i}\}$;

(b) $\hat{x}_e = f_e((x, q), \psi(q_y)) =$

$$\begin{cases} (x, rem(q, y)), & \text{se } y \text{ for definido;} \\ \text{indefinido, caso contrário.} \end{cases}$$

(c) Se $q_y \in \Sigma_{lo}$, defina $f_e((x, q), q_{y_i}) = \hat{x}_e$;

(d) Se $(\hat{x}_e \notin X_e) \wedge (\hat{x}_e \notin F)$, então faça $F = enqueue(F, \hat{x}_e)$;

Passo 5: Defina $\Gamma_e(x_e) = \{\sigma \in \Sigma_e : f_e(x_e, \sigma) \text{ é definido}\}$, para todo $x_e \in X_e$.

Exemplo 2 Considere o SEDRT ilustrado nas figuras 2(a,b), em que $\Sigma = \{\alpha, \beta, \gamma, \mu, \eta\}$, $\Sigma_o = \{\alpha, \beta, \gamma\}$ e $\Sigma_{lo} = \{\beta\}$. Aplicando-se o algoritmo 1 com as entradas G , $\Sigma_{o,1} = \{\alpha\}$, $\Sigma_{o,2} = \{\beta, \gamma\}$, $T_1 = 0.9s$, $T_2 = 0.3s$, $\Sigma_{lo} = \{\beta\}$ e t_{min} definida de acordo com a figura 2(b), obtém-se o autômato G_e , apresentado na figura 4. Alguns passos da execução do algoritmo 1 são descritos a seguir.

No passo 1, define-se $x_{0_e} = (x_0, 0)$, no qual a segunda componente igual a zero implica que nenhuma ocorrência de evento está sendo transmitida para o supervisor enquanto o sistema permanecer nesse estado. No passo 2, os conjuntos de eventos $\Sigma_e = \{\alpha, \beta, \gamma, \mu, \eta, \alpha_s, \beta_s, \gamma_s, \beta_l\}$, $\Sigma_o^s = \{\alpha_s, \beta_s, \gamma_s\}$ e $\Sigma_o^l = \{\beta_l\}$ são construídos. No passo 3, a fila de estados F é definida e inicializada como $F = [(x_0, 0)]$. Na sequência, o passo 4 é repetido até F se tornar vazia.

Nos passos 4.1 e 4.2, definem-se $(x, q) = (x_0, 0)$, $F = []$ e $X_e = \{(x_0, 0)\}$. No passo 4.3, cria-se o conjunto de índices I_o , o qual armazena os índices dos eventos observáveis em q . Dado que $q = 0$, então $I_o = \emptyset$. No passo 4.4, calcula-se, para cada $y \in I_o$, o tempo mínimo decorrido desde a ocorrência de q_y , o qual é denotado por $min_{et}(y)$. Uma vez que $I_o = \emptyset$, esse passo não é computado nessa primeira iteração. No passo 4.5 (resp. passo 4.6), definem-se as transições a partir de (x, q) rotuladas por eventos em Σ (resp. Σ_o^s ou Σ_o^l) e, portanto, tais transições correspondem a ocorrências de eventos na planta (resp. o sucesso ou a perda de observações). Então, no passo 4.5, definem-se duas transições a partir de $(x_0, 0)$, as quais são rotuladas pelos eventos α e η , uma vez que $\Gamma(x_0) = \{\alpha, \eta\}$ e $q = 0$. Os estados alcançados por essas transições são, respectivamente, (x_1, α) e $(x_4, 0)$, os quais são adicionados a F , a qual se torna $F = [(x_1, \alpha), (x_4, 0)]$. Note que o evento α (resp. η) é adicionado (resp. não é adicionado) à segunda componente do estado alcançado, porque ele é um evento observável (resp. não observável). Por sua vez, como $q = 0$, nenhuma transição é definida no passo 4.6.

