
CONTROLE SUPERVISÓRIO HIERÁRQUICO MODULAR POR AGREGAÇÃO DE ESTADOS

César R. C. Torrico*
torrico@das.ufsc.br

José E. R. Cury*
cury@das.ufsc.br

*Departamento de Automação e Sistemas, Universidade Federal de Santa Catarina, Caixa Postal 470
Campus Trindade, CTC, 88.040-000 Florianópolis, SC, Brasil – Fax: (55)(48)331-9934

ABSTRACT

This paper presents a model for the hierarchical modular control of discrete event systems based on state aggregation and classical modular control. A combination of the mentioned architectures is presented, and necessary and sufficient conditions are given such that the supervisor's components designed individually when acting jointly over the plant lead to a nonblocking and consistent solution.

KEYWORDS: Discrete Event Systems, Supervisory Control, Hierarchical Control, Modular Control.

RESUMO

Este artigo apresenta um modelo para controle hierárquico modular de sistemas a eventos discretos baseado no controle hierárquico por agregação de estados e no controle modular clássico. Apresenta-se uma combinação das arquiteturas mencionadas, e são dadas condições necessárias e suficientes para que supervisores projetados individualmente sobre a planta quando atuando conjuntamente levem a uma solução consistente e não bloqueante.

PALAVRAS-CHAVE: Sistemas a Eventos Discretos, Controle Supervisório, Controle Hierárquico, Controle Modular

1 INTRODUÇÃO

Em muitas áreas de aplicação, a complexidade dos processos incrementaram enormemente durante as últimas décadas. Isto principalmente porque a integração entre as componentes dos processos cresce constantemente para permitir que os recursos sejam usados com mais eficiência. Os controles hierárquico e modular são uma opção para manipular problemas complexos através respectivamente da decomposição vertical e horizontal do problema em sub-problemas menores, para depois montar as suas soluções numa estrutura hierárquica e modular.

O formalismo sobre controle hierárquico de Sistemas a Eventos Discretos (SEDs) foi introduzido por Zhong e Wonham (1990) formulado para especificações prefixo fechadas dentro do contexto de Ramadge e Wonham (1989). Na sequência foi estendido por Wong e Wonham (1996) permitindo a formulação do problema para linguagens com marcação. De outro ponto de vista, Caines e Hubbard (1998) desenvolveram uma abordagem para controle hierárquico por agregação de estados. Em (Torrico e Cury, 2002) foi introduzido um novo formalismo para o controle hierárquico de sistemas a eventos discretos envolvendo as abordagens mencionadas anteriormente em parceria com o novo modelo generalizado proposto em (Cury et al., 2001; Cury et al., 2004) para o controle de SEDs com estruturas de controle avançadas e marcação dinâmica associada. Em todas estas abordagens sobre controle hierárquico propõe-se uma hierarquia de dois níveis, um nível associado ao operador e um outro nível associado ao gerente.

Artigo submetido em em 12/12/02

1a. Revisão em 03/10/03

Aceito sob recomendação do Ed. Assoc. Prof. Cairo L. Nascimento

O controle modular de SEDs foi inicialmente proposto por Wonham e Ramadge (1988). A síntese modular permite que problemas complexos possam ser decompostos em módulos mais simples. Enquanto supervisores são projetados para serem não bloqueantes (em relação à planta) quando operam individualmente, não está garantida que as suas combinações não serão bloqueantes. Por isto é necessário verificar que a síntese dos componentes seja não conflitante.

Neste trabalho propõe-se uma combinação da arquitetura de controle hierárquico e modular envolvendo as abordagens desenvolvidas em (Torrice e Cury, 2002) e (Wonham e Ramadge, 1988). À decomposição hierárquica (decomposição vertical) proposta em (Torrice e Cury, 2002), adiciona-se a decomposição modular (decomposição horizontal) de (Wonham e Ramadge, 1988). O objetivo deste trabalho é propor uma arquitetura hierárquica modular baseada na agregação de estados, tal que, esta arquitetura permita um comportamento consistente entre os níveis de hierarquia e que a ação conjunta de vários controladores gerencias implementados no baixo nível sejam não bloqueantes.

O artigo está organizado como segue. A seção 2 introduz os conceitos e resultados principais sobre controle por agregação de estados; a seção 3 introduz o problema de controle hierárquico modular e apresenta uma solução a este problema; finalmente a seção 4 ilustra a metodologia proposta através de um exemplo de uma linha de transferência.

2 CONTROLE HIERÁRQUICO POR AGREGAÇÃO DE ESTADOS

Nesta seção, resume-se os principais conceitos e resultados sobre controle hierárquico por agregação de estados apresentados em (Torrice e Cury, 2002).

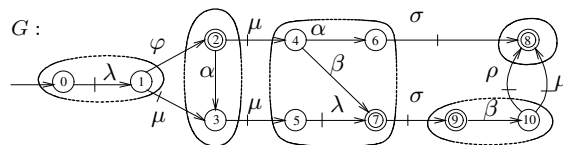
Propõe-se uma arquitetura de dois níveis de hierarquia, operacional e agregado. No nível operacional, o autômato da planta é dado por uma quintupla, $G = (\Sigma, X, \delta, q_0, Q_m)$, onde Σ é o conjunto finito de eventos, consistindo de eventos não-controláveis (Σ_u) e eventos controláveis (Σ_c), i.e. $\Sigma = \Sigma_u \cup \Sigma_c$, X é o conjunto finito de estados, $\delta : X \times \Sigma \rightarrow X$ é a função de transição de estados (em geral uma função parcialmente definida), $q_0 \in X$ é o estado inicial e $Q_m \subseteq X$ é o conjunto de estados marcados.

Uma partição do conjunto de estados X , denominada partição π , tem a característica a seguir: $\pi = \{X_1, X_2, \dots, X_N\}$ com $\bigcup X_i = X$, $X_i \neq \emptyset$ e $X_i \cap X_j = \emptyset$ para $i \neq j$.

Dado um autômato G definido sobre Σ , um conjunto de eventos relevantes $\Sigma^A \subset \Sigma$, e uma partição $\pi = \{X_i : x_j, x_k \in X_i \leftrightarrow x_j, x_k \in X \wedge \exists s \in (\Sigma - \Sigma^A)^*, \delta(x_j, s) = x_k\}$, define-se *autômato agregado* à quádrupla $G^A = (\Sigma^A, \pi, \delta^A, q_0^A)$,

onde Σ^A é o conjunto de eventos, π é o conjunto finito de estados obtido após a partição de X , $\delta^A : \pi \times \Sigma^A \rightarrow \pi$ é a função de transição parcial de estados tal que $\delta^A(X_i, \sigma) = X_j$ se $(\exists x \in X_i, \exists y \in X_j : \delta(x, \sigma) = y)$, e $q_0^A \in \pi$ é o estado inicial, tal que $q_0 \in q_0^A$.

