

**YURI GARCIA SILVA**

**CONTROLE SUPERVISÓRIO MODULAR LOCAL DE  
SISTEMAS DE VEÍCULOS AUTO-GUIADOS**

**FLORIANÓPOLIS  
2010**



**UNIVERSIDADE FEDERAL DE SANTA CATARINA**

**PROGRAMA DE PÓS-GRADUAÇÃO  
EM ENGENHARIA DE AUTOMAÇÃO E SISTEMAS**

**CONTROLE SUPERVISÓRIO MODULAR LOCAL  
DE SISTEMAS DE VEÍCULOS AUTO-GUIADOS**

Dissertação submetida à  
Universidade Federal de Santa Catarina  
como parte dos requisitos para a  
obtenção do grau de Mestre em Engenharia  
de Automação e Sistemas.

Yuri Garcia Silva

Florianópolis, junho de 2010



# **CONTROLE SUPERVISÓRIO MODULAR LOCAL DE SISTEMAS DE VEÍCULOS AUTO-GUIADOS**

Yuri Garcia Silva

‘Esta Dissertação foi julgada adequada para obtenção do Título de Mestre em Engenharia de Automação e Sistemas, Área de Concentração em Controle, Automação e Sistemas, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina.’

---

Max Hering de Queiroz, Dr.  
Orientador

---

José Eduardo Ribeiro Cury, Dr.  
Coordenador do Programa de Pós-graduação em Engenharia de  
Automação e Sistemas

Banca Examinadora:

---

Max Hering de Queiroz, Dr. – UFSC  
Presidente

---

José Eduardo Ribeiro Cury, Dr. – UFSC  
Co-orientador

---

André Bittencourt Leal, Dr. – UDESC

---

Jean-Marie Alexandre Farines, Dr. – UFSC

---

Leandro Buss Becker, Dr. – UFSC



## **AGRADECIMENTOS**

À Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo apoio financeiro.

Aos professores Max Hering de Queiroz e José Eduardo Ribeiro Cury que desempenharam um grande papel durante o decorrer deste trabalho.

Aos colegas do LTIC pela agradável convivência.

À família pelo incentivo.





Resumo da Dissertação apresentada à UFSC como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia de Automação e Sistemas.

## **CONTROLE SUPERVISÓRIO MODULAR LOCAL DE SISTEMAS DE VEÍCULOS AUTO-GUIADOS**

Yuri Garcia Silva

Junho/2010

Orientador: Max Hering de Queiroz, Dr.

Co-orientador: José Eduardo Ribeiro Cury, Dr.

Área de Concentração: Controle, Automação e Sistemas.

Palavras-chave: controle supervísório; sistemas a eventos discretos; controle modular local; autômatos; sistemas de controle; veículos auto-guiados; AGV.

Número de páginas: xxii + 72.

Este trabalho apresenta a aplicação da abordagem modular local da teoria de controle supervísório no controle de um sistema de veículos auto-guiados. Nesta abordagem busca-se alcançar um conjunto de supervisores não bloqueantes e minimamente restritivos que proíbem a ocorrência das seqüências de eventos indesejáveis no sistema. São discutidos os processos de síntese e implementação dos supervisores, os quais envolvem soluções que visam gerenciar o tráfego dos veículos, gerenciar a bateria dos veículos e coordenar a interação entre os pontos de transferências de materiais e os veículos. A solução de controle apresentada é feita para uma bancada experimental de um sistema com dois AGVs e duas estações de transferência de materiais, embora a solução possa ser generalizada para sistemas de maior porte. A etapa de implementação dos supervisores também é abrangida. O controle é implementado através de uma estrutura hierárquica de três níveis que objetiva manter as características propostas pela teoria de controle supervísório. Para avaliar a correção do controlador, foi desenvolvido um ambiente gráfico de emulação que permite prever as respostas das ações de controle dos supervisores através de animações gráficas. Constatou-se pela emulação que a abordagem aplicada realiza o controle do sistema sem a ocorrência de bloqueios e mantém a flexibilidade operacional.



Abstract of dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Automation and Control Engineering.

## **LOCAL MODULAR SUPERVISORY CONTROL OF AUTOMATED GUIDED VEHICLE SYSTEMS**

**Yuri Garcia Silva**

June /2010

Advisor: Max Hering de Queiroz, Dr.

Co-Advisor: José Eduardo Ribeiro Cury, Dr.

Area of Concentration: Control, Automation and Systems.

Keywords: supervisory control; discrete event systems; local modular control; automata; control systems; automated guided vehicles; AGV.

Number of pages: xxii + 72.

This work presents the application of the local modular approach of the supervisory control theory to control an automated guided vehicle system. This approach seeks to achieve a set of nonblocking and minimally restrictive supervisors that forbid the occurrence of undesirable sequences of events in the system. It is discussed the processes of synthesis and implementation of supervisors, which involve solutions that aim to manage the traffic of vehicles, to manage the vehicles battery charge and to coordinate the interaction between the shipment points of materials and the vehicles. The control solution presented is made for an AGV system testbed, composed of two AGVs and two material transfer stations, although the solution can be generalized to larger systems. The implementation of supervisors is also covered. The control is implemented through a three level hierarchy structure that aims to maintain the characteristics proposed by the supervisory control theory. To evaluate the controller correctness, it was developed a graphical environment of emulation that allows predicting the responses to the control actions through graphic animations. It was verified by emulation that the applied approach performs the system control without deadlocks or livelocks and preserves the operational flexibility.



