

UNIVERSIDADE FEDERAL DE SANTA CATARINA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

"UMA ARQUITETURA FÍSICA PARA A IMPLEMENTAÇÃO DO CONTROLE
SUPERVISÓRIO DE SISTEMAS A EVENTOS DISCRETOS"

DISSERTAÇÃO SUBMETIDA À UNIVERSIDADE FEDERAL DE SANTA CATARINA PARA
A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA ELÉTRICA

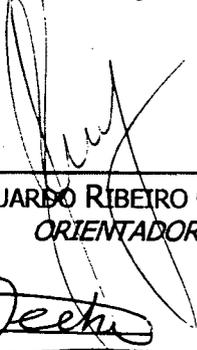
ELMO DOMINGUES MARTINS

FLORIANÓPOLIS, ABRIL DE 1999

"UMA ARQUITETURA FÍSICA PARA A IMPLEMENTAÇÃO DO CONTROLE
SUPERVISÓRIO DE SISTEMAS A EVENTOS DISCRETOS"

ELMO DOMINGUES MARTINS

ESTA DISSERTAÇÃO FOI JULGADA ADEQUADA PARA A OBTENÇÃO DO TÍTULO DE
MESTRE EM ENGENHARIA, ESPECIALIDADE *ENGENHARIA ELÉTRICA*, ÁREA DE
CONCENTRAÇÃO *INFORMÁTICA INDUSTRIAL*, APROVADA EM SUA FORMA FINAL
PELO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA.

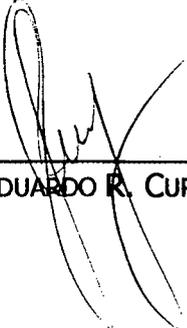


PROF. JOSÉ EDUARDO RIBEIRO CURY, DR. D'ÉTAT
ORIENTADOR

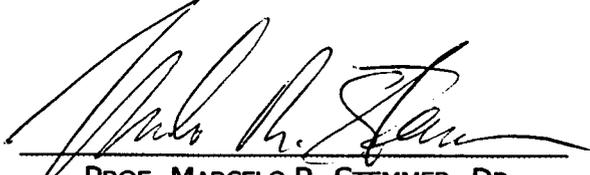


PROF. ILDEMAR CASSANA DECKER, D. SC.
COORDENADOR DO CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

BANCA EXAMINADORA:



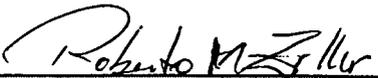
PROF. JOSÉ EDUARDO R. CURY, DR. D'ÉTAT



PROF. MARCELO R. STEMMER, DR.



PROF. JEAN-MARIE FARINES, DR.ING.



PROF. ROBERTO M. ZILLER, M.ENG.

*"VOCÊ VÊ COISAS E DIZ: POR QUE?
EU, PORÉM, SONHO COISAS QUE NÃO EXISTEM E DIGO: POR QUE NÃO?"
GEORGE BERNARD SHAW*

AGRADECIMENTOS

Ao professor José Cury, pela orientação, apoio e compreensão. Deixamos para trás a relação entre professor e aluno, segue adiante uma grande amizade.

Ao professor Augusto Brucciapaglia, Coordenador do Curso de Engenharia e Controle de Automação Industrial, pelo incentivo que me concedeu desde a graduação até agora.

Aos amigos Emerson, Junior e Henrique, pelas discussões e auxílio prestados durante todo este trabalho.

Ao LCMI, na figura do professor Jean-Marie Farines, por ter me possibilitado trabalhar em excelentes condições, a fim de proporcionar um trabalho de qualidade.

Aos meus pais, Edésio e Dorcas, e aos meus irmãos, Edson e Elcio, que me deram a base sólida para que hoje eu pudesse concretizar mais essa etapa.

A Deus, "Porque dEle e por meio dEle e para Ele são todas as coisas. A Ele, pois, a glória eternamente. Amém" *Romanos 11:36*

ÍNDICE

RESUMO	VII
ABSTRACT	VIII
CAPÍTULO 1 - INTRODUÇÃO	9
CAPÍTULO 2 - TEORIA CLÁSSICA DE CONTROLE DE SISTEMAS A EVENTOS DISCRETOS: CONCEITUAÇÃO	13
2.1. A Representação de um Sistema a Eventos Discretos por um Gerador	13
2.2. Composição de SED's: Produto Síncrono	15
2.2.1. <i>Projeção Natural</i>	15
2.2.2. <i>Projeção Inversa</i>	16
2.2.3 <i>Produto Síncrono</i>	16
2.3. O Controle	17
2.3.1 <i>Entrada de Controle</i>	18
2.4. Supervisores	19
2.5. Controlabilidade	21
2.6. Condições de existência do supervisor	23
2.6.1 <i>Formulação em termos de linguagens marcadas</i>	23
2.7. Máxima Linguagem Controlável	26
2.7.1 <i>Existência da máxima linguagem controlável</i>	26
CAPÍTULO 3 - A TEORIA CLÁSSICA: PROBLEMA ENCONTRADO NUM EXEMPLO DE APLICAÇÃO	30
3.1. A Célula de Manufatura	32
3.1.2 <i>A modelagem utilizando a teoria clássica</i>	32
3.2. Um novo modelo para o sistema "estação de espera + buffer"	38
CAPÍTULO 4 - EXTENSÃO À TEORIA CLÁSSICA: A NOÇÃO DE EVENTOS FORÇÁVEIS	42
4.1. O conjunto Σ de eventos	42
4.2. O conjunto Γ de entradas de controle	43
4.3. Linguagens Controláveis	43
4.4. Solução para o sistema "estação de espera + buffer" utilizando eventos forçáveis	45
CAPÍTULO 5 - PROPOSTA DE UMA ARQUITETURA FÍSICA PARA IMPLEMENTAÇÃO DE CONTROLE	48
5.1 Introdução	48
5.2 Proposta para uma Arquitetura Física	50

5.2.1 O Supervisor.....	51
5.2.2 O Módulo Intermediário.....	52
CAPÍTULO 6 - EXEMPLOS DE APLICAÇÃO	56
6.1 Introdução.....	56
6.2 Uma Estação de Inspeção.....	56
6.3 Uma Estação de Rejeição de Pallets.....	59
6.4. Uma Estação de Carregamento de Pallets.....	62
CONCLUSÕES E PERSPECTIVAS.....	65
REFERÊNCIAS BIBLIOGRÁFICAS	67

RESUMO

A partir de 1987 muito se tem discutido sobre o controle de Sistemas a Eventos Discretos. Vários trabalhos sobre este assunto foram publicados, mas poucos tratam, de forma clara, da aplicação prática da Teoria de Controle de Sistemas a Eventos Discretos.

Este trabalho mostra que, ao se aplicar a Teoria Clássica, introduzida por Wonham e Ramadge, sobre um modelo cujos eventos não possuam determinadas características, definidas pela própria Teoria Clássica, os resultados obtidos podem não ser coerentes com a realidade. A modelagem de sistemas é bastante discutida, e a noção de eventos forçáveis é apresentada, a qual serve como base para a solução de controle quando o modelo apresentado não está adequado aos moldes da Teoria Clássica.

Baseado nas discussões sobre modelagem e na noção de eventos forçáveis introduzida por Golaszewski e Ramadge, propõe-se uma arquitetura física para o controle supervisorio de Sistemas a Eventos Discretos. Tal arquitetura possui 4 módulos, sendo que um deles, o Módulo Intermediário, representa o papel de um agente controlador, o qual possui funções de decisão, capazes de determinar ou não a execução de comandos sobre a planta. Neste caso, o Módulo Intermediário também é capaz de gerar eventos, o que difere da Teoria Clássica, que afirma que somente a planta é responsável por fazê-lo.

ABSTRACT

The control of Discrete Event Systems has been deeply discussed since 1987. Lots of works have been written and published, but just a few of them give us a clear vision of the practice of the Discrete Event Systems Control Theory.

This dissertation shows that the application of Wonham and Ramadge's Classical Theory can present wrong results, when the model proposed does not follow some rules, imposed by this Theory. The modeling of systems is well discussed, and the forced events theory is presented, which can help on the control solution for those models that do not respect the Classical Theory's rules.

Based on the modeling discussions and on the forced events theory, a physical architecture is proposed to the implementation of Discrete Event Systems supervisory control. Such architecture has four modules, and one of them plays the role of a controller agent, being responsible for making decisions on the execution of commands at the plant level. In this case, this module is also an event generator, when the Classical Theory says, on the contrary, that the plant is the only one responsible for making events occur.

CAPÍTULO 1 - INTRODUÇÃO

A Teoria de Controle de Sistemas a Eventos Discretos surgiu a partir da necessidade de se estudar matematicamente alguns sistemas existentes no dia-a-dia. As células de manufatura, as redes de computadores, os sistemas de gerenciamento de base de dados, todos estes são exemplos clássicos de Sistemas a Eventos Discretos.

Um Sistema a Eventos Discretos (*SED*) é, por definição, um sistema dinâmico que evolui de acordo com a ocorrência assíncrona de *eventos*, os quais modificam o estado do sistema [KG95]. Por exemplo, no caso clássico de um sistema de manufatura, um evento poderia ser a chegada de uma peça para ser usinada, de um carro para ser pintado, uma pane em um equipamento, etc.

Os estudos sobre esta classe de sistemas se intensificaram a partir do final da década de 80, quando Ramadge e Wonham apresentaram a teoria de *Controle Supervisório* para Sistemas a Eventos Discretos. Nesta teoria, os autores apresentam métodos e ferramentas para a modelagem, análise e síntese de controladores para este tipo de sistemas [RW87].

A partir desta publicação, muito se tem discutido sobre o controle de SED's. Inúmeros trabalhos foram realizados desde então, o que mostra a importância dada a este assunto. Posteriormente, extensões à teoria de Wonham e Ramadge introduziram as noções de observabilidade, modularidade, controle hierárquico, controle descentralizado, etc [LW88][WR88][RW92][Zh87].

Esta ênfase no estudo dos SEDs é explicada não somente pelo grande número de sistemas deste tipo encontrados na prática, mas também pelo aumento da complexidade dos mesmos. Isto fez surgir a necessidade de estudos mais detalhados, na busca de soluções de controle mais adequadas, eficazes e seguras.

Muito se tem escrito sobre a teoria de controle aplicada aos sistemas a eventos discretos, porém são poucos os trabalhos que mostram a aplicação prática desta teoria, como, por exemplo, a implementação de malhas ou arquiteturas físicas de controle.

Nota-se que, na aplicação prática desta teoria, é necessário que alguns pontos sejam bem definidos, como:

- A relação existente entre os eventos que ocorrem fisicamente e aqueles tratados a nível de supervisor (controlador);
- As decisões que devem ser tomadas para tratar o não-determinismo;
- A lógica utilizada pelo supervisor para tratar a sequência de eventos que vêm da planta;
- A execução em tempo real;

Todos estes pontos surgem a partir do momento em que se deseja controlar um sistema real. Tais sistemas possuem características próprias, que adicionam ao problema de síntese de controle uma série de questões normalmente não tratadas na teoria.

O estudo dos SEDs baseia-se sempre num modelo matemático que descreve o sistema real. Este modelo permite não somente compreender formalmente o que ocorre na prática, mas também possibilita a utilização de técnicas de análise avançadas, com o objetivo de se construir complexos mecanismos de controle.

Devido à diversidade de aplicações dos SEDs, várias técnicas de modelagem foram desenvolvidas. Cada uma delas aplica-se ao estudo de problemas específicos. Uma destas técnicas, comumente aplicada, ignora a ação explícita do tempo, levando em conta principalmente a sequência na qual ocorrem os eventos. O modelo resultante desta técnica é denominado *modelo lógico*. Os *Autômatos* e as *Redes de Petri* são exemplos desta técnica de modelagem (*figuras 1.1 e 1.2*).

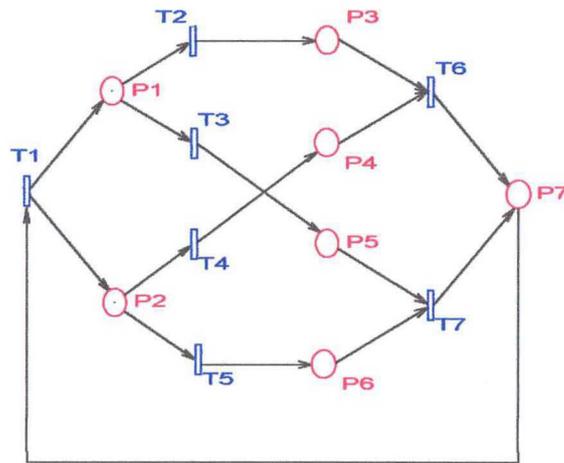


Figura 1.1 Rede de Petri

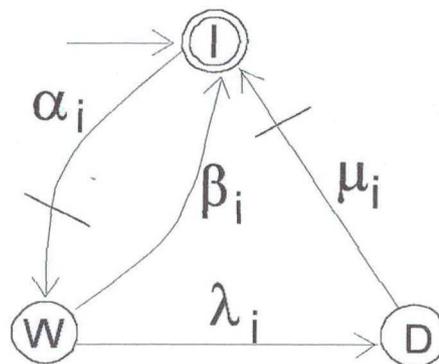


Figura 1.2 Autômato

A modelagem por autômatos tem sido bastante utilizada no estudo dos SEDs,

principalmente quando se trabalha sob a visão da Teoria de Controle. Durante todo este trabalho, esta será a técnica de modelagem utilizada.

O objetivo principal da Teoria de Controle é fazer com que um determinado sistema se mantenha, através de ações de controle, dentro de um comportamento desejado. Mas, para que isso aconteça na prática, ou seja, para que a solução encontrada teoricamente seja aplicável ao sistema real, é fácil notar o grau de importância que assume a forma como o sistema é modelado. É imprescindível que a modelagem reflita da melhor forma possível o comportamento real do sistema.

Num sistema real a eventos discretos controlado, existem no mínimo 2 elementos básicos: a *planta* (sistema a ser controlado) e o *supervisor* (controlador).

De um ponto de vista funcional, a planta basicamente recebe ordens de execução de tarefas e envia de volta informações através de sensores.

Assim, os eventos no nível físico, ou representam *comandos específicos aos atuadores* (ex.: mover a garra do robô do ponto "a" para o ponto "b"), ou representam a *leitura de sensores* (ex.: sensor 2 sensibilizado por uma peça).

O supervisor, por outro lado, necessita *conhecer o comportamento da planta* para que possa atuar sobre o sistema em momentos bem determinados. Este comportamento é descrito no nível de supervisão através de um modelo ou, formalmente, um gerador da planta. Neste modelo, encontram-se os possíveis estados da planta, ligados por eventos que modificam estes estados.

Portanto, de um lado existem eventos que acontecem fisicamente na planta, que são comandos aplicados a atuadores ou correspondem à leitura de sensores. De outro lado, no nível de supervisão, existem eventos descritos no modelo da planta, que fornecem ao supervisor a dinâmica do comportamento do sistema.

Para que o todo o sistema (planta + supervisor) funcione harmoniosamente, deve existir um mapeamento entre os eventos que acontecem no nível físico, e os eventos descritos no modelo da planta, no nível de supervisão. Uma questão que se coloca é a seguinte: "Qual a relação entre os eventos descritos no modelo da planta e os eventos que fisicamente ocorrem na planta?".

Este mapeamento depende da forma como a planta é modelada.

O objetivo deste trabalho, é fornecer uma solução clara e bem definida para a modelagem de sistemas reais utilizando autômatos, de forma a possibilitar a construção de um modelo que seja realizável na prática, e também propor uma arquitetura física que suporte as idéias aqui apresentadas, e que seja compatível com requisitos (lógicos, temporais, etc) impostos pelo sistema [ME97].

O Capítulo 2 fornecerá uma breve introdução à Teoria de Controle de Sistemas a Eventos Discretos.