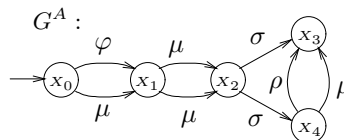
O autômato agregado como definido anteriormente mostra unicamente eventos relevantes. Na figura 1 apresentam-se autômatos G e G^A , mostrando como os estados originais foram agregados dentro de blocos- π .



$$\Sigma = \{\alpha, \beta, \lambda, \phi, \mu, \sigma, \rho\}, \Sigma^A = \{\phi, \mu, \sigma, \rho\}$$

$$\Sigma_u = \{\alpha, \beta, \phi\}$$

(a) Agregação de estados



(b) Autômato agregado- π

Figura 1: Exemplo de agregação

Dado um autômato G e o autômato agregado correspondente G^A , definimos o mapeamento canônico $\Theta : L(G) \rightarrow L(G^A)$, como $\Theta(\epsilon) = \epsilon$ e

$$\Theta(s\sigma) = \begin{cases} \Theta(s)\sigma & \text{se } \sigma \in \Sigma^A \\ \Theta(s) & \text{caso contrário} \end{cases}$$

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

O mapeamento canônico é estendido naturalmente para linguagens como $\Theta : 2^{\Sigma^*} \rightarrow 2^{\Sigma^{A*}}$, $\Theta(K) = \{\Theta(s) : s \in K\}$

Seja $L^A = \Theta(L(G))$ o mapeamento para o nível agregado de uma linguagem $L(G)$ e seja $L(G^A)$ a linguagem gerada pelo autômato agregado. Nota-se que L^A é diferente de $L(G^A)$ uma vez que algumas cadeias de $L(G^A)$ são impossíveis de serem obtidas através do mapeamento das cadeias de baixo nível. Pode-se mostrar que $L^A \subseteq L(G^A)$.

Para obter um autômato agregado com boas propriedades, previamente é necessário tratar o problema de não determinismo que pode aparecer em G^A . Seja o autômato agregado G^A , a seguir consideram-se as duas situações possíveis de ocorrer:

- (a) G^A é propriamente não determinista, isto é, existe $X_i, X_j, X_k \in \pi$ e $\sigma \in \Sigma^A$, tal que, $\delta^A(X_i, \sigma) = \{X_j, X_k\}$, com $X_j \neq X_k$. Na figura 1, as saídas do bloco X_2 com o evento σ ilustra um exemplo deste caso.
- (b) Existe $X_i, X_j \in \pi$ e $\sigma \in \Sigma^A$ tal que $\delta(X_i, \sigma) = X_j$, e tal que, existe $x, y \in X_i$ e $z, w \in X_j$, $z \neq w$, com $\delta(x, \sigma) = z$ e $\delta(y, \sigma) = w$. Na figura 1, as saídas do bloco X_1 com o evento μ ilustra um exemplo deste caso.

Considera-se também o caso (b) como não determinismo porque as continuções após aquelas transições afetadas não necessariamente serão iguais.

A princípio o não determinismo será eliminado renomeando os eventos $\sigma \in \Sigma$, que geram não determinismo, em eventos $\sigma, \sigma', \sigma'', \dots$ que irão a aumentar os alfabetos Σ e Σ^A . Por exemplo, o não determinismo da agregação ilustrada na figura 1, eliminou-se renomeando o evento μ que sai do bloco X_1 em μ, μ' , e o evento σ que sai do bloco X_2 em σ, σ' . Este refinamento em G^A permite eliminar qualquer ambigüidade no modelo agregado.

A estrutura de controle para G^A é construída dependente de cada bloco $X_i \in \pi$ e de cada estado de entrada no bloco $x_j \in X_i$ como em (Torrico e Cury, 2002). Consideram-se estados de entrada de um bloco X_i , denotado por $I(X_i)$, aqueles estados que são atingidos por eventos em Σ^A .

A estrutura de controle do modelo agregado é definida como um mapeamento $\Gamma : (\pi, X) \rightarrow 2^{2^{\Sigma^A} \times \{M, N\}}$, a qual associa a cada bloco $X_i \in \pi$ entrando por $x_j \in I(X_i)$ um conjunto de padrões de controle

$$\Gamma_{ij} = \{(\gamma, \#) \in 2^{\Sigma^A} \times \{M, N\}\}$$

onde $(\gamma, \#) \in \Gamma_{ij}$ é um padrão de controle válido em X_i entrando por x_j com,

1. $\gamma \subset \Sigma^A$ sendo um conjunto de eventos habilitados após X_i ;
2. $\# = M$ sendo um atributo de marcação significando que a cadeia atual executada em $L(G^A)$ é considerada uma tarefa do sistema; e
3. $\# = N$ significando que a cadeia atual executada em $L(G^A)$ não é considerada uma tarefa do sistema.

Em (Torrico e Cury, 2002) mostra-se que o conjunto de padrões de controle $\Gamma_{ij} \in \Gamma$ satisfaz os seguintes requerimentos, para todo $X_i \in \pi$ e $x_j \in X_i$,

1. $(\gamma_1, N), (\gamma_2, N) \in \Gamma_{ij} \rightarrow (\gamma_1 \cup \gamma_2, N) \in \Gamma_{ij}$

2. $(\gamma_1, M), (\gamma_2, \#) \in \Gamma_{ij} \rightarrow (\gamma_1 \cup \gamma_2, M) \in \Gamma_{ij}, \# = M, N$

Não obstante, a estrutura de controle Γ como definida depende de $X_i \in \pi$ e $x_j \in I(X_i)$. Uma vez que a informação do estado de entrada não poderá ser vista no autômato do nível agregado, propõe-se a representação de um autômato agregado com estruturas de controle dependentes do estado, denotado por G_s^A .

Para poder representar o autômato agregado com estruturas de controle dependentes do estado, é necessário refinar este, fazendo uma divisão de cada bloco para cada estado de entrada que tenha um conjunto de estruturas de controle diferente, garantindo desta forma uma única estrutura para cada bloco. O número de estados do autômato resultante G_s^A , no pior caso será igual ao somatório de todos os estados de entrada de todos os blocos. Como exemplo, na figura 1, apresenta-se um autômato G com uma agregação dada e logo abaixo, mostra-se o autômato G_s^A resultante do refinamento da agregação original tal que as estruturas de controle para o bloco X_1 entrando pelo estado 2 e pelo estado 3 são diferentes ($\Gamma_{1,2} \neq \Gamma_{1,3}$) e conseqüentemente o bloco X_1 é refinado em X'_1 e X''_1 .