Após algumas iterações do passo 4, tem-se que $(x, q) = (x_5, \alpha)$. Então, no passo 4.5, define-se uma transição rotulada por β de (x_5, α) para $(x_2, \alpha 0.5\beta)$, um vez que $\min_{et}(1) + t_{min}(x_5, \beta) = 0.5 < T_1 = 0.9$, ou seja, β pode ocorrer antes de α ser observado. No passo 4.6, a transição rotulada por α_s para o estado $(x_5, 0)$ é definida, a qual representa o sucesso da observação de α .

Por meio da figura 4, pode-se observar que, na iteração do passo 4 em que $(x, q) = (x_2, \alpha 0.8)$, embora $\gamma \in \Gamma(x_2)$, não se define uma transição rotulada por γ a partir de $(x_2, \alpha 0.8)$, uma vez que o evento γ não pode ocorrer antes da transmissão de α terminar. Por sua vez, na iteração do passo 4 em que $(x, q) = (x_3, \alpha 0.5\beta 0.2\gamma)$, embora γ pertença q , sua observação não pode ser recebida pelo supervisor no estado $(x_3, \alpha 0.5\beta 0.2\gamma)$, uma vez que β ocorreu primeiro do que γ e a sua observação ainda está sendo transmitida, e, adicionalmente, as ocorrências de β e γ são transmitidas por meio do mesmo canal.

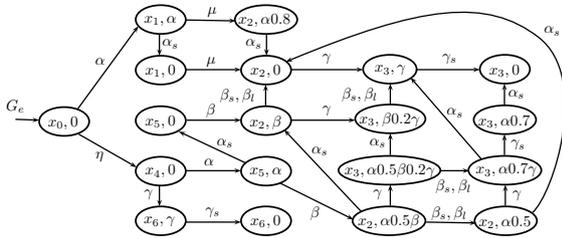


Figura 4: Autômato G_e obtido aplicando-se o algoritmo 1 no exemplo 2.

Observação 1 Sejam $T = \max\{T_i : i \in \{1, \dots, m\}\}$, $t = \min\{t_{min}(x, \sigma) : (x, \sigma) \in X \times \Sigma \wedge \sigma \in \Gamma(x)\}$, e $\mathcal{T} = \max\{z \in \mathbb{Z} : z < T/t\}$, em que \mathbb{Z} denota o conjunto dos números inteiros. Por meio de uma manipulação algébrica, pode-se inferir que a complexidade computacional do algoritmo 1 é $O(|X|^2 \cdot |\Sigma_o| \cdot (|\Sigma_o| + |\Sigma|)^{\mathcal{T}+1})$.

Teorema 1 Seja G_e o autômato gerado usando o algoritmo 1. Então, $L(G_e)$ é formada por, e somente por, seqüências estendidas que podem ser geradas pelo SEDRT na presença de atrasos e perdas de observações.

Definição 2 A operação Extensão é o mapeamento $E : 2^{L(G)} \rightarrow 2^{\Sigma_e^*}$ em que $E(L) = P_e^{-1}(L) \cap L(G_e)$, para todo $L \subseteq L(G)$.

De acordo com a definição 2, para uma dada seqüência $s \in L(G)$, a linguagem estendida $E(\{s\})$ (denotada, daqui em diante, simplesmente por $E(s)$) é formada por aquelas seqüências estendidas que podem ser geradas quando s é executada pela planta.

Seja uma especificação K , gerada por um autômato H , tal que $K \subseteq L(G)$. Na abordagem

proposta na próxima seção, dois autômatos associados à K são construídos: $H_e := H \parallel G_e$, cuja linguagem gerada é $E(K)$, e H_0 , cuja linguagem gerada deve ser igual a sublinguagem de $E(K)$, a ser denotada por $E_0(K)$, que é formada pelas seqüências estendidas geradas quando a planta executa as seqüências em K e não ocorrem perdas ou atrasos relevantes de observação (atrasos maiores que os tempos decorridos entre as ocorrências dos eventos).

Algoritmo 2 (Construção do autômato H_0)

Entrada: H_e , autômato cuja linguagem gerada é $E(K)$.

Saída: Autômato H_0 .