No Capítulo 3 será apresentada uma solução de controle para um sistema bastante simples, utilizando a teoria de Wonham e Ramadge. Isto servirá para mostrar que, caso o modelo não esteja de acordo com a Teoria Clássica, os resultados obtidos poderão não ser coerentes com a prática.

O Capítulo 4 apresentará uma outra alternativa de modelagem para o sistema do Capítulo 3, apresentando a extensão feita por Golaszewski e Ramadge [GR87], que inclui a noção de Eventos Forçáveis.

No Capítulo 5 será apresentada uma abordagem geral, incluindo uma metodologia para a mesma, e uma arquitetura que implemente a solução de controle.

No Capítulo 6 serão apresentados alguns exemplos e discutida a forma de modelagem destes sistemas.

Finalmente, apresenta-se as conclusões e perspectivas para que se possa dar prosseguimento a este trabalho.

CAPÍTULO 2 - TEORIA CLÁSSICA DE CONTROLE DE SISTEMAS A EVENTOS DISCRETOS: CONCEITUAÇÃO

Este capítulo apresentará uma síntese da Teoria Clássica introduzida por Wonham e Ramadge para o controle de Sistemas a Eventos Discretos. Não se pretende ser exaustivo na apresentação da Teoria Clássica, mas sim introduzir os principais conceitos sobre os quais este documento se baseia, e que servirão de base para a compreensão dos capítulos posteriores. Um estudo mais profundo pode ser feito a partir das seguintes publicações: [RW87][KG95][Cu96].

2.1. A Representação de um Sistema a Eventos Discretos por um Gerador

Um Gerador é, por definição, uma quintupla:

$$G = \{ \Sigma, Q, \delta, Q_m, q_0 \}$$

Onde:

Σ é um alfabeto (conjunto de eventos);

Q é um conjunto não vazio de estados;

$\delta: \Sigma \times Q \rightarrow Q$ uma função parcial de transição de estados;

$Q_m \subseteq Q$ é um conjunto de estados marcados.

Define-se a linguagem gerada por G como:

$$L(G) \triangleq \left\{ s \in \Sigma^* \mid \hat{\delta}(s, q_0) \text{ esteja definida} \right\}$$

Onde:

Σ^* é o conjunto de todas as cadeias finitas de eventos sobre o alfabeto Σ ,
e

$\hat{\delta}: \Sigma^* \times Q \rightarrow Q$ corresponde a extensão natural de δ a cadeias de eventos, definida como:

$$\hat{\delta}(s\sigma, q) = \delta(\sigma, \hat{\delta}(s, q))$$

$$\hat{\delta}(\sigma, q) = \delta(\sigma, q)$$

$$\hat{\delta}(\varepsilon, q) = q$$

$$\sigma \in \Sigma,$$

$$s \in \Sigma^*,$$

$$q \in Q$$

Define-se a linguagem marcada de G como:

$$L_m(G) \triangleq \left\{ s \in L(G) / \hat{\delta}(s, q_0) \in Q_m \right\}$$

Um gerador, da forma como definido acima, pode representar o funcionamento de um Sistema a Eventos Discretos. A linguagem $L(G)$ representa as possíveis seqüências de eventos geradas pelo sistema, e a linguagem $L_m(G)$ corresponde àquelas seqüências que representam tarefas completas no sistema.

O Gerador, conforme a Teoria de Controle, assume o papel da *planta* ou sistema a controlar.

Como exemplo, na figura 2.1 tem-se o modelo de um usuário de um recurso.

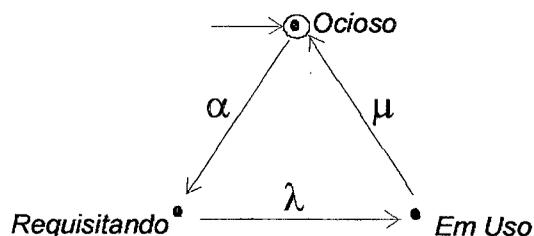


figura 2.1 – modelo de um usuário de um recurso

$$\Sigma = \{\alpha, \lambda, \mu\}$$

$q_0 = \text{ocioso}$

$$L(G) = (\alpha\lambda\mu)^* (\varepsilon + \alpha + \alpha\lambda)$$

$$L_m(G) = (\alpha\lambda\mu)^*$$

Na figura 2.2, um outro exemplo clássico: o filósofo que bebe.

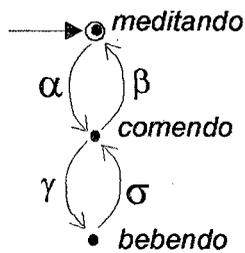


figura 2.2 – o modelo de “o filósofo que bebe”

$$\Sigma = \{\alpha, \beta, \gamma, \sigma\}$$

$$L(G) = (\alpha(\gamma\sigma)^* \beta)^* (\varepsilon + \alpha + \alpha\gamma + \alpha\gamma\sigma)$$

$$L_m(G) = (\alpha(\gamma\sigma)^* \beta)^*$$

2.2. Composição de SED's: Produto Síncrono

2.2.1. Projeção Natural

Sejam Σ_i e Σ , com $\Sigma_i \subseteq \Sigma$, define-se a projeção natural de Σ^* em Σ_i^* como:

$$P_i: \Sigma^* \rightarrow \Sigma_i^*$$

$$P_i(\sigma) = \begin{cases} \varepsilon & \text{se } \sigma \notin \Sigma_i \\ \sigma & \text{se } \sigma \in \Sigma_i \end{cases}$$

$$P_i(s\sigma) = P_i(s)P_i(\sigma) \quad s \in \Sigma^*, \sigma \in \Sigma$$

Estendendo-se para as linguagens, tem-se:

$$P_i L = L_i \triangleq \{s_i \in \Sigma_i^* / s_i \in P_i s, s \in L\}$$

2.2.2. Projeção Inversa

Sejam Σ_i e Σ , com $\Sigma_i \subseteq \Sigma$ e $L_i \subseteq \Sigma_i^*$.

Define-se

$$P_i^{-1} L_i \triangleq \{s \in \Sigma^* / P_i s \in L_i\}$$

2.2.3 Produto Síncrono

Define-se o produto síncrono $L_1 \parallel_s L_2 \subseteq \Sigma^*$ como sendo:

$$L_1 \parallel_s L_2 = P_1^{-1} L_1 \cap P_2^{-1} L_2$$

Interpretando-se esta definição em termos de Geradores, sendo $L_1 = L(G_1)$ e $L_2 = L(G_2)$, diz-se que a linguagem gerada pelo produto síncrono dos dois geradores representa a ação cooperativa entre eles, com a sincronização dos eventos comuns aos alfabetos Σ_1 e Σ_2 .

Como exemplo, mostra-se na figura 2.4 a composição síncrona entre os geradores da figura 2.3.

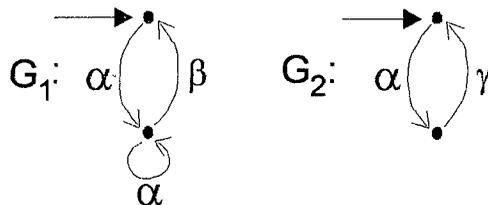


figura 2.3 – dois geradores quaisquer

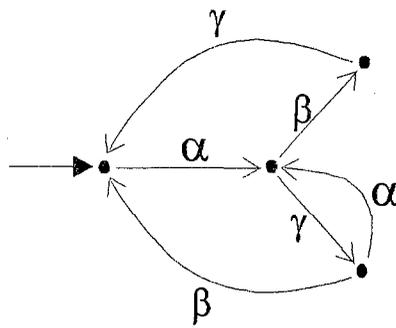


figura 2.4 – resultado da composição síncrona entre os geradores da figura 2.3

2.3. O Controle

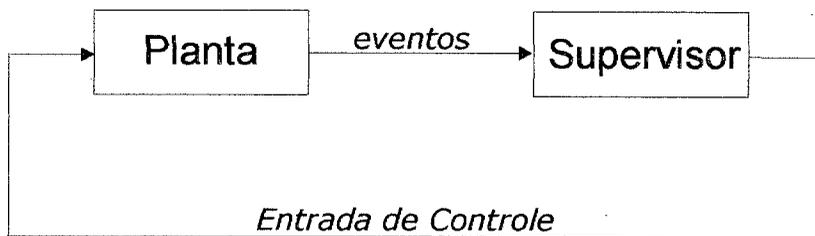


figura 2.5 – Desenho esquemático de um Sistema a Eventos Discretos Controlado

A noção de controle é introduzida particionando-se o conjunto de eventos como:

$$\Sigma = \Sigma_u \cup (\Sigma - \Sigma_u)$$

Onde:

Σ_u : conjunto dos eventos *não controláveis*, sobre os quais o supervisor não tem influência;

$\Sigma - \Sigma_u$ (Σ_c): conjunto dos eventos *controláveis*, que podem ser inibidos pelo supervisor;

O supervisor (controlador) restringe o comportamento da planta desabilitando dinamicamente alguns dos eventos controláveis, a cada execução de um evento da planta.

2.3.1 Entrada de Controle

A entrada de controle de um SED G é um subconjunto

$$\gamma \subset \Sigma / \Sigma_u \subset \gamma$$

Ou seja, caso $\sigma \in \gamma$, significa que σ está autorizado a ocorrer, ao passo que se $\sigma \notin \gamma$, este evento está inibido de ocorrer. O fato de $\Sigma_u \subset \gamma$ significa que o supervisor não pode atuar sobre os eventos não controláveis, ou seja, estes eventos não podem ser desabilitados pelo supervisor.

Para exemplificar, seja o autômato da figura 2.6, com $\Sigma_u = \{\beta\}$ e $\Sigma_c = \{\alpha, \mu\}$.

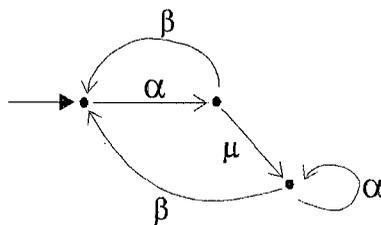


figura 2.6 – Um autômato qualquer

Seja o conjunto de entradas de controle descrito por:

$$\Gamma = \{\gamma_1, \gamma_2, \gamma_3\}$$

$$\gamma_1 = \{\alpha, \beta\}$$

$$\gamma_2 = \{\beta\}$$

$$\gamma_3 = \{\mu, \beta\}$$

$$\gamma_4 = \{\alpha, \mu, \beta\}$$

Aplicando-se a entrada γ_1 ao gerador da figura 2.6, obtêm-se o autômato representado na figura 2.7.

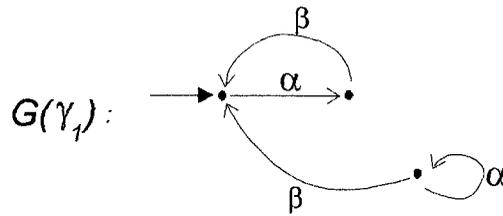


figura 2.7 – Resultado da aplicação da entrada γ_1 no autômato da figura 2.6

2.4. Supervisores

Para se controlar um SED, deve-se realizar o chaveamento das entradas de controle $\gamma_i \in \Gamma$, segundo a observação da cadeia de eventos gerada pela Planta.

O elemento responsável em associar as entradas de controle e os eventos gerados pela Planta chama-se **Supervisor**.

O Supervisor é definido como um mapa, $f : L \rightarrow \Gamma$, que associa cada cadeia gerada em L a uma entrada de controle pertencente a Γ .

Quando o sistema é posto em funcionamento, G opera respeitando as restrições impostas por f . O elemento resultante da associação entre Supervisor e Planta, após a observação da cadeia ω é

$$f(\omega) \cap \Sigma(\widehat{\delta}(\omega, q_0)),$$

que corresponde aos eventos possíveis de ocorrerem sob supervisão, após ω .

O sistema em malha fechada (Supervisor + Gerador) é descrito como (G, f) , ou ainda como f/G .

O comportamento do sistema f/G é definido pela linguagem $L(f/G) \subseteq L(G)$ descrita como:

- i. $\varepsilon \in L(f/G)$
- ii. $\omega\sigma \in L(f/G) \iff \omega \in L(f/G), \omega\sigma \in L(G) \text{ e } \sigma \in f(\omega)$

As seguintes propriedades podem ser observadas:

- i. é sempre verdade que $\{\varepsilon\} \subseteq L(f/G) \subseteq L(G)$;
- ii. $L(f/G)$ é não vazia e fechada.

O comportamento marcado de f/G é definido como:

$$L_m(f/G) = L(f/G) \cap L_m(G)$$

L_m representa as tarefas realizadas que “sobrevivem” sob supervisão.

A seguinte propriedade se observa:

$$\emptyset \subseteq L_m(f/G) \subseteq L_m(G)$$

Diz-se que um supervisor f é **não bloqueante** se:

$$\bar{L}_m(f/G) = L(f/G)$$

Como exemplo, sejam os geradores da figura 2.8.

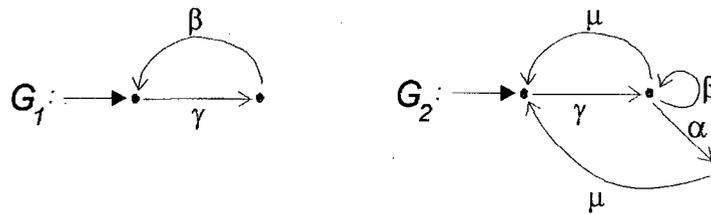


figura 2.8 – exemplo de dois geradores

Seja também:

$$\Sigma_u = \{\gamma, \beta\}$$

$$\Sigma_c = \{\alpha, \mu\}$$

A composição síncrona entre os geradores da figura 2.8 é mostrada na figura 2.9.

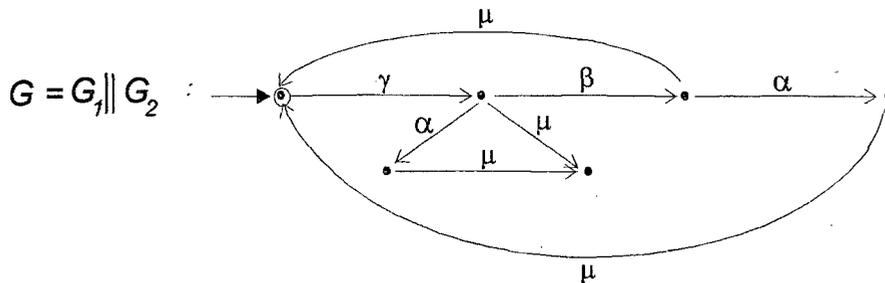


figura 2.9 – produto síncrono entre os geradores da figura 2.8

$$L = ((\gamma\beta)(\mu + \alpha\mu))^* + \gamma(\mu + \alpha\mu)$$

Seja o supervisor $f: L \rightarrow \Gamma$ definido por:

$$\left. \begin{aligned} f\left[\left((\gamma\beta)(\mu + \alpha\mu)\right)^* \gamma\right] &= \{\gamma, \beta\} \\ f\left[\left((\gamma\beta)(\mu + \alpha\mu)\right)^* \gamma\beta\right] &= \{\mu, \gamma, \beta\} \\ f[S] &= \{\mu, \alpha, \gamma, \beta\} \text{ para outros casos} \end{aligned} \right\} L(f/G) = (\gamma\beta\mu)^* (\varepsilon + \gamma + \gamma\beta)$$

O autômato resultante é mostrado na figura 2.10.

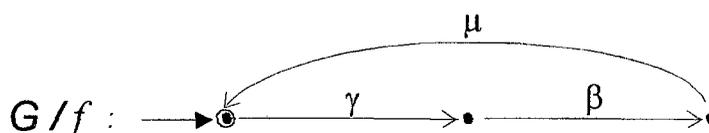


figura 2.10 – resultado da aplicação de f no autômato da figura 2.9

$$L_m(f/G) = L(f/G) \cap L_m(G) = (\gamma\beta\mu)^*$$

2.5. Controlabilidade

Dada uma especificação de funcionamento da Planta, é preciso encontrar um f capaz de atender a esta especificação. Para isso, é necessário aplicar o conceito de *controlabilidade*.