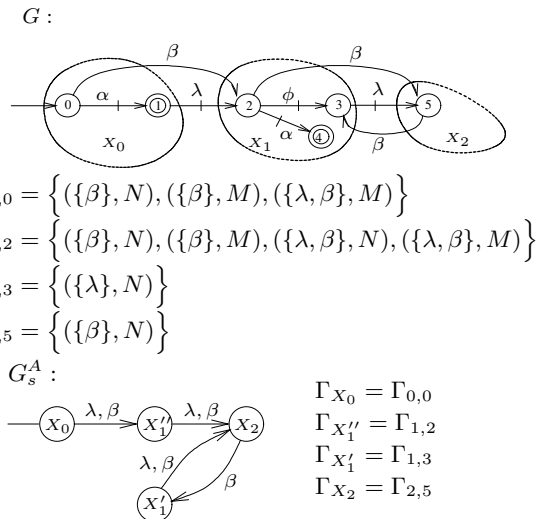


Figura 2: Autômato G_s^A , com estruturas de controle dependentes do estado.

A partir daqui G_s^A será referido como G^A , assim como o conjunto de padrões de controle associados a um novo estado X_i de G^A será referido como Γ_i .

O supervisor para o modelo agregado, S^A é definido como um mapeamento, $S^A : \pi \rightarrow 2^{\Sigma^A} \times \{M, N\}$. Para um dado $X_i \in \pi$, o supervisor seleciona um padrão de controle

$S^A(X_i) = (\gamma, \#) \in \Gamma_i$, tal que em malha fechada, o conjunto de eventos ativos após X_i ¹ é reduzido a $\gamma \cap \Sigma^A(X_i)$, e a cadeia que chega em X_i é considerada marcada se $\# = M$ e não marcada se $\# = N$.

O comportamento do sistema em malha fechada S^A/G^A é representado por um par de linguagens, uma linguagem prefixo fechada $L(S^A/G^A)$ e uma linguagem marcada $L_m(S^A/G^A)$, tal que $L_m(S^A/G^A) \subseteq L(S^A/G^A) \subseteq L(G^A)$. A linguagem prefixo fechada $L(S^A/G^A)$ é definida recursivamente como:

1. $\epsilon \in L(S^A/G^A)$
2. $s\sigma \in L(S^A/G^A) \leftrightarrow s \in L(S^A/G^A) \wedge s\sigma \in L(G^A) \wedge \sigma \in \gamma$, onde, para $X_i = \delta^A(q_0^A, s)$, $S(X_i) = (\gamma, \#)$

e a linguagem marcada $L_m(S^A/G^A)$ como

$$s \in L_m(S^A/G^A) \leftrightarrow s \in L(S^A/G^A) \wedge S(X_i) = (\gamma, M), \text{ onde } X_i = \delta^A(q_0^A, s)$$

O comportamento em malha fechada $L(S^A/G^A)$ é uma linguagem prefixo fechada e contém cadeias em $L(G^A)$ permitidas por S^A sob supervisão, e o comportamento marcado em malha fechada $L_m(S^A/G^A)$ contém as cadeias em $L(S^A/G^A)$ onde o supervisor escolheu o padrão de controle com atributo de marcação “M”. Em geral $L_m(S^A/G^A) \subseteq L(S^A/G^A)$, e o supervisor é dito ser não bloqueante se $L_m(S^A/G^A) = L(S^A/G^A)$. O comportamento em malha fechada pode ser representado por um gerador T onde $L(T) = L(S^A/G^A)$ e $L_m(T) = L_m(S^A/G^A)$.

De modo a propor uma solução para o problema de controle supervisorório do modelo agregado é introduzido o conceito de Γ -compatibilidade (Cury et al., 2004) como segue.

Definição 1 Γ -compatibilidade: A linguagem $K \subseteq L(G^A)$ é Γ -compatível em relação a $L(G^A)$ se, e somente se, $K = \emptyset$ ou;

1. $(\forall s \in K)(\exists(\gamma, M) \in \Gamma_i) : \gamma \cap \Sigma_{L(G^A)}(s) = \Sigma_K(s)$,
e
2. $(\forall s \in (\overline{K} - K))(\exists(\gamma, N) \in \Gamma_i) : \gamma \cap \Sigma_{L(G^A)}(s) = \Sigma_K(s)$

onde $\delta^A(q_0^A, s) = X_i$ e $\Sigma_L(s)$ representa o conjunto ativo de eventos em L após s .

¹dado um autômato $G = (\Sigma, X, \delta, q_0, Q_m)$, o conjunto de eventos ativos em $x \in X$ é $\Sigma(x) = \{\sigma \in \Sigma : \delta(x, \sigma) \text{ definido}\}$.

Em (Cury et al., 2004) mostra-se que a Γ -compatibilidade de uma linguagem é condição necessária e suficiente para existência de um supervisor não bloqueante que a implemente em malha fechada.

Caso uma linguagem K a qual especifica o comportamento desejado não seja Γ -compatível, em (Cury et al., 2004) mostra-se que existe uma aproximação de K que é única e ótima, no sentido de ser minimamente restritiva. No caso regular, apresenta-se um algoritmo para o cálculo desta aproximação.

3 O CONTROLE HIERÁRQUICO MODULAR

O controle hierárquico modular refere-se à decomposição do problema seja horizontal como verticalmente. Nesta seção apresenta-se o caso da decomposição vertical em dois níveis de hierarquia e a decomposição horizontal em vários níveis onde cada nível está associado a uma especificação.

3.1 O problema de controle hierárquico modular

Considera-se uma planta G , um modelo abstrato G^A , dois controladores S_i^A , $i = 1, 2$ para G^A , um controlador S para G , e um *operador de conjunção* (\odot), conectados tal como mostrado na Figura 3. G é a planta a ser controlada no mundo real por um operador S , enquanto que G^A é uma abstração de G obtida pela agregação de seus estados que será controlada pela ação conjunta de dois gerentes S_i^A , $i = 1, 2$. A planta G informa ao nível agregado somente alguns eventos considerados relevantes ($\sigma \in \Sigma^A$), atualizando o modelo G^A , mas também informa ao operador S a ocorrência de qualquer evento $\lambda \in \Sigma$. A cada ocorrência de um evento $\sigma \in \Sigma^A$, cada gerente S_i^A , $i = 1, 2$ envia um comando com_i ao *operador de conjunção* que processa os dois comandos com_i , $i = 1, 2$ e os sintetiza num único comando com a ser implementado pelo operador S . O operador S recebe duas informações, o comando com do operador de conjunção e a ocorrência de eventos $\lambda \in \Sigma$ gerados pela planta G , e com estes dados envia uma entrada de controle Con para a planta G . A entrada de controle Con_i^A resulta ser uma entrada de controle virtual, uma vez que o comportamento de G^A é determinado totalmente pelo comportamento de G .