Passo 1: Defina o autômato $A := (\{y_0, y_1\}, \Sigma_e, f_a, \Gamma_a, y_0)$, em que: (i) $\forall \sigma \in \Sigma_e, f_a(y_0, \sigma) = y_0$, se $\sigma \in \Sigma_{uo}, f_a(y_0, \sigma) = y_1$, se $\sigma \in \Sigma_o$, e $f_a(y_0, \sigma)$ não é definido, se $\sigma \in \Sigma_o^s \cup \Sigma_o^l, f_a(y_1, \sigma) = y_0$, se $\sigma \in \Sigma_o^s$, e $f_a(y_1, \sigma)$ não é definido, se $\sigma \in \Sigma \cup \Sigma_o^l$; (ii) $\Gamma_a(y_0) = \Sigma$ e $\Gamma_a(y_1) = \Sigma_o^s$;

Passo 2: $H_0 := A \parallel H_e$.

Exemplo 3 Considere o SEDRT ilustrado nas figuras 2(a,b), em que $\Sigma = \{\alpha, \beta, \gamma, \mu, \eta\}$, $\Sigma_o = \{\alpha, \beta, \gamma\}$ e $\Sigma_{io} = \{\beta\}$, e a especificação K gerada pelo autômato H , apresentado na figura 5(a). Os Autômatos $H_e = H \parallel G_e$ e H_0 , no qual o último foi obtido aplicando-se o algoritmo 2, são apresentados na figura 5(b) em que H_0 corresponde a parte de H_e representada por linhas sólidas.

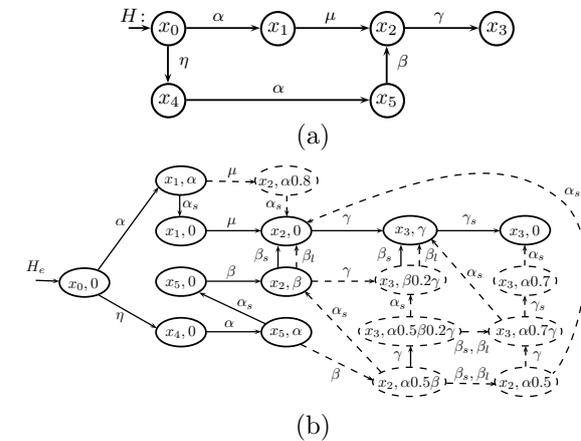


Figura 5: Autômatos do exemplo 3, cujas linguagens geradas são: (a) K e (b) $E(K)$ (linhas sólidas e tracejadas) e $E_0(K)$ (linhas sólidas).

5 Controle supervisorío em rede

5.1 Formulação do problema

De acordo com o modelo de SEDRTs apresentado na seção 4, na presença de atrasos e perdas de observações, a linguagem observada pelo supervisor é $\psi^{-1}(P_{e,s}(E(L(G))))$. Portanto, como ilustrado na figura 6, a ação de controle aplicada por um

supervisor em rede S_{net} , após a execução de uma sequência $s \in L(G)$, será igual a $S_{net}(\psi^{-1}(s_o^s))$ para algum $s_o^s \in P_{e,s}(E(s))$.

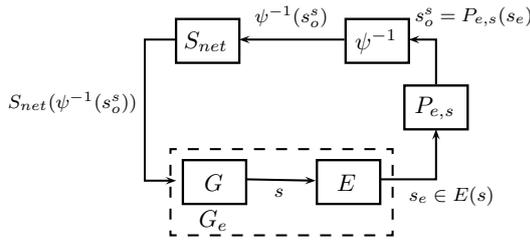


Figura 6: Arquitetura de controle supervisorio em rede sujeito a atrasos e perdas de observação.