Uma linguagem $K \subseteq \Sigma^*$ é dita controlável em relação a uma linguagem L se

$$(1) \quad \overline{K}\Sigma_u \cap L \subseteq \overline{K}$$

Onde \overline{K} (prefixo-fechamento) é a linguagem formada por todos os prefixos das palavras de K , ou seja, se $u \in \Sigma^*$:

$$\bar{K} \triangleq \{u / uv \in L, v \in \Sigma^*\}$$

Ou seja, a condição (1) indica que uma linguagem K será controlável se, em qualquer ponto no caminho de cadeias de K , caso um evento não controlável seja admissível por L , a ocorrência deste evento ainda esteja prevista no caminho descrito por K .

Para exemplificar, seja novamente o gerador representado pelo autômato da figura 2.9. Seja ainda a linguagem K descrita como:

$$K = (\gamma\beta\mu)^*$$

$$\bar{K} = (\gamma\beta\mu)^*(\varepsilon + \gamma + \gamma\beta)$$

Sejam os casos descritos abaixo:

caso 1:

$$\Sigma_c = \{\alpha, \mu\}$$

$$\Sigma_u = \{\beta, \gamma\}$$

$$\bar{K}\Sigma_u \cap L(G) = (\gamma\beta\mu)^*(\gamma + \gamma\beta) \subset \bar{K}$$

e, portanto, **controlável**.

caso 2:

$$\Sigma_c = \{\beta, \mu\}$$

$$\Sigma_u = \{\alpha, \gamma\}$$

$$\bar{K}\Sigma_u \cap L(G) = (\gamma\beta\mu)^*(\gamma + \gamma\alpha + \gamma\beta\alpha) \not\subset \bar{K}$$

e, portanto, **não controlável**.

2.6. Condições de existência do supervisor

Pode-se considerar 2 diferentes objetivos de uma supervisão: modificar o funcionamento do SED, descrito pela linguagem L , ou modificar o funcionamento marcado do SED descrito pela linguagem L_m , preservando a propriedade de não bloqueio. Estes objetivos dão origem a duas formulações de problemas:

2.6.1 Formulação em termos de linguagens marcadas

Inicialmente, é preciso apresentar o conceito de *L-fechamento*. Sejam K e L duas linguagens com $K \subseteq L \subseteq \Sigma^*$. A linguagem K é dita *L-fechada* se:

$$K = \overline{K} \cap L$$

Ou seja, K é *L-fechada* se todos os seus prefixos, que são palavras de L , forem palavras de K .

Proposição 2.1

“Dados um gerador G tal que $L_m(G)$ represente as tarefas que podem ser completadas na ausência de ação de controle, e uma linguagem alvo $K \subseteq L_m(G)$, $K \neq \emptyset$, existe um supervisor *não bloqueante* f tal que $L_m(f/G) = K$ se e somente se K é *L_m-fechada* e *L-controlável*.”

Demonstração

(se)

Seja K , L_m -fechada e L -controlável.

Define-se $f : L \rightarrow \Gamma$ de acordo com

$$f(\omega) = \Sigma_u \cup \{ \sigma \in \Sigma_c / \omega\sigma \in \overline{K} \}$$

será provado que $L(f/G) = \overline{K}$

$$(1) L(f/G) \subseteq \overline{K}$$

Suponha $\omega\sigma \in L(f/G)$, isto é,

$$\omega \in L(f/G), \sigma \in f(\omega) \text{ e } \omega\sigma \in L(G)$$

Observa-se que $\varepsilon \in \bar{K}$

Se $\omega \in \bar{K}$, tem-se:

i. para $\sigma \in \Sigma_u$

$$\omega\sigma \in \bar{K}\Sigma_u \cap L(G)$$

$$\omega\sigma \in \bar{K} \text{ (por controlabilidade de } K)$$

ii. para $\sigma \in \Sigma_c$

$$\omega\sigma \in \bar{K} \text{ (por definição de } f(\omega)).$$

$$(2) L(f/G) \supseteq \bar{K}$$

Suponha $\omega\sigma \in \bar{K}$, portanto, $\omega\sigma \in L(G)$.

Observa-se que $\varepsilon \in L(f/G)$

Se $\omega \in L(f/G)$, tem-se:

i. para $\sigma \in \Sigma_u$

$$\sigma \in f(\omega)$$

$$\text{e portanto } \omega\sigma \in L(f/G).$$

ii. para $\sigma \in \Sigma_c$

$$\omega\sigma \in \bar{K}$$

$$\omega\sigma \in f(\omega)$$

$$\text{e portanto } \omega\sigma \in L(f/G).$$

A igualdade está provada.

Finalmente,

$$L_m(f/G) = L(f/G) \cap L_m(G)$$

$$= \bar{K} \cap L_m(G)$$

$$= K$$

Também,

$$\bar{L}_m(f/G) = \bar{K} = L(f/G),$$

o que prova que f é não bloqueante.

(somente se)

Seja f um supervisor tal que $L_m(f / G) = K$ não bloqueante, ou seja:

$$\bar{L}_m(f / G) = \bar{K} = L(f / G)$$

$$K = L_m(f / G) = L(f / G) \cap L_m(G) = \bar{K} \cap L_m(G)$$

e portanto K é Lm-fechada.

Para mostrar que K é controlável, seja $\varpi \in \bar{K}$, $\sigma \in \Sigma_u$, e $\varpi\sigma \in L(G)$.

Então $\varpi \in L(f / G)$ e $\sigma \in f(\varpi)$.

Logo $\varpi\sigma \in L(f / G) = \bar{K}$, isto é, $\bar{K}\Sigma_u \cap L(G) \subseteq \bar{K}$.

Proposição 2.2

"Dados um gerador G tal que $L(G)$ represente o comportamento livre fisicamente possível do sistema, e uma linguagem alvo $K \subseteq L(G)$, existe um supervisor f tal que $L(f/G) = K$ se e somente se K é prefixo-fechada e L -controlável."

Demonstração

Considera-se que todos os estados da planta são marcados, ou seja $L_m(G) = L(G)$.

(somente se)

Seja f tal que $L(f/G) = K$. Então

$$(1) \quad L_m(f / G) = L(f / G) \cap L_m(G) = L(f / G) \cap L(G) = L(f / G) = K$$

Além disso:

$$(2) \quad \bar{L}_m(f / G) = \bar{L}(f / G) = L(f / G)$$

o que garante que f é não bloqueante em relação à marcação estabelecida.

Agora pela proposição anterior, (1) e (2) implicam em

K é controlável e

$$K = \bar{K} \cap L_m(G) = \bar{K} \cap L(G) = \bar{K}$$

ou seja, K é prefixo-fechada.

(se)

Seja $K = \bar{K}$ e L-controlável. Então

$$K = \bar{K} \cap L(G) = \bar{K} \cap L_m(G)$$

e, pela proposição 2.1, $\exists f / L_m(f / G) = K$. Mas

$$L_m(f / G) = L(f / G) = K, \text{ o que demonstra a proposição 2.2.}$$

2.7. Máxima Linguagem Controlável

Nesta seção pretende-se mostrar que existe uma aproximação única de K , denominada *máxima linguagem controlável* contida em K , a qual, sob certas condições, satisfaz as condições de existência de um supervisor associado. Este supervisor é também denominado *supervisor minimamente restritivo*.

2.7.1 Existência da máxima linguagem controlável

Sejam $L(G)$ a linguagem gerada por um SED controlado G e $K \subseteq \Sigma^*$.

Define-se então, o conjunto de linguagens controláveis contidas em K como:

$$C(K) \triangleq \{E \subseteq K / \bar{E}\Sigma_v \cap L(G) \subseteq \bar{E}\}$$

Proposição 2.3

$C(K)$ é não vazio e é fechado para operação de união de conjuntos. Em particular, $C(K)$ contém um elemento supremo único denotado $\sup C(K)$ ou $K \uparrow$.

Demonstração

\emptyset (linguagem vazia), é controlável, logo $\emptyset \in C(K)$.

Sejam K_1 e K_2 elementos quaisquer de $C(K)$. Temos:

$$K_1 \cup K_2 \subset K$$

$$\overline{[K_1 \cup K_2]} = K_1 \cup K_2,$$

$$K_1 \Sigma_u \cup K_2 \Sigma_u = \overline{[K_1 \cup K_2]} \Sigma_u$$

Portanto,

$$\overline{[K_1 \cup K_2]} \Sigma_u \cap L(G) = \overline{[K_1 \Sigma_u \cup K_2 \Sigma_u]} \cap L(G)$$

$$= \overline{[K_1 \Sigma_u \cap L(G)] \cup [K_2 \Sigma_u \cap L(G)]} \subseteq \overline{K_1 \cup K_2}$$

$$= \overline{K_1 \cup K_2}$$

Isto demonstra o fechamento do conjunto $C(K)$ para a operação de união de conjunto. Portanto existe um elemento único

$$\sup C(K) = \cup \{E / E \in C(K)\}$$

que contém todos elementos de $C(K)$.

Proposição 2.4

Seja $K \subset L_m(G)$ uma linguagem Lm-fechada ($K = \bar{K} \cap L_m$). Então $\sup C(K)$ é Lm-fechada.

Demonstração

Seja $K \uparrow = \sup C(K)$.

Supõe-se que $K \uparrow$ não é Lm-fechada.

Então existe $S \in \bar{K} \uparrow \cap L_m$ tal que $S \notin K \uparrow$.

Seja agora $K^* = K \uparrow \cup \{S\}$.

Obviamente:

$$\bar{K}^* = \bar{K} \uparrow$$

E, pela controlabilidade de $K \uparrow$, $K^* \in C(K)$ e $K^* \supset K \uparrow$ é controlável, contradizendo o fato de que $K \uparrow$ é supremo.

Portanto $K \uparrow$ é Lm-fechada.

Proposição 2.5

Seja $K \subset L(G)$ uma linguagem prefixo-fechada ($K = \bar{K}$). Então $\sup C(K)$ é fechada.

Demonstração

Seja $K^\uparrow = \sup C(K)$.

Supõe-se que K^\uparrow não é fechada. Então existe $S \in \bar{K}$ tal que $S \notin K^\uparrow$. Seja agora

$$K^* = K^\uparrow \cup \{S\}$$

Obviamente:

$$\bar{K}^* = \bar{K}^\uparrow$$

E, pela controlabilidade de K^\uparrow , K^* é controlável. Mas $K^* \supset K^\uparrow$, contradizendo o fato de que K^\uparrow é supremo.

Portanto K^\uparrow é fechada.

Corolário 2.1

Dado um SED descrito por G , não bloqueante, com funcionamento $L(G)$, temos:

- (1) Se $K \subset L$, $K \neq \emptyset$ e $K = \bar{K}$, existe um supervisor f tal que $L(f/G) = \sup C(K)$.
- (2) Se $K \subset L_m$, $K \neq \emptyset$ e $K = \bar{K} \cap L_m$, existe um supervisor f tal que $L_m(f/G) = \sup C(K)$.

Os supervisores do corolário acima são ótimos, no sentido de que são os que geram comportamento em malha fechada dentro das especificações descritas por K , da forma menos restritiva possível.

A síntese da Máxima Linguagem Controlável pode ser obtida através dos algoritmos descritos em [ZC94].

Neste capítulo, apresentou-se alguns conceitos teóricos básicos da Teoria Clássica introduzida por Wonham e Ramadge. Os conceitos aqui apresentados servem como base para o restante do documento.

O próximo capítulo apresentará uma solução de controle para um sistema bastante simples utilizando a teoria clássica, e demonstrará que, dependendo da forma como o sistema é modelado, pode-se obter resultados não plausíveis.

CAPÍTULO 3 - A TEORIA CLÁSSICA: PROBLEMA ENCONTRADO NUM EXEMPLO DE APLICAÇÃO

Este capítulo apresentará a solução de controle para um sistema bastante simples, que servirá para mostrar que a Teoria Clássica de Wonham e Ramadge, quando aplicada a um modelo que foge a determinadas características, nem sempre apresenta resultados plausíveis, ou seja, que representem a realidade.

Usualmente, ao modelar o comportamento de um sistema real, a tendência é mapear diretamente todos os comandos e leituras de sensores em eventos no modelo. O caso mais clássico de modelagem por autômatos, a "fabriquinha", é descrito da seguinte forma:

O sistema é composto por duas máquinas (M_1 e M_2) e um "buffer", como descrito na figura 3.1.

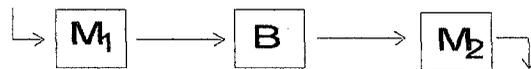


figura 3.1 - A "fabriquinha"

Apenas para exemplificar, as máquinas são modeladas como no autômato da figura 3.2.

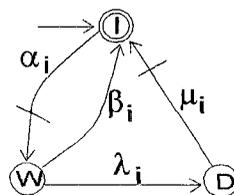


figura 3.2 - Modelo que representa as máquinas da "fabriquinha"

Os estados possíveis são I ("idle"- em repouso), W ("working" - em operação) e D ("down" - quebrada).

Os eventos neste caso, apesar de receberem nomes diferentes, representam diretamente comandos enviados às máquinas, ou a leitura de sensores. Esta relação pode ser vista no quadro abaixo:

Evento	Significado para o modelo	Possível significado para a planta
α_i	início de operação	comando: máquina "i" usinar
β_i	fim de operação	sensor ou sinal indicando fim de usinagem
λ_i	quebra de máquina	sensor ou sinal indicando quebra de máquina
μ_i	reparo da máquina	sensor ou sinal indicando o reparo da máquina

Na figura 3.2, as linhas cortadas representam eventos controláveis, enquanto que as linhas não cortadas representam eventos não-controláveis. É bastante razoável pensar que os eventos de *início de operação* e *recuperação de máquina* possam ser controlados pelo supervisor, enquanto que *fim de operação* e *quebra de máquina* são eventos sobre os quais não se tem controle.

O supervisor, para a teoria clássica, é responsável por habilitar e desabilitar eventos controláveis, mantendo assim o comportamento lógico da planta dentro de certos limites impostos por uma especificação.

No caso da "fábrica", nota-se claramente que, por se tratar de um mapeamento direto de eventos do modelo em comandos ou sensores da planta física, a desabilitação de um evento controlável representa simplesmente a *não aplicação de determinado comando*. Porém, a habilitação de um evento controlável não significa necessariamente a aplicação de um comando. Ou seja, se em determinado estado estiverem habilitados um evento controlável e um evento não-controlável, não é responsabilidade do supervisor, segundo a teoria clássica, decidir sobre a aplicação ou não do evento controlável.

É simples perceber que a planta não pode gerar um evento que é, literalmente, a aplicação de um comando sobre ela própria. Algum outro dispositivo deve ser capaz de gerar este evento.

Sendo assim, quando comandos físicos são mapeados diretamente em eventos controláveis no modelo do nível supervisor, podemos notar dois fatos importantes:

- i. Deve existir, além da planta e do supervisor, uma outra unidade, dotada de uma certa capacidade de "decisão", que defina, entre vários eventos

controláveis habilitados, aquele que efetivamente será ou não aplicado à planta;

- ii. A procedência dos eventos controláveis e não-controláveis, neste caso, é bastante clara. Ou seja, os eventos não-controláveis, que não podem ser desabilitados pelo supervisor, *são gerados pela planta*. Os eventos controláveis, habilitados ocasionalmente pelo supervisor, são gerados em um nível superior ao nível físico.

3.1. A Célula de Manufatura

Ao se tentar resolver um problema de síntese de supervisor para um caso real, utilizando um mapeamento direto de eventos controláveis do modelo em comandos da planta, observou-se que a aplicação da teoria clássica, neste caso, pode apresentar problemas para a solução de controle.

O exemplo a seguir visa mostrar claramente o problema encontrado.

3.1.2 A modelagem utilizando a teoria clássica

O sistema apresentado na figura 3.3 é parte de uma mini-célula de manufatura (figura 3.4), localizada no Laboratório de Automação Integrada, do Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina.