Sejam L_m e L linguagens definidas sobre um alfabeto Σ com $L = \overline{L_m}$, onde L e L_m representam respectivamente as linguagens gerada e marcada de uma planta G . G^A representa o modelo abstraído de G obtido pela agregação dos seus estados como em (Torricco e Cury, 2002). Σ^A é o alfabeto de eventos para G^A , onde $\Sigma^A \subseteq \Sigma$.

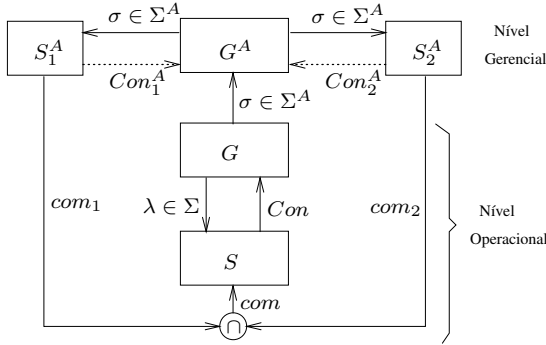


Figura 3: Arquitetura do controle hierárquico modular.

Deseja-se que para duas especificações realizáveis de alto nível a arquitetura proposta possua um comportamento consistente. Assim para $i = 1, 2$, sejam $E_i^A \subseteq L(G^A)$ especificações sobre G^A . A máxima linguagem Γ -compatível sobre $L(G^A)$ para a especificação E_i^A será $\sup \mathcal{CM}(E_i^A, L(G^A)) \subseteq E_i^A \subseteq L(G^A)$ (Cury et al., 2004). Define-se a *modularidade hierárquica*, como a propriedade verificada quando para dois supervisores do nível gerencial S_1^A e S_2^A , tais que, $L_m(S_i^A/G^A) = \sup \mathcal{CM}(E_i^A, L(G^A))$, $i = 1, 2$, $\Theta(L_m(S/G)) = \sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A))$ com $L_m(S/G) = L(S/G)$. Esta condição indica que a imagem da ação do supervisor S sobre a planta G será igual à implementação monolítica no alto nível, ou seja, aquela que seria obtida resolvendo-se o problema de síntese de controle para a especificação $E_1^A \cap E_2^A$. O problema de controle é obter condições necessárias e suficientes para modularidade hierárquica.

3.2 Modelo para controle hierárquico modular

Nesta seção serão detalhados o papel do operador de conjunção e o supervisor S .

O operador de conjunção é um elemento que se encarrega de processar os comandos com_1 e com_2 vindos de S_1^A e S_2^A respectivamente e traduzir num comando com .

Funcionamento do operador de conjunção.-

Na implementação on-line o *operador de conjunção* terá a seguinte função:

Para cada bloco atingido $X_i \in \pi$ de G^A .

- Se $com_1 = (\gamma_1, N) \in \Gamma_i$ e $com_2 = (\gamma_2, \#) \in \Gamma_i$, para $\# = M, N$, então, $com = ([\gamma_1 \cap \gamma_2], N)$ se $com \in \Gamma_i$, senão $com = \text{indefinido!}$

- Se $com_1 = (\gamma_1, M) \in \Gamma_i$ e $com_2 = (\gamma_2, M) \in \Gamma_i$, então, $com = ([\gamma_1 \cap \gamma_2], M)$ se $com \in \Gamma_i$, senão $com = \text{indefinido!}$

A implementação conjunta de S_1^A e S_2^A no baixo nível é realizada pelo supervisor S que interpreta o comando com como no caso do controle hierárquico por agregação de estados apresentado em (Torricco e Cury, 2002) no qual se tem um único supervisor no nível agregado.

Para o caso geral em que se tenha mais de dois supervisores, o funcionamento do *operador de conjunção* será da forma seguinte: Sejam P supervisores,

- Se $com_1 = (\gamma_1, N) \in \Gamma_i$, e $com_k = (\gamma_k, \#) \in \Gamma_i$ para $k = 2, \dots, P$ e $\# = M, N$, então, $com = ([\gamma_1 \cap \gamma_2 \cap \dots \cap \gamma_P], N)$ se $com \in \Gamma_i$, senão $com = \text{indefinido!}$
- Se $com_k = (\gamma_k, M) \in \Gamma_i$, para $k = 1, 2, \dots, P$, então, $com = ([\gamma_1 \cap \gamma_2 \cap \dots \cap \gamma_P], M)$ se $com \in \Gamma_i$, senão $com = \text{indefinido!}$

Observa-se que para um funcionamento adequado do supervisor S , o comando com deverá sempre existir dentro da estrutura de controle ativa. Caso não exista então não existe um supervisor S consistente, no sentido de processar o comando de entrada com . O exemplo a seguir ilustra esta situação.

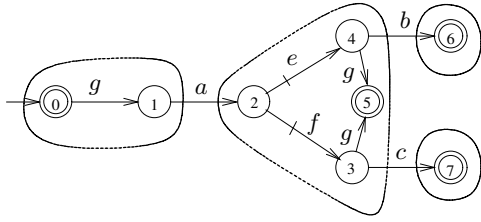
Exemplo 1 Considera-se a planta mostrada na figura 4(a); na figura 4(b) apresenta-se o correspondente modelo agregado com estruturas de controle dependentes do estado.

Para as especificações E_1^A e E_2^A do alto nível apresentadas na figura 5 observa-se que as mesmas são Γ -compatíveis. Portanto estes autômatos podem ser usados como supervisores S_1^A , S_2^A , tais que $L_m(S_1^A/G^A) = E_1^A$ e $L_m(S_2^A/G^A) = E_2^A$. Observa-se que as linguagens realizáveis no nível agregado são modulares, isto é, $\overline{E_1^A} \cap \overline{E_2^A} = \overline{E_1^A \cap E_2^A}$.