Com o objetivo de lidar com a ambiguidade na ação de controle, duas linguagens associadas ao sistema em malha fechada S_{net}/G são consideradas: (i) $L(S_{net}/G)$, que corresponde a linguagem gerada pelo sistema em malha fechada quando não ocorrem perdas ou atrasos de observação; (ii) $L^\downarrow(S_{net}/G)$, que corresponde ao limite superior da linguagem gerada pelo sistema em malha fechada e, portanto, é formado por, e somente por, sequências que podem ser geradas pelo sistema em malha fechada supondo possíveis atrasos ou perdas de observações. $L^\downarrow(S_{net}/G)$ é definida recursivamente da seguinte forma: (i) $\varepsilon \in L^\downarrow(S_{net}/G)$, e (ii) $s\sigma \in L^\downarrow(S_{net}/G) \Leftrightarrow s \in L^\downarrow(S_{net}/G) \wedge s\sigma \in L(G) \wedge (\exists \sigma_1 \sigma_2 \dots \sigma_k \in E(s\sigma)) [\forall i \in \{1, \dots, k\}, \sigma_i \in (\Sigma_e \setminus \Sigma_c) \cup S_{net}(\psi^{-1}(P_{e,s}(\sigma_1 \dots \sigma_{i-1})))]$. Essas linguagens induzem ao seguinte problema de controle supervisorio em rede. Dada uma linguagem prefixo-fechada K , calcular um supervisor em rede $S_{net} : \Sigma_o^* \rightarrow 2^{\Sigma}$ tal que $L(S_{net}/G) = L^\downarrow(S_{net}/G) = K$, ou seja, quando não ocorrer atraso ou perda de observação, o sistema em malha fechada deve ser capaz de gerar toda a linguagem K e, quando ocorrerem perdas e/ou atrasos de observações, a linguagem gerada pelo sistema em malha fechada deve permanecer contida em K .

5.2 Existência e projeto de supervisores em rede

Com o objetivo de determinar uma condição de existência de supervisores em rede, defina o conjunto estendido de eventos não controláveis $\Sigma_{euc} = \Sigma_{uc} \dot{\cup} \Sigma_o^s \dot{\cup} \Sigma_o^l$, e considere a seguinte classe de linguagens estendidas:

$$\mathcal{K}_C(K) = \{\overline{K}_e \subseteq \Sigma_e^* : (E_0(K) \subseteq \overline{K}_e \subseteq E(K)) \wedge (\overline{K}_e \text{ é controlável e.r.a } E(L(G)) \text{ e } \Sigma_{euc})\}.$$

Definição 3 (controlabilidade em rede)

Uma linguagem prefixo-fechada $K \subseteq L(G)$ é controlável em rede em relação a $L(G)$, E e Σ_{euc} se $\mathcal{K}_C(K) \neq \emptyset$.

Considere, agora, a seguinte classe de linguagens estendidas pertencentes a $\mathcal{K}_C(K)$:

$$\mathcal{K}_{CO}(K) = \{\overline{K}_e \in \mathcal{K}_C(K) : \overline{K}_e \text{ é observável e.r.a } E(L(G)) \text{ e } P_{e,s}\}.$$

Definição 4 (observabilidade em rede)

Uma linguagem prefixo-fechada e controlável em rede $K \subseteq L(G)$ é observável em rede em relação a $L(G)$, E e $P_{e,s}$ se $\mathcal{K}_{CO}(K) \neq \emptyset$.

De acordo com as definições 3 e 4, uma linguagem prefixo-fechada K será controlável em rede e observável em rede se, e somente se, existir uma linguagem estendida controlável e observável \overline{K}_e tal que $E_0(K) \subseteq \overline{K}_e \subseteq E(K)$. Como a classe de superlinguagens prefixo-fechadas, controláveis e observáveis de $E_0(K)$ é fechada em relação a interseção de linguagens, então, a superlinguagem ínfima de $E_0(K)$, prefixo-fechada, controlável (em relação a $E(L(G))$ e Σ_{euc}) e observável (em relação a $E(L(G))$ e $P_{e,s}$), denotada por $E_0(K)^{\downarrow CO}$, sempre existe. Portanto, uma linguagem prefixo-fechada $K \subseteq L(G)$ será controlável em rede e observável em rede se, e somente se, $E_0(K)^{\downarrow CO} \subseteq E(K)$.