Este pequeno sistema é composto por uma estação de espera e um buffer, ligados por uma esteira, como mostra a figura 3.3.

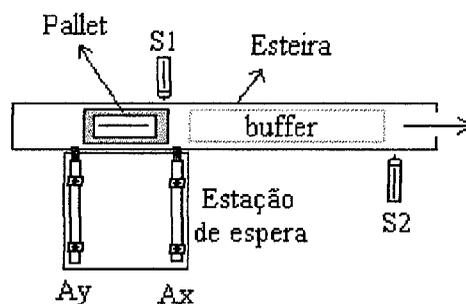


figura 3.3 - Sistema Estação de Espera + Buffer

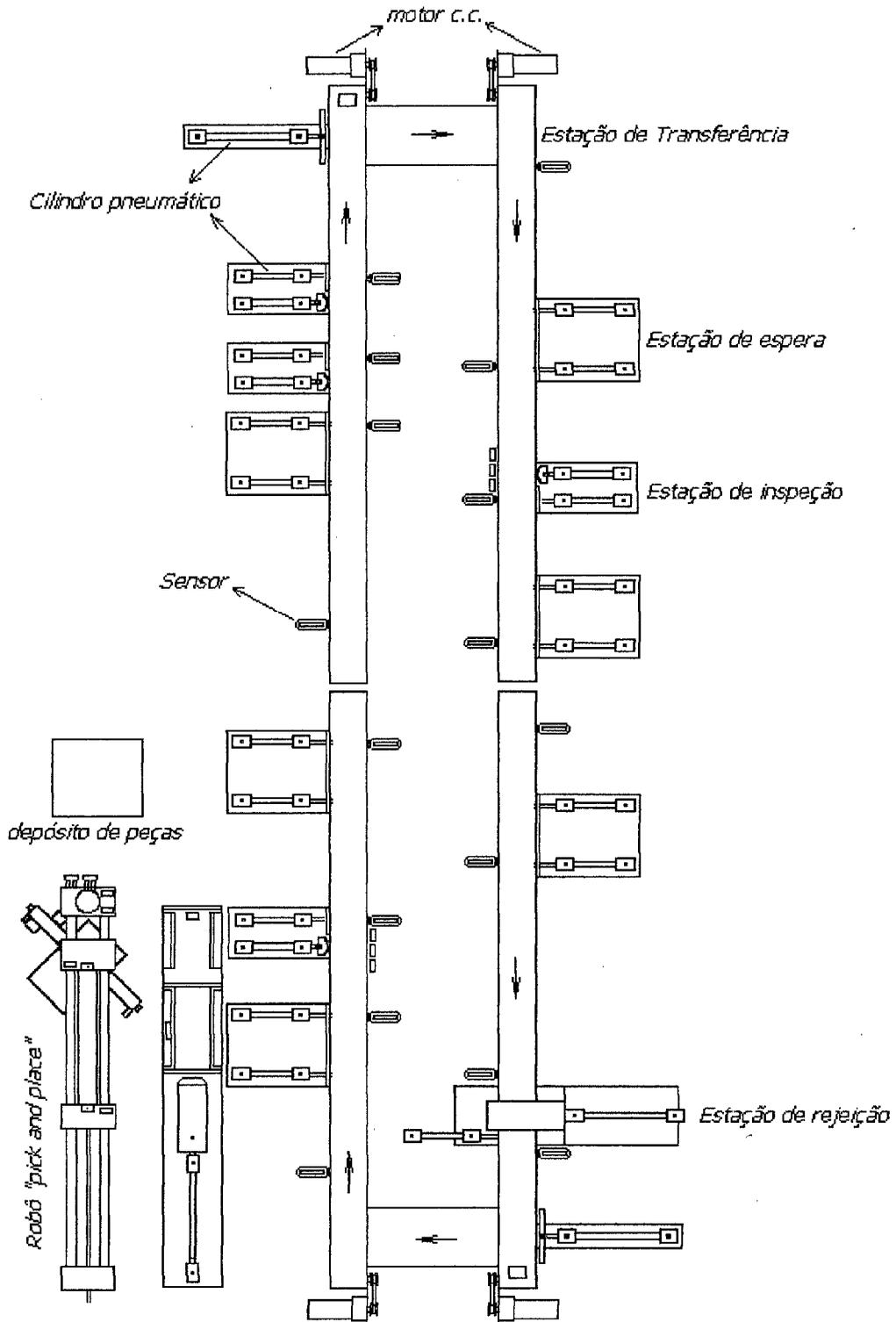


figura 3.4 – Mini-Célula de Manufatura do LAI - DAS

O sistema apresentado na figura 3.3 possui um funcionamento bastante simples: A estação de espera é composta por dois atuadores pneumáticos A_x e A_y . S_1 e S_2 são sensores óticos que detectam a passagem de "pallets".

Quando um "pallet" chega a estação de espera, ou fica parado (atuadores A_x e A_y estendidos) ou passa livremente pela estação (atuadores A_x e A_y recolhidos), sendo carregado pela esteira para outras partes da mini-célula. Quando um "pallet" passa na frente de um sensor, este fica sensibilizado (S_{on}), e caso contrário, o sensor está desligado (S_{off}). O "buffer" é, neste caso, o espaço existente entre os sensores S_1 e S_2 .

O modelo escolhido para a estação de espera foi inicialmente aquele mostrado na figura 3.5.

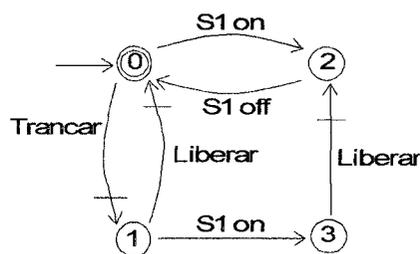


figura 3.5 - Modelo que representa o comportamento livre da "estação de espera"

Nota-se que este modelo reflete, de forma bastante razoável, o comportamento livre da estação. No estado inicial, ou estende-se o atuador A_x (evento "trancar"), ou acontece a chegada de um "pallet" na estação de espera (evento " S_1on "). No estado 1, já que o atuador A_x está estendido, ou pode-se recolhê-lo (evento "liberar"), ou um "pallet" chega a estação e fica bloqueado (o sensor S_1 fica sensibilizado - evento " S_1on "). No estado 2, um "pallet" está passando na frente de S_1 . Se o atuador A_x for estendido, ele pressionará o "pallet" contra a parede da célula, podendo causar danos ao sistema. O Controlador Lógico Programável, responsável pela implementação dos comandos na planta, impede tal situação, representando assim um mecanismo de segurança. Neste caso, a ocorrência do evento "trancar" no estado 2 não é possível, e portanto, o único evento que pode acontecer é o fim da passagem do "pallet" por S_1 (evento " S_1off "). No estado 3, o atuador A_x está estendido e existe um "pallet" parado na estação sensibilizando S_1 . Portanto, o único evento que pode ocorrer é o recolhimento do atuador A_x (evento "liberar").

O atuador A_y é comandado pelo CLP (controlador lógico programável), localizado junto à célula. Toda vez que o atuador A_x for estendido, e o sensor S_1 for sensibilizado em seguida (presença de "pallet" na frente de S_1), o atuador A_y é imediatamente estendido, e somente é recolhido quando o "pallet" for liberado e passar totalmente por S_1 (ocorrência do evento " S_1off ").

O modelo do comportamento do "buffer" é descrito na figura 3.6.

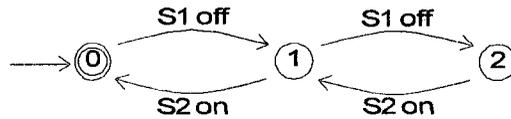


figura 3.6 - Modelo do comportamento do "buffer"

A modelagem do "buffer" levou em conta o número de "pallets" que o espaço entre S_1 e S_2 comporta. Neste caso podem existir fisicamente dois "pallets", sem que estes estejam sensibilizando nenhum dos dois sensores, como mostra a figura 3.7.

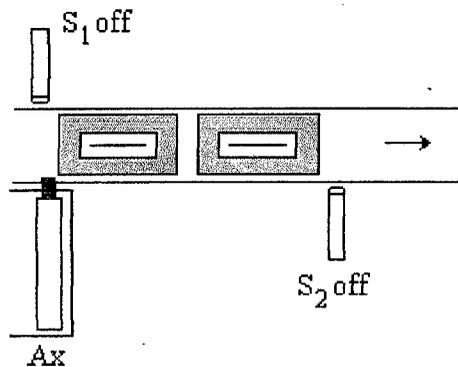


figura 3.7 - Capacidade de armazenagem do "buffer"

A composição síncrona dos dois modelos ("buffer" + estação de espera) é representada pelo autômato da figura 3.8.

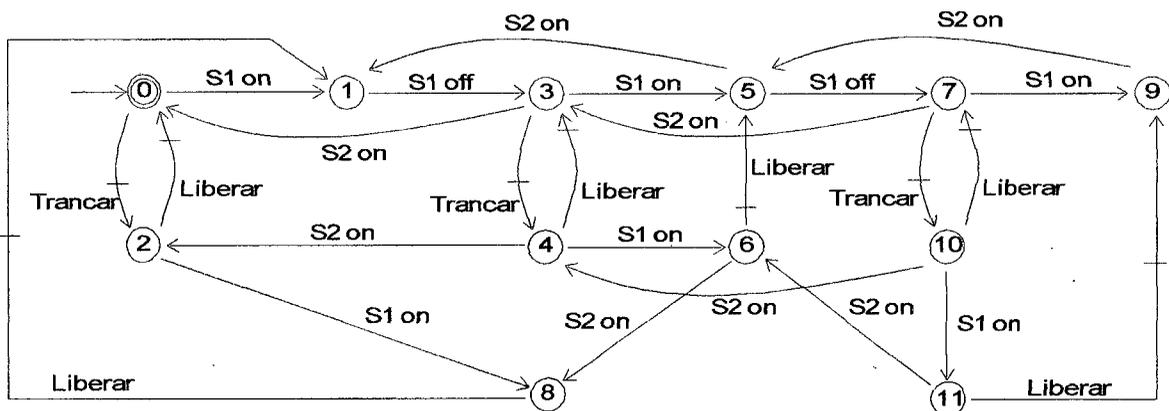


figura 3.8 - Comportamento livre do sistema "estação de espera + buffer".

Este autômato descreve todas as possibilidades de ocorrência de eventos caso o sistema estivesse funcionando sem a presença de um supervisor.

No autômato da figura 3.8, baseando-se na partição do conjunto de eventos proposta pela Teoria Clássica, os eventos "trancar" e "liberar" são descritos como *eventos controláveis*, enquanto que os eventos "S₁off" e "S₂on" são descritos como *eventos não-controláveis*.

Segundo a Teoria Clássica, a natureza do supervisor é de agente desabilitador. Os eventos são assumidos como gerados pela planta.

De fato, na realidade, a situação apresentada na figura 3.8 corresponde a de um sistema sob a ação de um elemento de controle que pode gerar comandos sem nenhuma ação restritiva. Assim, por exemplo, se este agente fosse desligado, todos os eventos controláveis, apesar de habilitados, não ocorreriam. Todos os pallets que chegassem ao sistema seriam transportados livremente pela esteira, sem que ficassem bloqueados na estação.

Uma possível especificação que poderia ser aplicada a este sistema, seria a de **permitir que apenas uma peça por vez possa ocupar o espaço entre S₁ e S₂**, ou seja, o "buffer". Esta especificação poderia ser modelada pelo autômato da figura 3.9.

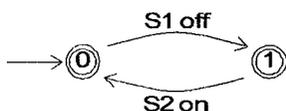


figura 3.9 - Especificação

Este modelo representa claramente uma restrição à capacidade de armazenamento do buffer.

O que se deseja é construir um supervisor que seja capaz de atender a esta especificação.

Aplicando-se esta restrição ao sistema, obtem-se a seguinte linguagem alvo, representada pelo autômato da figura 3.10, resultado da composição síncrona dos autômatos das figuras 3.8 e 3.9.

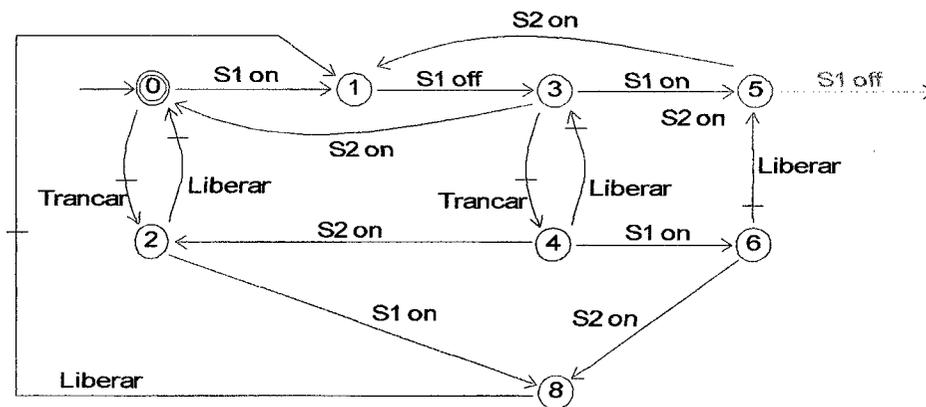


figura 3.10 - Modelo que representa a linguagem-alvo para o sistema a ser controlado

A linha tracejada no estado 5, representa o evento não-controlável " S_1off " que passa a não pertencer ao comportamento desejado da planta. Porém, a ocorrência deste evento no estado 5, apesar de não desejada, não pode ser impedida pelo supervisor, visto que se trata de um evento não-controlável. Ou seja, se o supervisor permitir que a planta evolua até este estado, será impossível impedir que o evento " S_1off " aconteça. O estado 5 é denominado de *mau-estado*, e o modelo acima não representa a máxima linguagem controlável.

Mas o problema principal encontrado na linguagem-alvo acima é que existe um caminho de eventos não-controláveis, partindo do estado inicial, que pode inevitavelmente conduzir a planta ao estado 5, e conseqüentemente, a uma situação não desejada. Não existe ao menos um evento controlável que possa ser desabilitado por um supervisor, que impeça a planta de atingir o estado 5. Ou seja, a máxima linguagem controlável para esta especificação, segundo a teoria clássica, é vazia.

Este resultado é mais uma vez resultado da interpretação de que os eventos são gerados pela planta. Se, no entanto, mais uma vez imaginarmos a existência de um "agente controlador" que é responsável pela geração dos eventos correspondentes a comandos, o resultado acima pode não ser plausível. De fato, é fácil notar que, se toda vez que um "pallet" passar por S_1 (evento " S_1off "), o atuador A_x for estendido (evento "trancar") e permanecer assim até que o "pallet" saia do "buffer" (evento " S_2on "), a especificação acima será atendida.

Este comportamento corresponde a um caminho existente no modelo da figura 3.10, e que satisfaz plenamente a especificação dada, como mostra a figura 3.11.

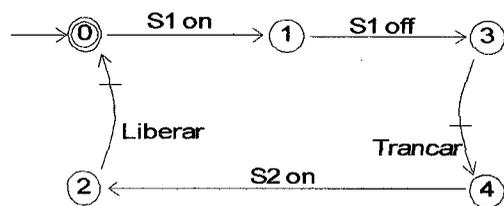


figura 3.11 - Possível caminho que satisfaz a especificação dada

Porém, para que este caminho seja factível, é obrigatória a ocorrência do evento "trancar" no estado 3. Isto implica em dois requisitos:

- que o comando de trancar seja efetivamente enviado ao atuador A_x toda vez que um pallet passar por S_1 ;

- que se assuma que este comando possa ser executado “instantaneamente”, antecipando-se a qualquer outro evento gerado pela planta naquele estado;

É interessante notar que este modelo inicial impõe ao “agente de controle” a responsabilidade de gerar os eventos controláveis, porém a teoria clássica não considera a existência explícita deste agente e portanto não permite que a responsabilidade do supervisor vá além de habilitar ou desabilitar eventos.