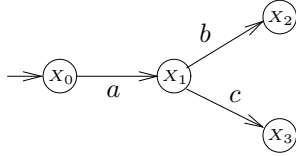
Neste caso, quando o sistema alcança o bloco X_1 , S_1^A envia a entrada de controle $com_1 = (\{b\}, M)$ e S_2^A a entrada de controle $com_2 = (\{c\}, M)$. Para um comportamento consistente teria que existir a entrada de controle (\emptyset, M) , permitindo desabilitar os eventos b e c simultaneamente, o que não ocorre no caso considerado.

No exemplo anterior também é mostrado que a condição de modularidade de linguagens realizáveis no nível agregado, não é suficiente para garantir uma realização não bloqueante.

Para garantir a existência do supervisor S , o teorema a seguir apresenta as condições para que isto seja possível.



(a) Planta



$$\begin{aligned} \Gamma_{X_0} &= \{(\{a\}, M)\} \\ \Gamma_{X_1} &= \{(\{b\}, M), (\{c\}, M), (\{b, c\}, M)\} \\ \Gamma_{X_2} &= \Gamma_{X_3} = \{(\emptyset, M)\} \end{aligned}$$

(b) Autômato agregado com Γ dependente do estado

Figura 4: Caso de estudo

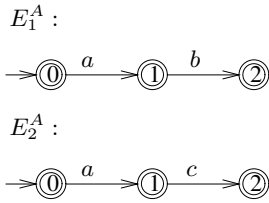


Figura 5: Especificações

Teorema 1 Dados dois supervisores S_1^A e S_2^A tais que $L_m(S_1^A/G^A) = K_1^A$, $L_m(S_2^A/G^A) = K_2^A$, com $\overline{K_1^A} \cap \overline{K_2^A} = \overline{K_1^A \cap K_2^A}$. A Γ -compatibilidade de $K_1^A \cap K_2^A$ é condição necessária e suficiente para a modularidade hierárquica.

Prova: \rightarrow (Somente se)

Sejam K_1^A e K_2^A duas linguagens tais que $L_m(S_1^A/G^A) = K_1^A$ e $L_m(S_2^A/G^A) = K_2^A$. Por hipótese $\overline{K_1^A} \cap \overline{K_2^A} = \overline{K_1^A \cap K_2^A}$.

Para qualquer $s \in L(S_1^A \odot S_2^A/G^A)$ tal que $\delta^A(q_0^A, s) = X_i \in \pi$, sejam $com_1 = (\gamma_1, \#_1) \in \Gamma_i$ e $com_2 = (\gamma_2, \#_2) \in \Gamma_i$, entradas de controle aplicadas pelos supervisores S_1^A e S_2^A respectivamente.

Por Γ -compatibilidade de $K_1^A \cap K_2^A$, para qualquer bloco $X_i \in \pi$ atingido em G^A , existe uma entrada de controle no alto nível $(\gamma, \#) \in \Gamma_i$, esta entrada de controle pode ser interpretada como a ação de um supervisor equivalente S_e^A tal que $L_m(S_e^A/G^A) = K_1^A \cap K_2^A$, que garante uma realização controlável e não bloqueante no baixo nível, seja $L_m(S/G)$ a implementação de $K_1^A \cap K_2^A$

K_2^A no baixo nível, segundo o Teorema 5.2 em (Torrice, 2003) $\Theta(L_m(S/G)) = K_1^A \cap K_2^A$

\leftarrow (Se)

Pelo Teorema 5.3 em (Torrice, 2003), $\theta(L_m(S/G))$ é Γ -compatível. □

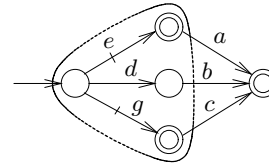
O teorema anterior garante que para cada cadeia de eventos da intersecção $\overline{K_1^A} \cap \overline{K_2^A}$ no alto nível existe uma entrada de controle que permite uma realização não bloqueante no baixo nível.

Existem condições suficientes para modularidade hierárquica em que o teste da Γ compatibilidade não será necessário. Para provar estas condições, previamente são necessárias algumas definições as quais são apresentadas a seguir.

Definição 2 (Fechamento de conjunção das estruturas de controle de G^A)

A estrutura de controle de G^A é dita ser fechada para a conjunção se

1. $(\gamma_1, N), (\gamma_2, \#) \in \Gamma_i \rightarrow \exists(\gamma_3, N) \in \Gamma_i$ tal que $\gamma_3 = \gamma_1 \cap \gamma_2$ para $\# = M, N$
2. $(\gamma_1, M), (\gamma_2, M) \in \Gamma_i \rightarrow \exists(\gamma_3, M) \in \Gamma_i$ tal que $\gamma_3 = \gamma_1 \cap \gamma_2$



$$\Gamma_i = \{(\{b\}, N), (\{a, b\}, M), (\{b, c\}, M), (\{a, b, c\}, M)\}$$

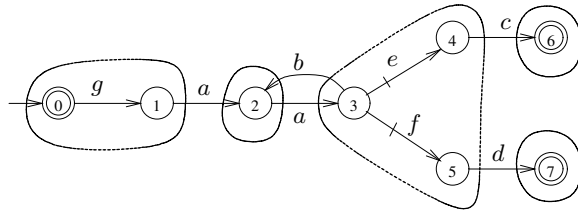
Figura 6: Análise de um bloco

Para o exemplo da agregação mostrada na Figura 6, observa-se que as estruturas de controle não satisfazem a condição de fechamento de conjunção, pois:

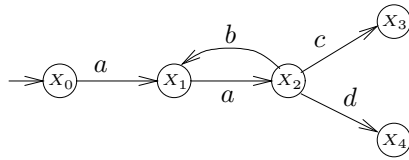
$$(\{a, b\}, M), (\{b, c\}, M) \in \Gamma_i, \text{ entretanto } (\{b\}, M) \notin \Gamma_i$$

Dados dois supervisores S_1^A e S_2^A tal que $L_m(S_1^A/G^A) = K_1^A$ e $L_m(S_2^A/G^A) = K_2^A$, o fechamento para conjunção não garante o não bloqueio da ação conjunta de ambos supervisores, conforme mostra o contra-exemplo a seguir:

Exemplo 2 Seja a planta mostrada na figura 7(a); na figura 7(b) apresenta-se o correspondente modelo agregado com estruturas de controle dependentes do estado.