Quando K for controlável em rede e observável em rede, sempre será possível obter uma linguagem $\overline{K}_e \in \mathcal{K}_{CO}(K)$, dado que, no pior caso, pode-se definir \overline{K}_e igual a $E_0(K)^{\downarrow CO}$. Adicionalmente, como \overline{K}_e é controlável (em relação a $E(L(G))$ e Σ_{euc}) e observável (em relação a $E(L(G))$ e $P_{e,s}$), pode-se projetar um supervisor $S_e : \Sigma_o^{s*} \rightarrow 2^{\Sigma_e}$ tal que $L(S_e/G_e) = \overline{K}_e$ aplicando-se a teoria de controle supervisorio (Cassandras and Lafortune, 2008), considerando a planta G_e , a especificação \overline{K}_e e supondo os conjuntos Σ_{euc} e Σ_o^s como os conjuntos de eventos não controláveis e observáveis, respectivamente. Então, um supervisor em rede pode ser calculado a partir de S_e , como determinado pelo seguinte resultado.

Lema 2 *Seja \overline{K}_e uma linguagem pertencente a classe $\mathcal{K}_{CO}(K)$, e um supervisor $S_e : \Sigma_o^{s*} \rightarrow 2^{\Sigma_e}$ tal que $L(S_e/G_e) = \overline{K}_e$. Considere o supervisor em rede $S_{net} : \Sigma_o^* \rightarrow 2^{\Sigma}$ definido como $S_{net}(s_o) = S_e(\psi(s_o)) \cap \Sigma$, para todo $s_o \in \Sigma_o^*$. Então, $L(S_{net}/G) = L^\downarrow(S_{net}/G) = K$.*

No resultado a seguir, apresenta-se uma condição necessária e suficiente para a existência de um supervisor em rede.

Teorema 3 *Considere um SEDRT que satisfaz as hipóteses A1 a A5, e cuja planta é modelada por um autômato G . Considere, também, uma especificação prefixo-fechada $K \subseteq L(G)$. Existe um supervisor em rede $S_{net} : \Sigma_o^* \rightarrow 2^{\Sigma}$ tal que $L(S_{net}/G) = L^\downarrow(S_{net}/G) = K$ se, e somente se, K é controlável em rede em relação a $L(G)$, E e Σ_{euc} e observável em rede em relação a $L(G)$, E e $P_{e,s}$.*

Exemplo 4 *Considere o SEDRT apresentado nas figuras 2(a,b), e suponha que $\Sigma = \{\alpha, \beta, \gamma, \mu, \eta\}$, $\Sigma_o = \{\alpha, \beta, \gamma\}$, $\Sigma_{lo} = \{\beta\}$ e $\Sigma_c = \{\alpha, \gamma, \mu, \eta\}$. Considere, também, a especificação*

K gerada pelo autômato H apresentado na figura 5(a). Os autômatos cujas linguagens geradas são $E(K)$ e $E_0(K)$, calculados no exemplo 3, são apresentados na figura 5(b). O autômato H^\downarrow que gera $E_0(K)^{\downarrow CO}$ é apresentado na figura 7. Comparando as figuras 5(b) e 7, pode-se concluir que $E_0(K)^{\downarrow CO} \subseteq E(K)$ e, portanto, K é controlável em rede e observável em rede, o que implica, de acordo com o teorema 3, que existe um supervisor em rede S_{net} . Com o objetivo de calcular S_{net} , calcula-se inicialmente o supervisor S_e tal que $L(S_e/G_e) = E_0(K)^{\downarrow CO}$, supondo-se $\Sigma_{euc} = \{\alpha_s, \beta, \beta_s, \beta_l, \gamma_s\}$ e $\Sigma_o^s = \{\alpha_s, \beta_s, \gamma_s\}$. O supervisor S_e obtido é definido como: $S_e(\varepsilon) = \Sigma_e \setminus \{\mu, \gamma, \gamma_s\}$, $S_e(\alpha_s) = \Sigma_e \setminus \{\alpha, \alpha_s, \eta\}$, $S_e(\beta_s) = \{\alpha_s\}$, $S_e(\alpha_s\beta_s) = \{\gamma, \gamma_s\}$, e $S_e(\beta_s\alpha_s) = S_e(\alpha_s\gamma_s) = S_e(\alpha_s\beta_s\gamma_s) = \emptyset$. Então, o supervisor em rede S_{net} tal que $L(S_{net}/G) = L^\downarrow(S_{net}/G) = K$, obtido a partir de S_e da forma descrita pelo lema 2, é definido como: $S_{net}(\varepsilon) = \{\alpha, \beta, \eta\}$, $S_{net}(\alpha) = \{\beta, \gamma, \mu\}$, $S_{net}(\alpha\beta) = \{\gamma\}$ e $S_{net}(\beta) = S_{net}(\beta\alpha) = S_{net}(\alpha\gamma) = S_{net}(\alpha\beta\gamma) = \emptyset$.