Para resolver este problema existem duas opções:

- i. Rever a modelagem de todo o sistema;
- ii. Utilizar novas abordagens para o tratamento de síntese de supervisores.

3.2. Um novo modelo para o sistema “estação de espera + buffer”

Para este novo modelo, não será utilizado o mapeamento direto de eventos controláveis do nível supervísório em comandos do nível físico. Isto significa que os comandos de “trancar” e “liberar”, aplicados aos atuadores A_x e A_y , não serão explicitamente tratados como eventos.

Nesta nova modelagem, os eventos que ocorrem na planta serão tratados de uma maneira menos específica e mais abstrata.

Sob um ponto de vista funcional, a estação de espera “tem a função” de permitir ou não a passagem de pallets. Esta “permissão” pode ser vista como um evento controlável, já que a decisão sobre a passagem ou não de determinado pallet caberá ao supervisor.

Com este único evento pode-se descrever o comportamento da estação de uma maneira bastante simples e funcional. Este modelo é mostrado na figura 3.12.



figura 3.12 - Novo modelo para a “estação de espera”

O evento “passar” continua sendo descrito como um evento controlável. Porém, da forma como foi modelado, não existe nenhuma correspondência direta entre este evento e um comando no nível físico.

Por outro lado, habilitar o evento “passar” significa permitir que um pallet passe pela estação. Fisicamente, esta permissão implica na implementação de um comando de recolher o atuador A_x . Por outro lado, desabilitar este evento significa proibir a passagem de “pallets”. Isto implica implementar o comando de estender o atuador A_x . Já a

ocorrência deste evento significa que um pallet acabou de passar pela estação, o que fisicamente significa que o sensor S_1 passa do estado de sensibilização para o estado de não sensibilização - " S_{1off} ".

O quadro abaixo mostra uma comparação entre o modelo proposto inicialmente e este novo modelo, para o mapeamento dos eventos controláveis.

Modelo Inicial			Novo Modelo		
<i>Eventos controláveis são mapeados diretamente em comandos do nível físico</i>			<i>Eventos controláveis representam associações de comandos e leitura de sensores</i>		
Habilitar	Desabilitar	Ocorrer	Habilitar	Desabilitar	Ocorrer
-	-	aplicar um comando	aplicar um comando	aplicar um comando	alteração do estado de um sensor

Portanto, a habilitação e desabilitação dos eventos controláveis é de responsabilidade do supervisor. Já a ocorrência destes eventos depende exclusivamente da planta, o que representa exatamente aquilo que a Teoria Clássica define como "evento controlável". Ou seja, se no modelo inicial era exigido que, se fosse considerada a existência de um "agente de controle", o supervisor gerasse os eventos controláveis, neste novo modelo quem gera estes eventos é a planta, e o modelo é compatível com a Teoria Clássica.

Isto representa exatamente o que a teoria clássica espera de um modelo.

O modelo do buffer segue a mesma idéia, mantendo a mesma estrutura, como mostra a figura 3.13.

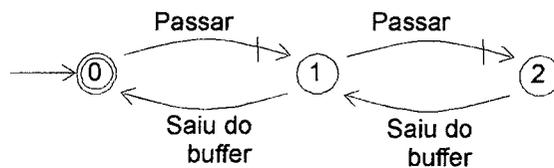


figura 3.13 - Novo modelo para o "buffer"

O evento "*saiu do buffer*" pode ser mapeado diretamente no evento " S_{2on} ".

Ao se compor os modelos apresentados nas figuras 3.12 e 3.13, obtém-se um modelo para o comportamento livre da planta que é idêntico ao modelo apresentado para

o "buffer". Considera-se, pois, o modelo da figura 3.13, como sendo também o modelo do comportamento livre da planta.

O primeiro aspecto importante a ser notado é a diminuição do número de estados.

Outro fato interessante é que o nível de abstração utilizado tornou o modelo mais "legível".

A especificação inicial, assim como o "buffer", apresenta a mesma estrutura, como mostra a figura 3.14.

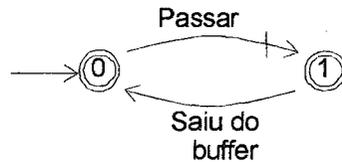


figura 3.14 - Especificação utilizando novos eventos

Aplicando-se esta especificação ao sistema obtém-se o autômato da figura 3.15.

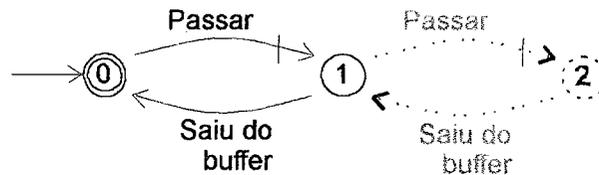


figura 3.15 - Linguagem-alvo para o novo modelo

Da mesma forma como no modelo anterior, apesar de ser fisicamente possível atingí-lo, o estado 2 não pertence ao comportamento desejado, o que obriga o supervisor a impedir que o sistema atinja este estado.

Para que isso aconteça, basta ao supervisor desabilitar o evento "passar" no estado 1. A máxima linguagem controlável pode ser então representada pelo seguinte autômato da figura 3.16. Este modelo é idêntico ao modelo da especificação, apresentado na figura 3.14.

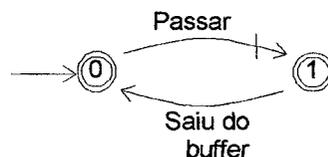


figura 3.16 - Máxima Linguagem Controlável para o novo modelo

A desabilitação do evento "passar" no estado 1 significa, como mencionado anteriormente, estender o atuador A_x , bem como a habilitação deste evento no estado inicial implica em recolher A_x .

É interessante observar que este modelo representa, de uma forma mais abstrata, o mesmo caminho apresentado na figura 3.11.

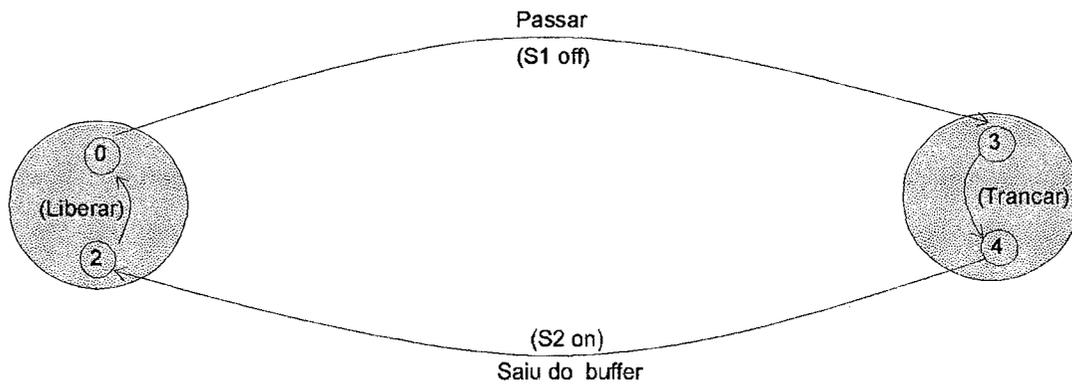


figura 3.17 – Equivalência entre o caminho exibido na figura 3.11 e o modelo da figura 3.16

O fato deste modelo apresentar um mapeamento indireto entre eventos controláveis e eventos físicos requer a existência de um módulo intermediário, colocado entre o supervisor e a planta. Este módulo deve ser capaz de "traduzir" a habilitação e desabilitação dos eventos controláveis em comandos da planta.

Neste capítulo, demonstrou-se que a aplicação da Teoria Clássica sobre modelos que fogem à determinadas características, pode resultar em soluções não coerentes com a realidade. Tal teoria supõe que todos os eventos, controláveis e não-controláveis, sejam gerados pela planta. Dependendo da modelagem que se utilize, tal suposição poderá apresentar soluções não adequadas à realidade. Na prática, existem eventos que, para ocorrerem, precisam necessariamente da intervenção de um agente controlador externo. Uma possível solução aqui apresentada foi a de refazer o mapeamento inicial, de forma que todos os eventos da planta se adequassem ao modelo proposto por Wonham e Ramadge. Esta solução requer a existência de um módulo externo, que seja capaz de implementar os comandos que a planta, por si só, não pode gerar.

O próximo capítulo estudará uma segunda solução para o problema aqui apresentado, utilizando uma nova abordagem para o tratamento de síntese de supervisores.

CAPÍTULO 4 - EXTENSÃO À TEORIA CLÁSSICA: A NOÇÃO DE EVENTOS FORÇÁVEIS

Este capítulo estudará a noção de "Eventos Forçáveis", apresentada por Golaszewski e Ramadge em uma extensão à Teoria Clássica [RW87]. O modelo do capítulo 3 será revisto utilizando esta abordagem.

A idéia de estender a teoria clássica foi inicialmente proposta por *Golaszewski e Ramadge*, onde foi introduzida a noção de uma nova classe de eventos, os *eventos forçáveis*.

Um evento desta classe ocorre se e somente se for forçado por alguma entrada externa.

A utilização deste conceito, implica na revisão de três pontos importantes da teoria clássica:

- conjunto Σ de eventos;
- conjunto Γ de entradas de controle;
- Linguagens controláveis.

4.1. O conjunto Σ eventos

Os eventos, pela teoria clássica, são divididos em dois conjuntos distintos Σ_u e Σ_c , onde Σ_u contém todos os eventos classificados como não-controláveis, e Σ_c contém todos os eventos controláveis.

Nesta nova abordagem, o conjunto Σ é dividido em três conjuntos distintos de eventos: Σ_u e Σ_c , como descritos anteriormente, e o conjunto Σ_f , que contém os *eventos forçáveis*.

Voltando ao modelo inicial do sistema "estação de espera + buffer", os eventos forçáveis seriam todos os eventos mapeados diretamente em comandos na planta. Ou seja, os eventos anteriormente classificados como eventos controláveis, passam a pertencer ao conjunto Σ_f .

4.2. O conjunto Γ de entradas de controle

A partir do novo conjunto de eventos, define-se um novo conjunto de entradas de controle, como sendo:

$$\Gamma = \left\{ \begin{array}{l} \gamma: \gamma \in 2^\Sigma \wedge \gamma = \{\sigma\} \wedge \sigma \in \Sigma_f \\ \vee \\ \gamma \subseteq 2^{\Sigma_u \cup \Sigma_c} \wedge \Sigma_u \subseteq \gamma \end{array} \right\}$$

Neste conjunto, nota-se a existência de dois tipos bem definidos de entradas de controle: ou um comando aplicado por um nível superior ao nível da planta; ou um conjunto contendo os eventos controláveis, que ocasionalmente estão habilitados, e todos os eventos não-controláveis.

A utilização deste conjunto de entradas de controle implica na realização de um supervisor determinístico.

É interessante permitir ao supervisor a opção de fornecer uma entrada de controle que seja a união dos dois tipos de conjuntos citados acima. Isto pode ser feito definindo-se o conjunto de entradas de controle como sendo o fechamento de Γ , representado por $cl(\Gamma)$. Este novo conjunto garante a existência de uma máxima linguagem controlável, pois é fechado para a operação de união.

O conjunto $cl(\Gamma)$ será utilizado adiante como o conjunto de entradas de controle para o sistema "estação de espera + buffer".

4.3. Linguagens Controláveis

Como definido em [GR87], para cada $K \subseteq \Sigma^*$ e $s \in \bar{K}$, o conjunto ativo de K depois de s é:

$$\sum_K(s) = \{\sigma \in \Sigma: s\sigma \in \bar{K}\} ; \text{ ou seja,}$$

$\sum_K(s)$ contém todos os eventos possíveis de ocorrer depois que s ocorre.

Seja também $L(G)$ a linguagem gerada pelo autômato da planta.

Um linguagem $K \subseteq L(G)$ é dita $(L(G), cl(\Gamma))$ -controlável se

$$\forall s \in \bar{K} \exists \gamma \in cl(\Gamma) : \gamma \cap \sum_{L(G)}(s) = \sum_K(s)$$

Isto significa que, para toda palavra s contida em \bar{K} , deve existir uma entrada de controle tal que, o conjunto dos eventos que a definem e que sejam fisicamente admissíveis, é igual ao conjunto dos eventos que podem ocorrer em \bar{K} , após s .

Para exemplificar esta noção de controlabilidade, sejam os seguintes conjuntos de eventos:

$$\Sigma_c = \{c\}$$

$$\Sigma_u = \{u\}$$

$$\Sigma_f = \{f\}$$

Seja também o seguinte sistema:

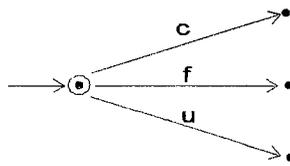
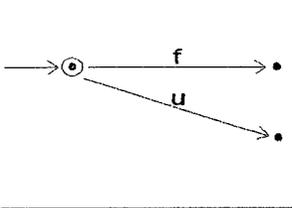
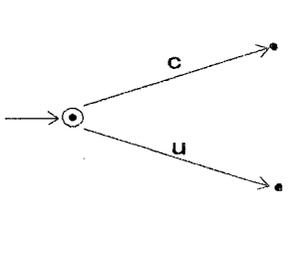
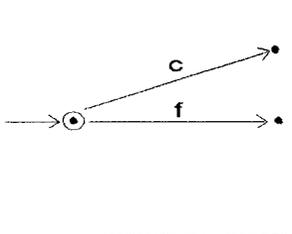
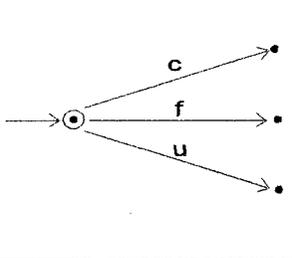


figura 4.1 - Um Sistema exemplo para a aplicação da noção de controlabilidade com eventos forçáveis

O quadro abaixo mostra todas as possíveis linguagens-alvo para o sistema acima, classificando-as como controláveis ou não, segundo esta nova noção de controlabilidade.

Linguagem-alvo K	$\Sigma_K(s)$	γ	Controlabilidade	Comentário
	$\{f\}$	$\{f\} \in \text{cl}(\Gamma)$	Controlável	A aplicação de $\gamma = \{f\}$ se antepõe a ocorrência de qualquer outro evento
	$\{u\}$	$\{u\} \in \text{cl}(\Gamma)$	Controlável	
	$\{c\}$	$\{c\} \notin \text{cl}(\Gamma)$	Não-Controlável	Se c for habilitado, u deve ser também habilitado

	$\{f, u\}$	$\{f, u\} \in cl(\Gamma)$	Controlável	f pode ou não ser aplicado
	$\{c, u\}$	$\{c, u\} \in cl(\Gamma)$	Controlável	
	$\{f, c\}$	$\{f, c\} \notin cl(\Gamma)$	Não-Controlável	Se c for habilitado, u deve ser também habilitado
	$\{f, c, u\}$	$\{f, c, u\} \in cl(\Gamma)$	Controlável	f pode ou não ser aplicado

4.4. Solução para o sistema "estação de espera + buffer" utilizando eventos forçáveis

Deseja-se então, utilizar esta nova abordagem afim de construir um controlador (supervisor) para o sistema "estação de espera + buffer".

Primeiramente, o conjunto Σ de eventos será novamente definido como sendo

$\Sigma = \Sigma_u \cup \Sigma_c \cup \Sigma_f$; onde:

$$\Sigma_u = \{ "S_{1on}" , "S_{1off}" , "S_{2on}" \}$$

$$\Sigma_f = \{ "trancar" , "liberar" \}$$

$$\Sigma_c = \emptyset$$

Nada é modificado nos modelos propostos.

O que deve ser revisto é o cálculo da máxima linguagem controlável, a partir da linguagem-alvo descrita na *figura 3.10*.