(a) Planta

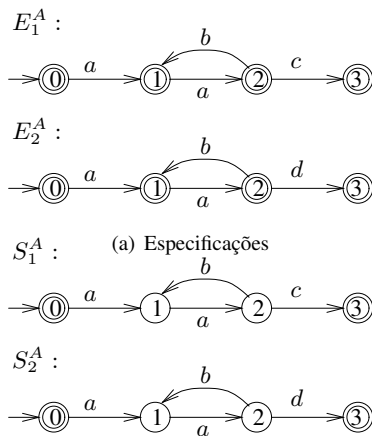


$$\begin{aligned} \Gamma_{X_0} &= \{(\{a\}, M)\} \\ \Gamma_{X_1} &= \{(\{a\}, N)\} \\ \Gamma_{X_2} &= \{(\{b\}, N), (\{b, c\}, N), (\{b, d\}, N), (\{b, c, d\}, N)\} \\ \Gamma_{X_3} &= \Gamma_{X_4} = \{(\emptyset, M)\} \end{aligned}$$

(b) Autômato agregado com Γ dependente do estado

Figura 7: Caso de estudo

Para as especificações E_1^A e E_2^A do alto nível apresentadas na figura 8(a), na figura 8(b) apresentam-se os autômatos das respectivas máxima linguagem Γ -compatível. Observa-se que as estruturas de controle são fechadas para conjunção, então para cada cadeia no nível agregado observada por S_1^A e S_2^A existe um comando com resultante da conjunção entre com_1 e com_2 que permite a evolução do sistema.



(b) Máxima linguagem realizável

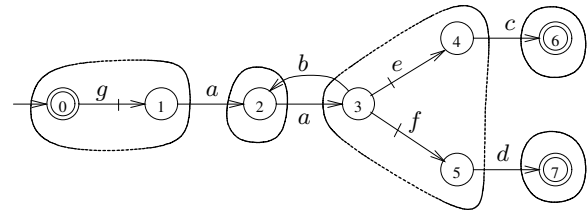
Figura 8: Especificações e máximas realizações

No caso, quando o sistema alcançar o bloco X_2 , S_1^A envia a entrada de controle $com_1 = (\{b, c\}, N)$ e S_2^A a entrada de controle $com_2 = (\{b, d\}, N)$, como resultado da conjunção tem-se que $com = (\{b\}, N)$ desabilitando os eventos c e d ; então observa-se que o sistema no nível agregado ficará bloqueado entre os estados X_1 e X_2 sem poder completar ne-

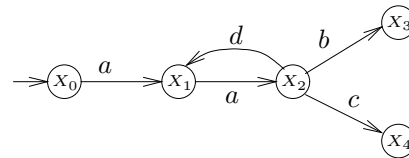
nhuma tarefa, correspondendo no baixo nível a um bloqueio nos estados 2 e 3.

O fechamento para conjunção das estruturas de controle não garante o não bloqueio, mesmo que a intersecção entre as linguagens realizáveis do nível agregado seja Γ -compatível. O exemplo a seguir ilustra este caso.

Exemplo 3 Considera-se a agregação da Figura 9.



(a) Planta



$$\begin{aligned} \Gamma_{X_0} &= \{(\emptyset, M), (\{a\}, M)\} \\ \Gamma_{X_1} &= \{(\{a\}, N)\} \\ \Gamma_{X_2} &= \{(\{b\}, N), (\{b, c\}, N), (\{b, d\}, N), (\{b, c, d\}, N)\} \\ \Gamma_{X_3} &= \Gamma_{X_4} = \{(\emptyset, M)\} \end{aligned}$$

(b) Autômato agregado com Γ dependente do estado

Figura 9: Caso de estudo

Para as mesmas especificações E_1 e E_2 do alto nível apresentadas na figura 8(a), neste caso os autômatos das respectivas máxima linguagens Γ -compatíveis resultam iguais aos da figura 8(b). Ao contrário do exemplo anterior (Exemplo 2), verifica-se que a intersecção entre as máxima linguagens Γ -compatíveis é Γ -compatível, mas isto não implica que estas linguagens serão modulares, isto é, $\overline{L_m(S_1)} \cap \overline{L_m(S_2)} \neq \overline{L_m(S_1) \cap L_m(S_2)}$. Portanto acontece o mesmo problema que no exemplo anterior.

O teorema a seguir garante que, dado um modelo abstrato com estruturas de controle fechada para conjunção, a intersecção de duas especificações realizáveis e modulares no alto nível é Γ -compatível.

Teorema 2 Dados dois supervisores S_1^A e S_2^A tais que $L_m(S_1^A/G^A) = K_1^A$, $L_m(S_2^A/G^A) = K_2^A$, com $\overline{K_1^A} \cap \overline{K_2^A} = \overline{K_1^A \cap K_2^A}$. O fechamento de conjunção das estruturas de controle do nível gerencial implica que $K_1^A \cap K_2^A$ é Γ -compatível.

Prova: Sejam $E_1^A, E_2^A \subseteq L(G^A)$ duas linguagens quaisquer de alto nível, tais que $\sup \mathcal{CM}(E_1^A, L(G^A)) = K_1^A$ e $\sup \mathcal{CM}(E_2^A, L(G^A)) = K_2^A$. Por hipótese $\overline{K_1^A} \cap \overline{K_2^A} = \overline{K_1^A \cap K_2^A}$

Denota-se $L(S_1^A \odot S_2^A / G^A)$ como a implementação conjunta dos supervisores do alto nível. Para qualquer $s \in L(S_1^A \odot S_2^A / G^A)$ tal que $\delta^A(q_0^A, s) = X_i \in \pi$, sejam, $com_1 = (\gamma_1, \#_1) \in \Gamma_i$ e $com_2 = (\gamma_2, \#_2) \in \Gamma_i$, entradas de controle aplicadas pelos supervisores de alto nível. Por fechamento para conjunção das estruturas de controle $\exists com = (\gamma_3, \#_3) \in \Gamma_i$, tal que $\gamma_3 = \gamma_1 \cap \gamma_2$ e $\#_3 = M$ se $\#_1 = M$ e $\#_2 = M$, outro caso $\#_3 = N$.

Então o comando com pode ser interpretado como a ação de um supervisor equivalente S_e^A tal que $L_m(S_e^A / G^A) = L_m(S_1^A \odot S_2^A / G^A) = \sup \mathcal{CM}(E_1^A, L(G^A)) \cap \sup \mathcal{CM}(E_2^A, L(G^A))$.

Seja $s \in K_1^A \cap K_2^A$; obviamente $s \in K_1^A$ e $s \in K_2^A$, portanto $\exists (\gamma_1, M), (\gamma_2, M)$ tais que $\gamma_j \cap \Sigma_{L(G^A)}(s) = \Sigma_{K_j^A}(s)$ para $j = 1, 2$. Tem-se $\gamma_1 \cap \gamma_2 \cap \Sigma_{L(G^A)}(s) = \Sigma_{K_1^A}(s) \cap \Sigma_{K_2^A}(s) = \Sigma_{\overline{K_1^A} \cap \overline{K_2^A}}(s) = \Sigma_{\overline{K_1^A \cap K_2^A}}(s) = \Sigma_{K_1^A \cap K_2^A}(s)$. Pelo fechamento para conjunção $(\gamma_1 \cap \gamma_2, M) \in \Gamma(s)$.