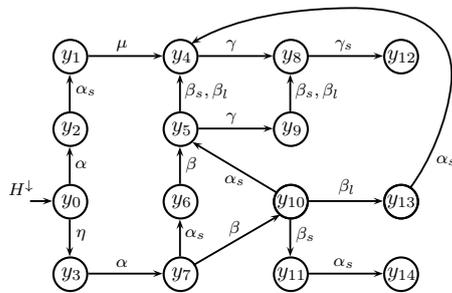


Figura 7: Autômato que gera $E_0(K)^{\downarrow CO}$.

6 Conclusão

Embora seja possível calcular um supervisor em rede, sempre que existe um, definindo-se $\bar{K}_e = E_0(K)^{\downarrow CO}$ e projetando S_{net} da forma descrita na seção 5.2, a permissividade do supervisor nas situações em que ocorrem atrasos e/ou perdas de observações pode ser melhorada por meio da busca de linguagens $\bar{K}_e \subseteq \mathcal{K}_{CO}(K)$ mais permissivas. Nesse contexto, tem-se trabalhado para entender a definição de observabilidade relativa (Cai et al., 2015; Alves et al., 2017) para observabilidade relativa em rede, com o objetivo de obter o supervisor maximamente permissivo dentro da classe de linguagens controláveis e relativamente observáveis em rede. Outra possível extensão desse trabalho consiste no relaxamento da hipótese A4, com o objetivo de considerar atrasos e perdas de ações de controle.

Referências

Alves, M. V. S., Basilio, J. C., da Cunha, A. E. C., Carvalho, L. K. and Moreira, M. V. (2014).

Robust supervisory control against intermittent loss of observations, *12th Workshop on Discrete Event Systems*, Cachan, France, pp. 294–299.

Alves, M. V. S., Carvalho, L. K. and Basilio, J. C. (2017). New algorithms for verification of relative observability and computation of supremal relatively observable sublanguage, *IEEE Transactions on Automatic Control*. DOI: 10.1109/TAC.2016.2644640.

Balemi, S. (1994). Input/output discrete event processes and communication delays, *Discrete Event Dynamic Systems* 4(1): 41–85.

Cai, K., Zhang, R. and Wonham, W. M. (2015). Relative observability of discrete-event systems and its supremal sublanguages, *IEEE Transactions on Automatic Control* 60(3): 659–670.

Carvalho, L. K., Basilio, J. C. and Moreira, M. V. (2012). Robust diagnosis of discrete-event systems against intermittent loss of observations, *Automatica* 48(9): 2068–2078.

Cassandras, C. G. and Lafontaine, S. (2008). *Introduction to Discrete Event Systems*, 2nd edn, Springer, New York, NY, USA.

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

Nunes, C. E. V., Moreira, M. V., Alves, M. V. S. and Basilio, J. C. (2016). Network codiagnosability of discrete-event systems subject to event communication delays, *13th International Workshop on Discrete Event Systems*, Xi'an, China, pp. 217–223.

Park, S. J. and Cho, K. H. (2006). Delay-robust supervisory control of discrete-event systems with bounded communication delays, *IEEE Transactions on Automatic Control* 51(5): 911–915.

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

Park, S. J. and Cho, K. H. (2008). Nonblocking supervisory control of timed discrete event systems under communication delays: The existence conditions, *Automatica* 44(4): 1011 – 1019.

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

Shu, S. and Lin, F. (2015). Supervisor synthesis for networked discrete event systems with communication delays, *IEEE Transactions on Automatic Control* 60(8): 2183–2188.

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