Utilizando a noção de eventos forçados, nota-se que o estado 5 continua a ser um mau-estado, pois para uma palavra s qualquer que atinja este estado, $\sum_K(s) = \{ "S_{2on}" \}$, e não existe uma entrada de controle que habilite " S_{2on} " e não habilite " S_{1off} ".

Ao eliminar o estado 5, obtem-se o seguinte:

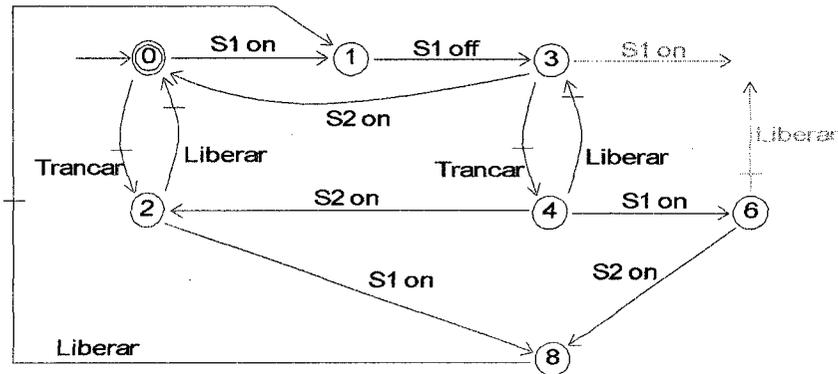


figura 4.2 - Aplicação da noção de controlabilidade ao sistema "estação de espera + buffer" utilizando eventos forçáveis

Para se evitar que o sistema evolua para o estado 5, eliminado do modelo, o evento " S_{1on} " no estado 3 não pode mais ocorrer, e $\sum_K(s) = \{ "S_{2on}" , "trancar" \}$. Não existe, porém, uma entrada de controle que habilite " S_{2on} " e não habilite " S_{1on} ". No entanto, se o evento " S_{2on} " for retirado do modelo, $\sum_K(s) = \{ "trancar" \}$, e a entrada de controle $\gamma = \{ "trancar" \}$ pertence ao conjunto de entradas de controle $cl(\Gamma)$.

É lógico perceber que, para "desabilitar" um evento não-controlável é necessário forçar a ocorrência de um outro evento (caso seja possível). Portanto, em situações como esta (estado 3 no modelo acima), apenas eventos forçáveis podem ser habilitados.

Já no estado 6, para evitar que o sistema atinja o estado 5, basta que o evento " $liberar$ " seja desabilitado.

Finalmente, a máxima linguagem controlável, utilizando a noção de eventos forçáveis, é descrita pelo seguinte autômato:

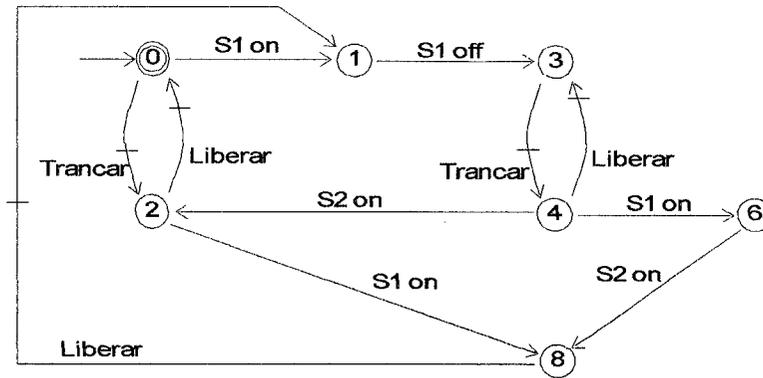


figura 4.3 - Máxima Linguagem Controlável para o sistema "estação de espera + buffer" utilizando a abordagem de eventos forçáveis

Neste capítulo foi apresentada a noção de Eventos Forçáveis. O conceito de controlabilidade foi revisto, aplicando-o ao modelo do capítulo 3.

O próximo capítulo apresentará uma discussão sobre a utilização de eventos controláveis, conforme apresentado no capítulo 3, e os eventos forçáveis, e também apresentará uma proposta de arquitetura física para a implementação do controle de Sistemas a Eventos Discretos.

CAPÍTULO 5 – PROPOSTA DE UMA ARQUITETURA FÍSICA PARA IMPLEMENTAÇÃO DE CONTROLE

Este capítulo apresenta uma abordagem geral para o tratamento de sistemas a Eventos Discretos, considerando as idéias discutidas nos capítulos anteriores. Propõe também um modelo de arquitetura física para o supervisor de uma planta orientada a eventos discretos.

5.1 Introdução

Nos capítulos anteriores o problema da modelagem dos SED`s foi bastante discutido. Foram apresentadas duas soluções para o tratamento prático deste tipo de sistemas, as quais são:

- i. A noção de *eventos controláveis* não pode ser apenas aquela que afirma, conforme a teoria clássica, que o supervisor tem apenas o papel de habilitar ou desabilitar estes eventos. Muitos eventos classificados como controláveis não são gerados pela planta, o que impõe a necessidade da existência de um outro mecanismo capaz de fazer com que estes eventos ocorram. Viu-se também que, para que a Teoria Clássica de Controle de SED`s apresente sempre resultados coerentes, os eventos controláveis precisam ser mapeados como associações entre comandos e leitura de sensores, o que lhes concede uma característica muito mais funcional do que operacional. Conforme descrito no capítulo 3, isto significa que, ao se utilizar eventos controláveis numa modelagem de sistemas, tais eventos não serão vistos diretamente no nível da planta (operacional), mas como funções existentes no sistema (e.g.: o evento "passar pallet" descrito na figura 3.11);
- ii. A noção de *eventos forçáveis*, que traz consigo uma extensão no conceito de controlabilidade de sistemas, permite a modelagem de eventos que não são apenas habilitados ou desabilitados pelo supervisor, mas que podem ser forçados por um mecanismo distante da planta. Isto indica que, forçar a ocorrência de um evento deste tipo,

exclui a possibilidade de ocorrência de qualquer outro evento, mesmo que este seja gerado pela planta.

Com relação às duas soluções acima, pode-se observar que a solução utilizando eventos forçáveis engloba a primeira. Isto é devido ao fato de que a noção de controlabilidade apresentada no capítulo 4 é uma extensão àquela apresentada pela Teoria Clássica. Logo, o conceito de controlabilidade utilizado na seção 4.4 poderia ser também utilizado na seção 3.2, obtendo-se o mesmo resultado apresentado na figura 3.16. É bom lembrar que os conceitos introduzidos por *Golaszewski* e *Ramadge* não excluem a existência do conjunto de eventos controláveis, mas sim adiciona um novo conjunto, no qual se encontram os eventos forçáveis. Isto mostra também que as duas soluções não são mutuamente exclusivas, mas que podem coexistir, aplicando-se para isso o conceito de controlabilidade apresentado na seção 4.3.

Uma segunda observação que pode ser feita é a de que as duas soluções necessitam da existência de um terceiro componente no sistema, o qual deve ser capaz de gerar eventos para a planta. É claro observar que, diferentemente da solução que utiliza eventos controláveis, os conceitos apresentados no capítulo 4 já contemplam a existência deste elemento. Ou seja, a noção de eventos forçáveis já inclui a idéia de que estes eventos não são gerados pela planta, mas sim por um outro elemento do sistema. Já a solução com eventos controláveis não afirma isto explicitamente, mas é óbvio notar a necessidade que se tem da existência de outro elemento, além da planta, que aplique os comandos de habilitação e desabilitação dos eventos controláveis.

Uma questão que pode ser levantada é: "Mas qual solução utilizar?". Como visto anteriormente, ambas podem estar contidas numa mesma solução para um determinado problema. Esta decisão dependerá:

- i. do sistema em estudo. Existem sistemas onde trabalhar com o conceito de eventos controláveis, tal como descrito no capítulo 3, torna-se complicado ou mesmo impossível. Nestes casos, o uso do conceito de eventos forçáveis torna-se a única opção para a solução do problema. Já para os sistemas onde são possíveis as associações entre comandos e sinais de sensores, compondo desta forma os eventos controláveis, pode-se notar que, através do capítulo 3, a utilização deste conceito leva a modelos menores, o que pode facilitar a implementação de controladores.
- ii. da forma como se deseja modelar o sistema. Cabe à pessoa que está estudando o problema, decidir, dentro das possibilidades que ela possui, a utilização de uma ou outra forma de modelagem. É importante notar que se deve, na medida do possível, buscar um comprometimento entre

a fidelidade e clareza na modelagem do sistema, com os algoritmos que serão implementados a partir do modelo proposto, buscando sempre uma solução otimizada;

5.2 Proposta para uma Arquitetura Física

No contexto de *gerenciamento de sistemas reais*, várias soluções já foram apresentadas e publicadas [AC86][BCJ89][Br84][VY89]. Estas soluções abordam este problema utilizando conceitos como monitoramento de execução, diagnóstico de falhas, inteligência artificial, processamento paralelo etc.

Porém, poucos são os trabalhos que abordam a questão baseando-se na teoria de controle de sistemas a eventos discretos, como a solução descrita em [BWB92]. Ainda assim, as soluções apresentadas se mostram bastante complicadas e não elucidativas do ponto de vista de integração entre modelo e arquitetura física.

O gerenciamento de SED's na prática, necessita não só da concepção de estratégias de controle supervisorio, mas também da implementação destas estratégias, ou seja, do controle supervisorio realizado em tempo-real [BWB91]. Isto demonstra que a implementação da solução deve ser coerente com as estratégias planejadas.

A idéia aqui não é a de detalhar uma arquitetura física para um sistema qualquer. Na verdade, o que se pretende é fornecer, de uma forma geral, uma base sólida, através da qual se possa construir programas ou sistemas (incluindo-se aqui o hardware) que sejam realizáveis na prática, e que sejam coerentes com a modelagem do sistema em estudo. Além disso, a arquitetura aqui proposta reflete toda a análise sobre modelagem de sistemas feita nos capítulos anteriores.

A arquitetura física aqui apresentada possui 4 elementos ou módulos:

1. *Planta*. Sistema a ser controlado;
2. *Interface*. Este elemento é responsável por enviar à planta os comandos relativos aos eventos forçáveis ou controláveis, bem como relacionar os eventos ocorridos no nível físico com os eventos no nível do supervisor. Ou seja, ele deve "traduzir" a linguagem tratada no nível do supervisor para a linguagem da planta, e vice-versa. Como exemplo, suponha o autômato mostrado na figura 5.1, o evento "trancar" descrito no nível do supervisor seria "traduzido" para a planta como sendo o comando de estender o atuador A_x . Já a sensibilização do sensor S_2 , mostrado na mesma figura 5.1 seria enviada ao supervisor como sendo o evento " S_{2on} ".

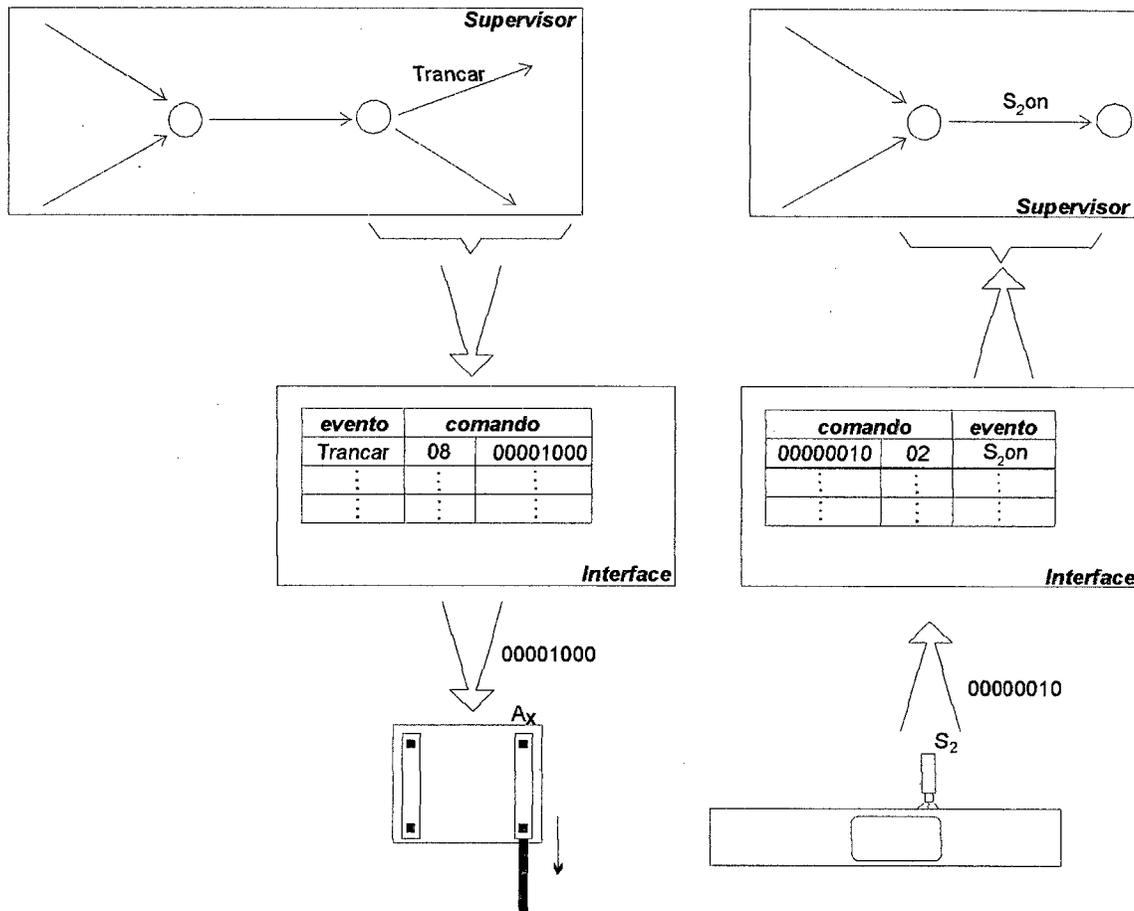


figura 5.1 – Funcionamento do módulo Interface

3. *Módulo Intermediário*. É responsável em enviar à interface os eventos controláveis habilitados e desabilitados, bem como decidir qual dentre os eventos forçáveis habilitados deve ocorrer.
4. *Supervisor*. Como proposto pela teoria clássica, é o elemento que impõe à planta as restrições que levam o sistema a atender as especificações desejadas.

5.2.1 O Supervisor

Conforme visto no capítulo 4, o supervisor, mesmo com a noção de *eventos forçáveis*, continua sendo um mapa que a cada estado estabelece quais eventos devem ser habilitados e quais devem ser desabilitados. Esta informação deve ser levada ao *módulo intermediário* que decidirá ou não sobre o envio de um comando ao nível físico.

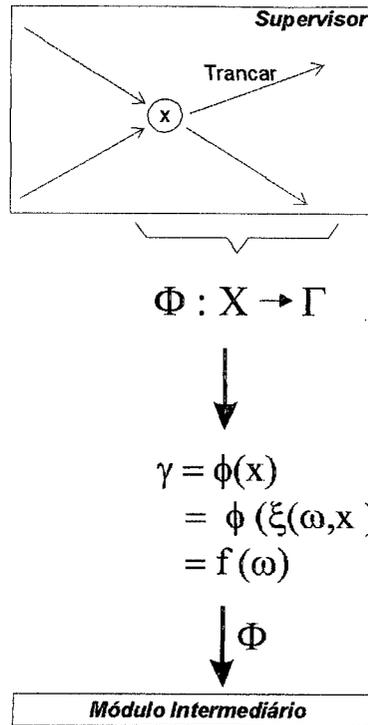


figura 5.2 – O Supervisor como um mapa $\Phi(x)$

5.2.2 O Módulo Intermediário

O Módulo Intermediário (MI), como descrito anteriormente, tem a responsabilidade de funcionar como um “agente controlador”, com a função de agir sobre a planta, enviando eventos que a planta, por si só, não pode gerar.