Seja agora $s \in \overline{K_1^A} - K_1^A$ e $s \in \overline{K_2^A}$; pode-se facilmente deduzir que $s \in \overline{K_1^A} \cap \overline{K_2^A}$ e $s \notin K_1^A \cap K_2^A$, ou seja, $s \in \overline{K_1^A} \cap \overline{K_2^A} - K_1^A \cap K_2^A = \overline{K_1^A \cap K_2^A} - K_1^A \cap K_2^A$. Nesse caso $\exists (\gamma_1, N), (\gamma_2, \#)$ tais que $\gamma_j \cap \Sigma_{L(G^A)}(s) = \Sigma_{K_j^A}(s)$ para $j = 1, 2$. Pelo fechamento para conjunção $(\gamma_1 \cap \gamma_2, N) \in \Gamma(s)$. Raciocínio similar para $s \in \overline{K_2^A} - K_2^A$ e $s \in \overline{K_1^A}$ permite concluir que $K_1^A \cap K_2^A$ é Γ -compatível. Portanto, $K_1^A \cap K_2^A \subseteq \sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A))$.

Para a inclusão reversa usamos o fato que $\sup \mathcal{CM}$ é monótono i.e., se $E \subseteq E'$, $\sup \mathcal{CM}(E, L) \subseteq \sup \mathcal{CM}(E', L)$. Como $E_1^A \cap E_2^A \subseteq E_1^A$, isto implica que $\sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A)) \subseteq K_1^A$, similarmente $\sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A)) \subseteq K_2^A$. Assim, $K_1^A \cap K_2^A \supseteq \sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A))$.

Portanto $K_1^A \cap K_2^A = \sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A))$. □

Decorrente deste teorema, o corolário a seguir garante que, dado um modelo abstrato com estruturas de controle fechada para conjunção, duas especificações realizáveis e modulares no alto nível poderão ser implementadas no baixo nível sem nenhum conflito ou perda de otimalidade.

Corolário 3 Dados dois supervisores S_1^A e S_2^A tais que $L_m(S_1^A / G^A) = K_1^A$, $L_m(S_2^A / G^A) = K_2^A$, com $\overline{K_1^A} \cap \overline{K_2^A} = \overline{K_1^A \cap K_2^A}$. O fechamento de conjunção das estruturas de controle do nível gerencial é condição suficiente para que a modularidade hierárquica seja satisfeita.

Prova: Pelo teorema 2 sabe-se que $K_1^A \cap K_2^A$ é Γ -compatível, então é possível uma implementação $L_m(S/G)$

realizável e não bloqueante no baixo nível, segundo o Teorema 5.2 em (Torrico, 2003) $\theta(L_m(S/G)) = \sup \mathcal{CM}(E_1^A \cap E_2^A, L(G^A))$ □

Os resultados obtidos serão exemplificados na seção a seguir.

4 EXEMPLO DE APLICAÇÃO

Esta abordagem será ilustrada pelo desenvolvimento de vários supervisores hierárquicos para uma linha de transferência apresentada por Wonham (1998) no controle hierárquico tradicional. A linha de transferência consiste de duas máquinas M_1 e M_2 mais uma unidade de teste TU ligadas por dois buffers B_1 e B_2 na seqüência M_1, B_1, M_2, B_2, TU como mostrado na Fig. 10. A peça testada por TU pode ser aceita ou rejeitada; se for aceita, esta é liberada do sistema; senão volta para o buffer B_1 para re-processamento por M_2

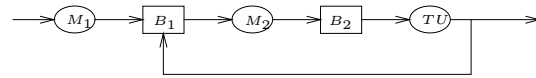


Figura 10: Linha de Transferência.

Os modelos dos componentes do sistema são mostrados na Fig. 11(a).

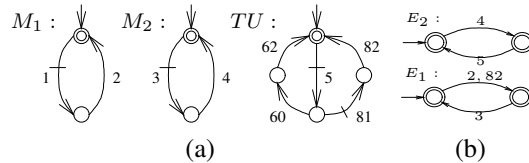


Figura 11: (a) Modelo dos componentes do sistema e (b) especificações para os buffers.

Os eventos controláveis 1 e 3 indicam o início de operação das máquinas M_1 e M_2 , respectivamente, e os eventos não controláveis 2 e 4 indicam o fim de operação das mesmas. O início da unidade de teste é representado pelo evento controlável 5 e os sinais de decisão de “aceito” ou “rejeitado” são representados pelos eventos 60 e 81 respectivamente. Caso a peça seja aceita pela unidade de teste, esta é enviada para fora do sistema (evento 62), caso contrário retorna para o buffer B_1 (evento 82). Considera-se os buffers de capacidade unitária. Na Fig. 11(b) mostra-se a especificação para não overflow e não underflow dos buffers. A Fig. 12 apresenta o modelo de uma solução a este problema, e esta é adotada como planta G para tratar do problema de controle hierárquico modular.

No alto nível, a preocupação dos gerentes atribui-se ao controle da relação entre peças de entrada e peças aceitas ou recusadas. Portanto, considera-se como eventos relevantes os eventos relacionados com este fato (1, 60, 81). Com es-

tes dados constrói-se uma agregação e o resultado conjuntamente com os padrões de controle são mostrados na Fig. 12. Observa-se que o bloco X_0 apresenta um não determinismo, motivo pelo qual o evento que produz não determinismo, 1 é instanciado em $1'$ e $1''$.

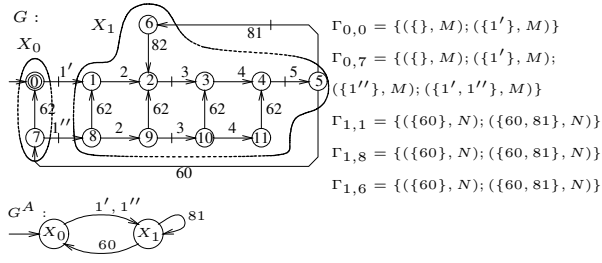


Figura 12: Planta e sua agregação.

Observa-se que o bloco X_0 do autômato agregado (Fig. 12) apresenta conjuntos de estruturas de controle diferentes para os dois estados de entrada 0 e 7 ($\Gamma_{0,0} \neq \Gamma_{0,7}$), motivo pelo qual o bloco X_0 é dividido em dois, um associado a $\Gamma_{0,0}$ e outro a $\Gamma_{0,7}$, representado pelos estados Y_0 e Y_2 na Fig. 13. Já o bloco X_1 , para todas as entradas, apresenta um mesmo conjunto de estruturas de controle ($\Gamma_{1,1} = \Gamma_{1,1} = \Gamma_{1,6}$), sendo mantido tal como está. Na Fig. 13, X_1 está representado por Y_1 .

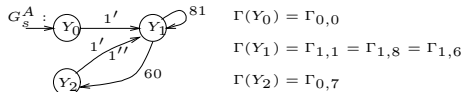


Figura 13: Planta hierárquica com estruturas de controle dependentes do estado.

Pode-se observar que as estruturas de controle de G_s^A são fechadas para conjunção.

De modo a ilustrar a condição de modularidade hierárquica consideram-se duas especificações para o sistema (Fig 14). A primeira especificação (E_1) indica que, toda vez que uma mesma peça for recusada duas vezes seguidas pela unidade de teste, o sistema deve parar. A segunda especificação (E_2) indica que se duas peças contíguas que entraram no sistema forem recusadas pela unidade de teste mesmo que seja só por uma vez, então uma parada do sistema é também forçada.

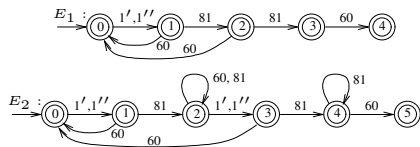


Figura 14: Especificações.

Continuando, obtêm-se as linguagens alvo e aplica-se o Algoritmo para síntese da máxima linguagem Γ -compatível apresentado para sistemas com estruturas de controle dependentes do estado (Cury et al., 2004). Os resultado são mostrados na Fig. 15.

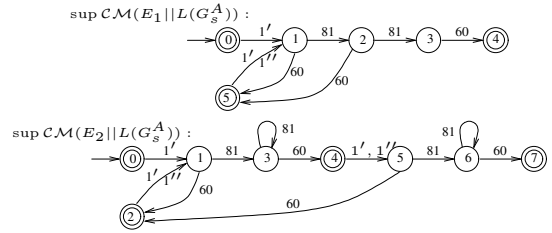


Figura 15: Máximas linguagens Γ -compatíveis.

Sejam $\sup \mathcal{CM}(E_1 || L(G_s^A)) = K_1^A$ e $\sup \mathcal{CM}(E_2 || L(G_s^A)) = K_2^A$, pode ser verificada a modularidade de alto nível, i.e. $K_1^A \cap K_2^A = \overline{K_1^A} \cap \overline{K_2^A}$, conseqüentemente é possível uma implementação realizável e não bloqueante no baixo nível. A seguir mostra-se uma parte da implementação no baixo nível resultante da ação conjunta de ambos supervisores de alto nível.

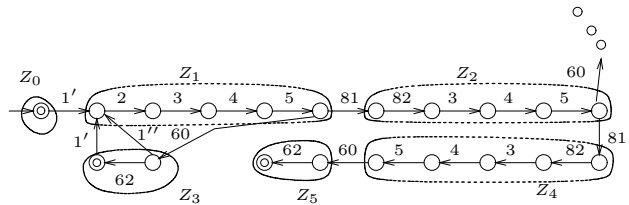


Figura 16: Implementação no baixo nível

A função do operador de conjunção será interpretar as entradas de controle vindos de ambos supervisores S_1^A e S_2^A e traduzir num comando único. Para o caso do exemplo, no bloco Z_5 (Bloco Y_2 atingido em G_s^A) da implementação, o supervisor S_1^A aplica a entrada de controle $(\{\}, M)$ e o supervisor S_2^A aplica a entrada de controle $(\{1', 1''\}, M)$, finalmente o operador de conjunção traduz as duas entradas e manda como resposta a entrada de controle $(\{\}, M)$.

5 DISCUSSÃO

O controle hierárquico modular foi aplicado com sucesso ao modelo proposto. Neste modelo a condição de fechamento para conjunção das estruturas de controle e a modularidade de alto nível garantem a existência de um supervisor equivalente à ação conjunta de dois supervisores no alto nível, permitindo assim um comportamento consistente entre os níveis de hierarquia e uma implementação não bloqueante no baixo nível. A abordagem proposta combina os ganhos obtidos pela agregação e pela modularidade. A análise da complexidade computacional é o próximo passo a ser desenvolvido.

AGRADECIMENTOS

César Torrico foi apoiado pela CAPES. José E. R. Cury foi apoiado em parte pelo CNPq, processos 3000953/93-3 e PRONEX 015/98.

REFERÊNCIAS

- Caines, P. e Hubbard, P. (1998). A state aggregation approach to hierarchical supervisory control with applications to a transfer-line example, *Proceedings of the WODES 98*, Cagliari, Italy, pp. 2–7.
- Cury, J., Torrico, C. e da Cunha, A. (2001). A new approach for supervisory control of discrete event systems, *Proceedings of the European Control Conference 2001*.
- Cury, J., Torrico, C. e da Cunha, A. (2004). Supervisory control of discrete event systems with flexible marking, *Aceito para publicação em 2004 na revista European Journal of Control*.
- Ramadge, P. e Wonham, W. (1989). The control of discrete event systems, *Proceeding of the IEEE* **77**(1): 81–98.
- Torrico, C. (2003). *Controle Supervisório Hierárquico de Sistemas a Eventos Discretos: Uma Abordagem Baseada na Agregação de Estados*, Tese(doutorado), Pós-Graduação em Engenharia Elétrica - Universidade Federal de Santa Catarina, Fpolis - SC.
- Torrico, C. e Cury, J. (2002). Hierarchical supervisory control of discrete event systems based on state aggregation, *Proceedings of the 15th IFAC World Congress*.
- Wong, K. e Wonham, W. (1996). Hierarchical control of discrete-event systems, *Discrete Event Dynamical Systems* **6**: 241–273.
- Wonham, W. (1998). *Notes on Control of Discrete-Event Systems. Course Notes for ECE 1636F/1637S*, Revision 98.09.01.
- Wonham, W. e Ramadge, P. (1988). Modular supervisory control of discrete event systems, *Mathematics of control, Signals and Systems* **1**(1): 13–30.
- Zhong, H. e Wonham, W. (1990). On the consistency of hierarchical supervision in discrete-event systems, *IEEE Transactions on Automatic Control* **35**(10): 1125–1134.