Este módulo divide-se em dois submódulos, a saber:

- i. submódulo MIC, que trata os eventos controláveis, habilitados e desabilitados;
- ii. submódulo MIF, que trata os eventos forçáveis, com uma característica decisória com respeito a aplicação ou não destes eventos.

O MI receberá do Supervisor a informação sobre quais eventos podem ocorrer em cada estado.

Os eventos não-controláveis não necessitam ser tratados pelo MI. Estes eventos, como são sempre mapeados em leitura de sensores, são gerados pela planta e enviados diretamente ao supervisor, através do módulo Interface.

Para o tratamento dos eventos controláveis descritos em 3.2, o MI deve possuir tanto a informação sobre os eventos habilitados quanto sobre os eventos desabilitados, já que ambas as situações sempre representam comandos que devem ser enviados ao nível físico. Ao receber esta informação do supervisor, o MI deve **imediatamente** enviar

os comandos associados ao módulo Interface. Logo, para este tipo de evento não há a necessidade de nenhum tipo de tomada de decisão sobre qual comando enviar.

No caso dos *eventos forçáveis*, o supervisor envia ao MI somente os eventos habilitados, já que a desabilitação destes eventos significa apenas a não aplicação do comando associado.

O MI deve, em seguida, decidir se aplica ou não o comando associado ao evento. Além disso, deve decidir, entre vários eventos forçáveis, qual efetivamente será aplicado. Neste caso, exige-se uma tomada de decisão, que pode ser feita utilizando-se regras de prioridades, ou mesmo a aplicação de métodos probabilísticos.

Quando a informação enviada pelo supervisor contém a habilitação de eventos forçáveis e também a habilitação e desabilitação de eventos controláveis, a aplicação dos comandos associados aos eventos controláveis deve ser imediata, enquanto que a aplicação dos comandos associados aos eventos forçáveis depende da decisão a ser tomada. Se não houver a aplicação de nenhum *evento forçável*, a planta se encarregará de gerar algum evento (controlável ou não-controlável).

Porém, em casos onde a informação enviada pelo supervisor apenas contém a habilitação de *eventos forçáveis*, a aplicação de um destes eventos deve **obrigatoriamente** ocorrer. Isto se faz necessário quando acontece uma das duas situações abaixo:

- i. Existe um evento não-controlável "indesejado" que pode ser gerado pela planta. Caso não se aplique um evento forçável, o sistema pode ser levado para um "mau-estado";
- ii. Existem apenas eventos forçáveis que podem ocorrer naquele estado. Se um destes eventos não ocorrer a planta pára de evoluir.

Se um *evento forçável* for aplicado, esta informação deve voltar ao supervisor de modo a alertá-lo da ocorrência do evento, já que a planta não pode fornecer este tipo de informação, a qual é gerada exclusivamente pelo MI.

Para efeito de ilustração, seja ϕ_c os eventos controláveis habilitados e desabilitados pelo supervisor; ϕ_f os eventos forçáveis habilitados pelo supervisor; γ_c os comandos associados a habilitação e desabilitação dos eventos controláveis; γ_f os comandos associados aos eventos forçáveis; η_c os eventos controláveis ocorridos; η_f os eventos forçáveis ocorridos; e η_{nc} os eventos não-controláveis ocorridos;

A arquitetura proposta pode ser vista na *figura 5.3*,

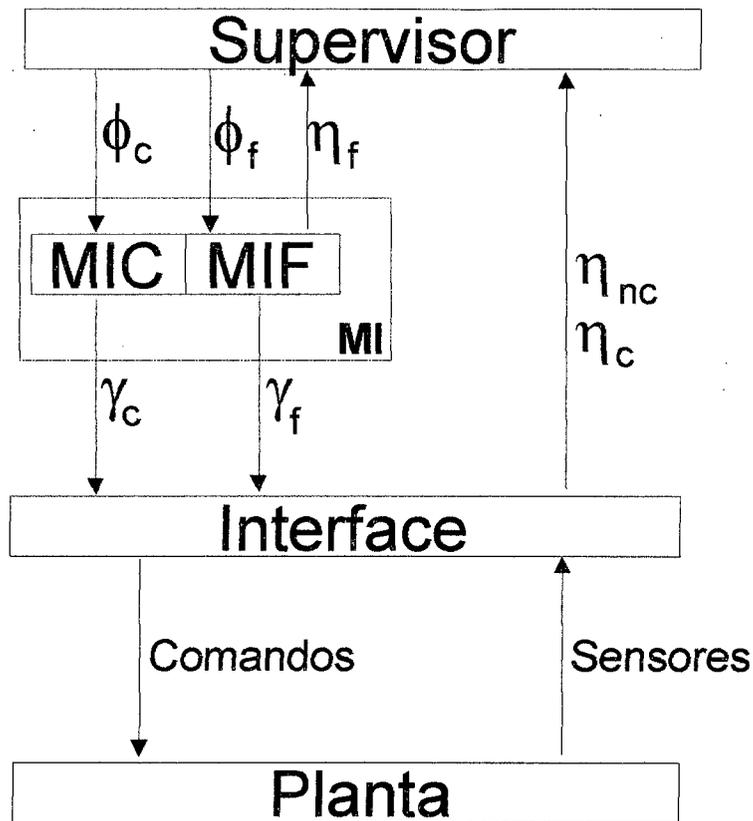


figura 5.3 - Proposta de uma arquitetura física para o controle supervísório de sistemas reais

Ou ainda, considerando um diagrama mais sintético que representa a malha fechada de controle, como mostra a figura 5.4. Nesta figura, considera-se G a planta e S o supervisor, sendo que a Interface está incorporada à planta.

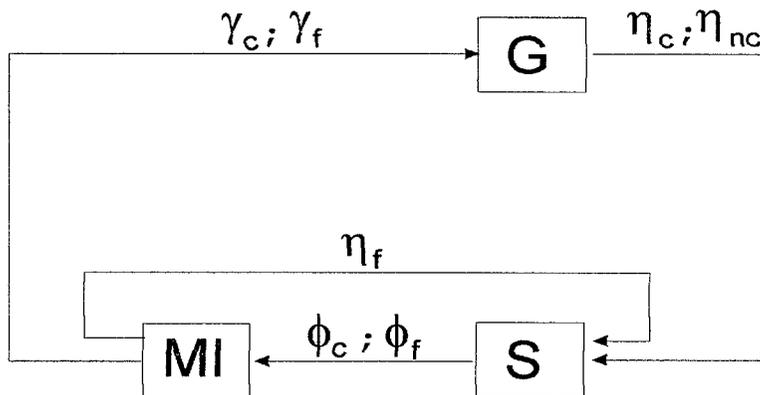


figura 5.4 - Malha fechada para o controle supervísório de sistemas reais

Neste capítulo, apresentou-se uma proposta de uma arquitetura física para o controle de sistemas a eventos discretos. Esta arquitetura é baseada na Teoria Clássica, mas inclui a

extensão feita por *Golaszewski* e *Ramadge* em [GR87] e discutida neste trabalho durante o capítulo 4.

No próximo capítulo serão apresentados alguns modelos que exemplificarão também o uso de eventos forçáveis e controláveis.

CAPÍTULO 6 - EXEMPLOS DE APLICAÇÃO

Este capítulo apresenta três exemplos de modelagem, destacando-se o uso de eventos forçáveis e controláveis conforme descrito nos capítulos 3 e 4.

6.1 Introdução

Como afirmado no capítulo 5, a modelagem de um sistema a eventos discretos depende tanto do sistema com o qual se está trabalhando, como também da maneira como se deseja modelar o sistema, em casos onde existe a possibilidade de escolha. Este capítulo pretende exemplificar o que foi dito no capítulo 5, através da modelagem de alguns sistemas.

Pretende-se também fornecer exemplos que mostrem quando é possível ou não fazer uso de eventos controláveis. Como visto no capítulo 3, o uso de eventos controláveis diminui o tamanho do modelo, o que diminui a carga computacional.

6.2 Uma Estação de Inspeção

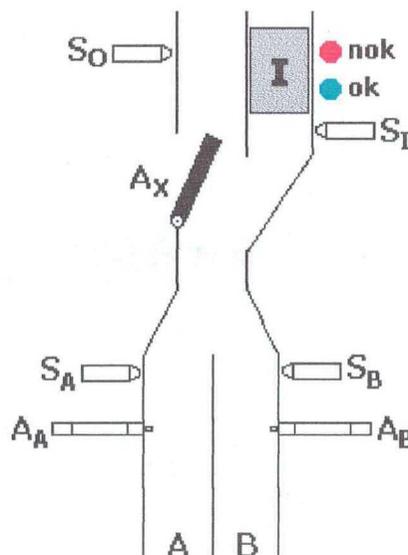


figura 6.1: Desenho esquemático da Estação de Inspeção

A Estação de Inspeção apresentada na figura 6.1 tem o seguinte funcionamento: Existem dois caminhos de entrada para o sistema, um pela *Esteira A* e outro pela *Esteira B*. Quando as peças chegam (eventos S_{Aoff} ou S_{Boff}), elas podem ser selecionadas para inspeção (atuador A_x) ou podem seguir livremente para a estação seguinte (evento S_{Ooff}). Caso a peça siga para a *Estação de Inspeção*, um dos sensores será sensibilizado: *ok* indicando que a peça está boa, e *nok*, indicando que a peça não apresenta padrões de qualidade aceitáveis. Após a inspeção, a peça segue para a próxima estação.

Neste caso, tem-se uma possibilidade de escolha na modelagem do sistema. Pode-se tanto modelar um evento para cada atuador, quanto introduzir um grau superior de abstração, pela associação de atuadores e sensores na composição de eventos controláveis. Obviamente, o uso de eventos controláveis diminui o número de estados e eventos do modelo global, o que é muito mais interessante do ponto de vista computacional.

Pode-se trabalhar inicialmente com dois modelos: um para a chegada das peças e outro para a inspeção das mesmas. A figura 6.2 apresenta os modelos para a chegada de peças, enquanto que a figura 6.3 apresenta o modelo para a *Estação de Inspeção*.



figura 6.2: Modelos para a chegada de peças à Estação de Inspeção



figura 6.3: Modelo para a inspeção de peças

Nos modelos para a chegada de peças (figura 6.2) temos os seguintes eventos controláveis: *chegar peça A* e *chegar peça B*. Esta abstração é possível de ser feita dada a característica do sistema, que permite a associação de atuadores (A_A e A_B) e sensores (S_A e S_B), compondo assim estes eventos controláveis. Nota-se que a ocorrência destes eventos será de responsabilidade da planta. O mesmo ocorre com o evento *selecionar* no modelo que descreve a inspeção.

Os eventos S_{Aoff} , S_{Boff} , S_{Ooff} , ok e nok são eventos não-controláveis.

Abaixo é apresentado um quadro que mostra a composição dos eventos controláveis.

Evento	Habilitar	Desabilitar	Ocorrer
Chegar peça A	Recuar atuador A_A	Estender atuador A_A	S_{Aon}
Chegar peça B	Recuar atuador A_B	Estender atuador A_B	S_{Bon}
Selecionar	Girar atuador A_x para a direita	Girar atuador A_x para a esquerda	S_{Ion}

O modelo sugerido para a "planta livre" é descrito na figura 6.4, resultado do produto síncrono entre os modelos das figuras 6.2 e 6.3.

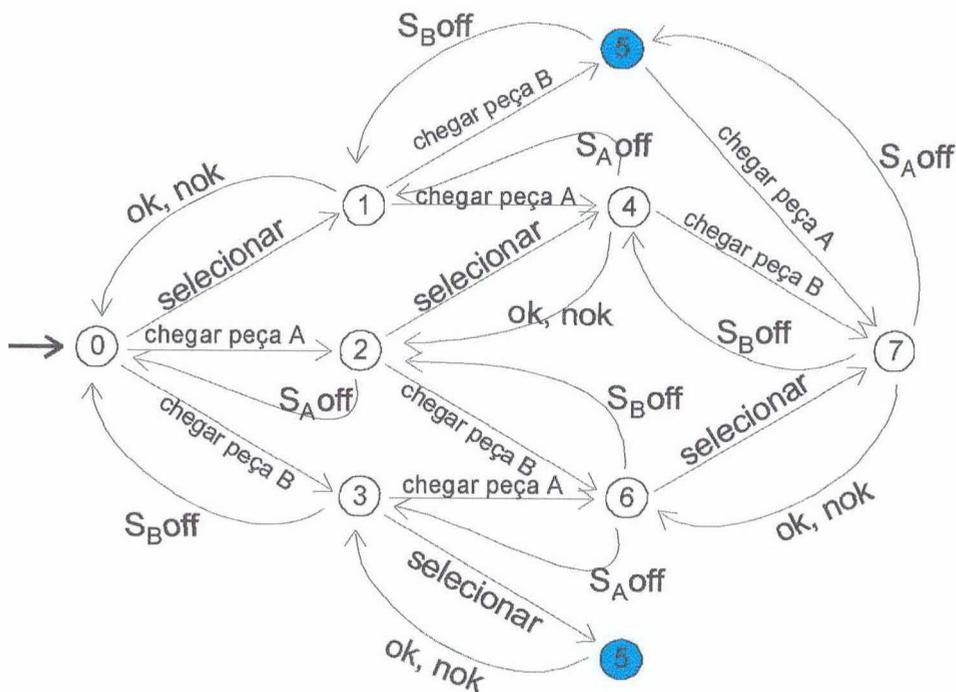


figura 6.4: Comportamento livre da Estação de Inspeção

Uma possível especificação para este exemplo seria a de "inspeccionar todas as peças vindas da esteira A". A solução completa não será mostrada aqui já que o objetivo é apenas estudar a modelagem do sistema.

Alguns aspectos podemos notar com relação a este modelo:

- É fácil verificar que, se fossem utilizados apenas eventos forçáveis, o modelo cresceria bastante, visto que todas as ações dos atuadores deveriam ser descritas;

- Para a solução de controle, dada uma determinada especificação, pode-se tanto utilizar a teoria clássica quanto a extensão apresentada no capítulo 4, visto que não existem eventos forçáveis neste modelo;
- Para uma implementação prática, a arquitetura proposta se mostra válida para todos os seus módulos, visto que existe a necessidade de um agente externo controlador (módulo intermediário) que comande a execução dos atuadores. Sem este agente, a habilitação ou desabilitação dos eventos controláveis nunca ocorreria;
- A existência de eventos controláveis só ocorreu, visto que a planta permite que se faça associações entre atuadores e sensores, de forma a compor os eventos controláveis. Deve-se tomar bastante cuidado para não se confundir simples associações de eventos com eventos controláveis. Tal evento, como descrito no capítulo 3, possui características particulares, que é a existência de um comando para habilitação e desabilitação do mesmo, além do fato de que a planta é a geradora deste evento.

6.3 Uma Estação de Rejeição de Pallets

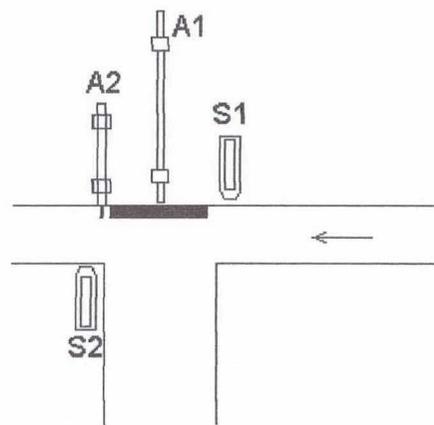


figura 6.5: Desenho esquemático da Estação de Rejeição

A Estação de Rejeição apresentada na figura 6.5 possui o seguinte funcionamento: Enquanto não chega nenhum *pallet*, o atuador A_2 pode estender e recolher livremente.

Também um *pallet* pode passar livremente pela estação, sem que seja bloqueada ou rejeitada. Algumas restrições físicas já foram incluídas no modelo inicial da estação:

- Um *pallet* só poderá ser rejeitado se ele antes tiver sido bloqueado pela pelo atuador A_2 e tiver passado totalmente pelo sensor S_1 .
- Se um *pallet* estiver passando pelo sensor S_2 , os atuadores A_1 e A_2 não podem ser acionados.
- Quando um *pallet* passa por S_1 e antes que ele comece a passar por S_2 chega um novo *pallet* em S_1 , não se pode mais acionar o atuador A_2 , pois fatalmente o êmbolo pressionaria o segundo *pallet* contra a parede lateral da célula, danificando o êmbolo.

O modelo da *Estação de Rejeição* pode ser visto na figura 6.6.

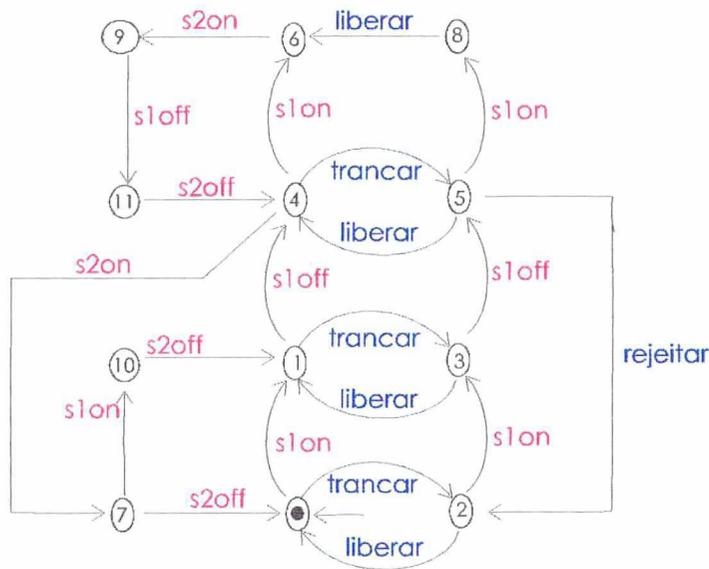


figura 6.6: Modelo que representa o comportamento da Estação de Rejeição

No modelo apresentado na figura 6.6 existem apenas eventos não-controláveis e eventos forçáveis. São eventos não-controláveis: $S1on$, $S1off$, $S2on$ e $S2off$. Os eventos forçáveis são os seguintes: *trancar* (estender atuador A_2), *liberar* (recuar atuador A_2), e *rejeitar* (estender e depois recuar o atuador A_1).

A *Estação de Rejeição* é um exemplo bastante peculiar, onde compor eventos para formar eventos controláveis é bastante complexo ou quase impossível. A composição de eventos neste caso, ou leva a uma perda de informações, ou a uma situação indesejada, como prensar um *pallet* contra a parede da esteira, na qual não é possível modelar restrições.

Para exemplificar, supõe-se descrever um evento controlável *passar pallet* com a seguinte composição:

habilitar \Rightarrow recuar o atuador A_2

desabilitar \Rightarrow estender o atuador A_2
 ocorrer \Rightarrow sinal S_{2off}

Supõe-se também descrever um evento forçável *rejeitar* que seja a composição dos comandos “estender atuador A_2 ” e “estender e depois recuar o atador A_1 ”.

Uma possível modelagem utilizando estes eventos poderia ser aquela descrita na figura 6.7.

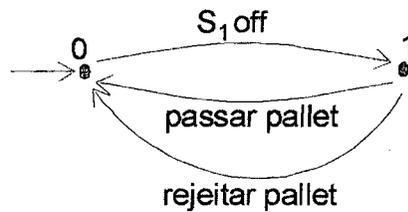


figura 6.7: Modelo para a Estação de Rejeição utilizando o evento controlável “passar pallet”

Duas considerações podem ser feitas a respeito deste modelo:

- No estado 1, caso seja habilitado o evento *passar pallet* e forçada a ocorrência do evento *rejeitar pallet*, pode haver um conflito de comandos, já que, enquanto o primeiro recua o atuador A_2 , o segundo evento tenta estender o mesmo atuador;
- Não existe a descrição completa da planta, portanto a perda de informações pode levar a erros ou danos. Por exemplo, fisicamente pode ocorrer a sequência $S_{1on} \Rightarrow S_{1off} \Rightarrow S_{2on} \Rightarrow S_{1on}$. Esta seqüência levaria o modelo da figura 6.7 ao estado 1, podendo, desta forma, estender o atuador A_1 e causar danos ao sistema, visto que os dois pallets que chegaram à estação seriam prensados contra a parede da estação.

Uma outra modelagem possível, utilizando os mesmos eventos, é apresentada na figura 6.8.

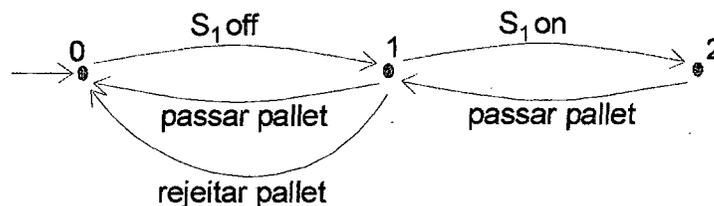


figura 6.8: Um novo modelo utilizando o evento controlável “passar pallet”

No modelo apresentado na figura 6.8, além do conflito de comandos, descrito no modelo anterior, pode também acontecer danos ao sistema. Isto aconteceria caso dois pallets chegassem juntos na estação, formando a seguinte seqüência de sinais: $S1off \rightarrow S1on \rightarrow S2off$. Isto levaria o sistema ao estado 1, podendo ocorrer o evento *rejeitar pallet*, o que levaria o atuador A_1 a prensar o segundo pallet contra a parede da estação, danificando o sistema.

Enfim, um tempo muito grande foi gasto na tentativa de se chegar a um modelo simples utilizando eventos controláveis. Em todas estas tentativas, encontrou-se pequenas falhas que impossibilitavam o uso do modelo. Já o modelo da figura 6.6 foi conseguido num tempo curto e, apesar do tamanho, descreve integralmente o funcionamento da estação, possibilitando um estudo preciso no controle deste sistema.

Ao se controlar este sistema, se faz necessário o uso da extensão à Teoria Clássica proposta por Golaszewski e Ramadge, descrita no capítulo 4, visto que, neste caso, a Teoria Clássica por si própria poderia fornecer resultados não coerentes, como visto no capítulo 3. Isto devido ao fato de que, no modelo proposto, encontram-se eventos forçáveis, o que exige a presença de um agente externo que seja capaz de gerar estes eventos, como descrito nos capítulos 3 e 4.

6.4. Uma Estação de Carregamento de Pallets

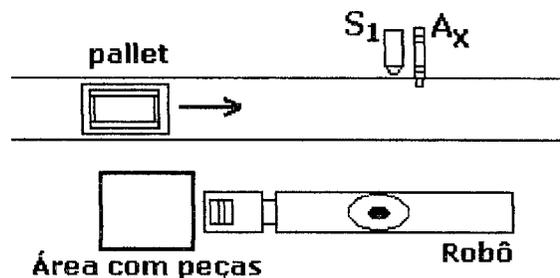


figura 6.9: Desenho esquemático da Estação de Carregamento de Pallets

A Estação de Carregamento de Pallets possui um funcionamento muito simples: um pallet, ao chegar à estação, pode passar livremente sem ser carregado, ou ser bloqueado e então carregado. A cada carregamento, o robô pega uma peça na área onde estas são estocadas (área com peças) e a leva até a esteira, colocando-a em cima do pallet.

Este caso é mais um onde se pode escolher entre utilizar somente eventos

forçáveis, ou utilizar também eventos controláveis. Um modelo bastante simples para esta estação é apresentado na figura 6.10.

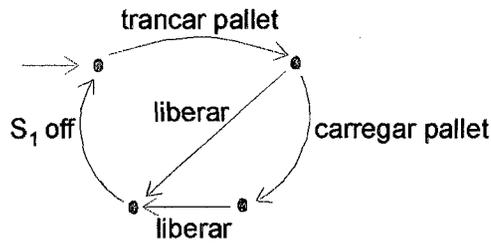


figura 6.10: Modelo para a Estação de Carregamento de Pallets

Nota-se no modelo da figura 6.10 o uso dos dois tipos de eventos no mesmo modelo (controláveis e forçáveis). O evento *trancar pallet* é descrito como controlável, enquanto que os eventos *liberar* e *carregar pallet* são eventos forçáveis. Obviamente, o evento *S₂off* é tipicamente um evento não-controlável.

O evento *trancar pallet* é composto da seguinte forma:

- habilitar ⇒ estender o atuador A_x
- desabilitar ⇒ recuar o atuador A_x
- ocorrer ⇒ Sensor S_1 sensibilizado

Já o evento forçável *carregar pallet* é uma composição de outros comandos, como: estender braço do robô, pegar peça, girar robô, colocar robô sobre o pallet, abrir a garra, retornar para a posição inicial, etc. Esta composição foi feita apenas para simplificar o modelo.

O evento forçável *liberar* é mapeado diretamente no comando "recuar atuador A_x ".

É importante novamente ressaltar que neste exemplo, ao se fazer especificações para um determinado controle, é importante que se use a extensão de Golaszewski e Ramadge descrita no capítulo 4. Como descrito anteriormente, o uso da Teoria Clássica neste caso pode levar a resultados não compatíveis com a prática, dependendo das especificações. Além disso, a implementação prática do resultado necessitará de um agente externo conforme descrito no capítulo 3, o que não está previsto na Teoria Clássica de controle de SED's.

Neste capítulo foram estudados alguns exemplos, mostrando a modelagem de sistemas a eventos discretos utilizando eventos forçáveis e eventos controláveis, tal como apresentados nos capítulos 3 e 4.

No próximo capítulo será apresentada a conclusão final deste trabalho, juntamente com as perspectivas e propostas para trabalhos futuros.

CONCLUSÕES E PERSPECTIVAS

Este trabalho permitiu mostrar que a aplicação da Teoria Clássica de Controle de Sistemas a Eventos Discretos, desenvolvida por Wonham e Ramadge, em modelos que fogem às características definidas na própria Teoria, pode apresentar resultados não coerentes na prática. Isto devido ao fato de que tal teoria apresenta a "planta" ou o sistema a ser controlado como sendo o *gerador de todos os eventos do sistema*. Isto nem sempre é verdade, visto que muitos eventos necessitam de um agente externo capaz de forçar a ocorrência dos mesmos.

A metodologia para modelagem de sistemas aqui apresentada mostra a necessidade de existência deste "agente controlador" e como tratá-lo formalmente. Para isso utilizou-se a extensão à Teoria Clássica proposta por Golaszewski e Ramadge, onde é incluída a noção de eventos forçáveis.

Além disso, a arquitetura física proposta para o controlador é coerente com a metodologia apresentada, o que auxilia a implementação prática da solução teórica.

O "módulo intermediário", componente da arquitetura física proposta, é o responsável pela aplicação ou não dos eventos forçáveis. Constitui-se desta forma, o "agente controlador" anteriormente citado. O supervisor, também componente da arquitetura proposta, continua sendo o elemento que *habilita e desabilita eventos*. A diferença está no fato de que o cálculo da máxima linguagem controlável inclui a noção de eventos forçáveis, como descrito no capítulo 4.

De uma forma geral, a arquitetura proposta pretende nortear àqueles que desejam construir sistemas de controle, incluindo-se aqui hardware e software. Não se teve a intenção de especificar detalhadamente cada módulo, mas de fornecer a função que cada elemento deve exercer dentro do sistema de controle.

Ao longo do desenvolvimento deste trabalho, ficou clara a idéia de que o controle de Sistemas a Eventos Discretos vai muito além da resolução de problemas utilizando-se a Teoria Clássica. O estudo prático, realizado sobre a célula de manufatura existente no LAI (Laboratório de Automação Industrial), demonstrou a necessidade de um estudo amplo, que abranja não só a Teoria de Controle de Sistemas a Eventos Discretos, mas também outras áreas como Sistemas em Tempo Real, Engenharia de Software, Eletrônica, elementos de Automação Industrial, entre outras.

Algumas atividades poderiam ser realizadas afim de dar continuidade ao trabalho aqui proposto:

- Desenvolvimento de um software que implemente fisicamente a arquitetura aqui apresentada;
- Analisar a metodologia apresentada conforme os conceitos de controle hierárquico, controle modular, etc;
- Estudo de um sistema de decisão para o módulo intermediário, seja ele baseado em Inteligência Artificial, Lógica Fuzzy, etc;
- Estudo da influência do tempo sobre a metodologia aqui apresentada;

REFERÊNCIAS BIBLIOGRÁFICAS

- [AC86] **Alami, R.; Chocon, H.:** "NNS, A Knowledge-Based On-Line System for an Assembly Workcell", IEEE Proc. Int. Conference on Robotics and Automation, pp. 603-609 (1986)
- [BCJ89] **Benhabib, B.; Chen, C.Y.; Johnson:** "An Integrated Manufacturing Workcell Management System", ASME Manufacturing Review, vol.2, n°. 4, Dec. 1989, pp. 266-276 (1989)
- [BGK89] **Brandt, R.D.; Garg, V.; Kumar, R.:** "Formulas for Calculating Supremal Controllable and Normal Sublanguages". 27th Annual Allerton Conference on Communication, Control and Computing (1989)
- [Br84] **Bruno, G.:** "Communication and Programming Issues in Robotics Manufacturing Cells". IEEE Proc. Int. Conference on Robotics, pp. 361-367 (1984)
- [BWB91] **Brandin, B.A.; Wonham, W.M.; Benhabib, B.:** "Discrete Event Supervisory Control Applied to the Management of Manufacturing Workcells". 7th International Conference on Computer Aided Manufacturing Engineering, V.C. Venkatesh and J.A. McGeough Eds., Elsevier, pp. 527-536 (1991)
- [BWB92] **Brandin, B.A.; Wonham, W.M.; Benhabib, B.:** "Manufacturing Cell Supervisory Control - A Timed Discrete Event System Approach". IEEE, Conference on Robotics and Automation, Nice, France (1992)
- [Cu96] **Cury, J.E.R.:** "Abordagem Autômatos", notas de aula (1996)
- [GR87] **Golaszewski, C.H.; Ramadge, P.J.:** "Control of discrete event processes with forced events". 28th Conference on Decision and Control, Los Angeles, pp. 247-251, December (1987)
- [KG95] **Kumar, R.; Garg, V.K.:** "Modeling and Control of Logical Discrete Event Systems". Kluwer Academic Publishers (1995)
- [LW88] **Lin, F.; Wonham, W.M.:** "On observability of discrete-event systems". Information Sciences, 44(3):173-198 (1988)

- [ME97] **Martins, E.D.; Cury, J.E.R.:** "Uma Arquitetura para Implementação de Controle Supervisório de Sistemas a Eventos Discretos". III Simpósio Brasileiro de Automação Inteligente (1997)
- [RW87] **Ramadge, P.J.; Wonham, W.M.:** "Supervisory Control of a class of discrete event process". SIAM Journal of Control and Optimization, 25(1):206-230 (1987)
- [RW92] **Rudie, K.; Wonham, W.M.:** "Think globally, act locally: decentralized supervisory control". IEEE Transactions on Automatic Control, 37(11):1692-1708, November (1992)
- [VY89] **Valavanis; Yuan:** "Hardware and Software for Intelligent Robotic Systems". Conference on Robotics and Automation, JISTA Journal (1989)
- [WR88] **Wonham, W.M.; Ramadge, P.J.:** "Modular Supervisory Control of Discrete Event Systems". Mathematics of Control Signals and Systems, 1(1):13-30 (1988)
- [ZC94] **Ziller, R.M.; Cury, J.E.R.:** "On the Supremal L_m -closed and the Supremal L_m -closed and L -controllable Sublanguages of a Given Language". Lecture Notes in Control and Information Sciences 199, London, pp. 80-85 (1994)
- [Zh87] **Zong, H.:** "Control of discrete-event systems: decentralized and hierarchical control". M. A. Sc. Thesis, Department of Electrical Engineering, University of Toronto (1987)