## SUMÁRIO

1	Introdução .....	1
2	Teoria de controle supervisorio de sistemas a eventos discretos.....	5
2.1	Geradores como modelos para sistemas a eventos discretos .....	5
2.2	Teoria de Controle Supervisorio .....	6
2.3	A célula flexível de manufatura.....	10
2.4	Modelagem dos subsistemas da CFM.....	12
2.5	Modelagem das especificações .....	15
2.5.1	Síntese dos supervisores modulares locais da CFM.....	19
2.6	Redução e implementação de supervisores.....	21
2.7	Conclusão do capítulo.....	21
3	Sistemas de veículos auto-guiados .....	23
3.1	Estrutura de sistema de AGV .....	23
3.1.1	Veículos .....	24
3.1.2	Sistema de navegação .....	25
3.1.3	Controlador .....	27
3.1.4	Sistema de comunicação .....	28
3.2	Problemas de controle de sistema de AGV.....	28
3.2.1	Despacho e roteamento dos veículos .....	29
3.2.2	Gerenciamento de tráfego .....	29
3.2.3	Gerenciamento da carga da bateria dos AGVs.....	30
3.2.4	Sincronização entre operações .....	31
3.3	Conclusão do capítulo.....	31
4	Aplicação da TCS a um sistema de AGV .....	33
4.1	O sistema de AGV .....	33
4.2	Modelagem do sistema de AGV .....	36
4.3	Modelagem das especificações do sistema de AGV.....	39
4.4	Síntese dos supervisores do sistema de AGV .....	43

4.5	Emulação dos supervisores.....	45
4.6	Expansão para sistemas de grande porte .....	45
4.7	Conclusão do capítulo .....	46
5	Implementação de supervisores em sistema de AGV .....	47
5.1	Estrutura genérica de controle .....	47
5.2	Estrutura de controle aplicada ao sistema de AGV .....	48
5.3	Ambiente de emulação do controlador.....	54
5.4	Implementação da bancada experimental do sistema de AGV	55
5.5	Conclusão do capítulo .....	56
6	Conclusão.....	59
	Apêndice .....	63

## LISTA DE ABREVIATURAS

AGV	<i>automated guided vehicle</i> (veículo auto-guiado)
CFM	célula flexível de manufatura
CLP	controlador lógico programável
e.r.a	em relação a
SED	sistema a eventos discretos
TCS	teoria de controle supervisão





## LISTA DE SÍMBOLOS

$\parallel$	operador de produto síncrono
$\bar{L}$	prefixo-fechamento de $L$
$\emptyset$	conjunto vazio
$K, L$	linguagens
$\Sigma$	alfabeto
$\Sigma_c$	conjunto de eventos controláveis
$\Sigma_u$	conjuntos de eventos não controláveis
$\Sigma^*$	conjunto de todas as cadeias finitas de $\Sigma$
$\varepsilon$	cadeia vazia
$G$	gerador
$L(G)$	linguagem gerada por $G$
$L_m(G)$	linguagem marcada por $G$
$E$	especificação
$S$	supervisor
$S_{loc}$	supervisor modular local
$S_{red}$	supervisor modular local reduzido
$S/G$	supervisor $S$ controlando $G$
$SupC(K, G)$	suprema sublinguagem de $K$ controlável e.r.a $G$



## LISTA DE FIGURAS

Figura 2.1. Visualização gráfica de um gerador.....	6
Figura 2.2. Estrutura em malha fechada do controle supervísório .....	7
Figura 2.3. Exemplificação de plantas locais .....	9
Figura 2.4. Célula Flexível de Manufatura.....	10
Figura 2.5. Geradores (a) $G_1$ e (b) $G_2$ .....	12
Figura 2.6. Gerador $G_3$ .....	14
Figura 2.7. Geradores (a) $G_4$ , (b) $G_5$ e (c) $G_6$ .....	14
Figura 2.8. Gerador $G_7$ .....	15
Figura 2.9. Gerador $E_1$ .....	15
Figura 2.10. Geradores (a) $E_{2a}$ , (b) $E_{2b}$ , (c) $E_{2c}$ e (d) $E_{2d}$ .....	16
Figura 2.11. Gerador genérico para $E_{3a}$ , $E_{3b}$ , $E_{3c}$ .....	17
Figura 2.12. Sequência de operações relacionada à especificação $E_{3a}$ .....	17
Figura 2.13. Gerador $E_{3d}$ .....	18
Figura 2.14. Geradores (a) $E_{4a}$ e (b) $E_{4b}$ .....	19
Figura 3.1. Tipos de AGVs: (a) reboque (JBT, 2009), (b) carga de unidade (ELETTRIC80, 2009), (c) linha de montagem (JBW, 2009).....	25
Figura 3.2. Tipos de AGVs: (a) carga de pallet (ELETTRIC80, 2009), (b) cargas leves (AUTOMOTION, 2009).....	25
Figura 4.1. Sistema de veículos auto-guiados .....	34
Figura 4.2. Metodologia para obtenção do sistema de controle .....	36
Figura 4.3. Gerador $G_{AGV1}$ .....	37
Figura 4.4. Gerador $G_{AGV2}$ .....	38
Figura 4.5. Geradores (a) $G_{CARGA}$ e (b) $G_{DESCARGA}$ .....	38
Figura 4.6. Geradores (a) $G_{SENT}$ , (b) $G_{SBAT1}$ , (c) $G_{SBAT2}$ .....	39
Figura 4.7. Especificação de exclusão mútua genérica .....	39
Figura 4.8. Especificação de exclusão mútua $E_{COL9}$ para a zona HKL .....	40
Figura 4.9. Especificação (a) $E_{MAN1}$ e especificação genérica para (b) $E_{MAN2}$ e $E_{MAN3}$ .....	40
Figura 4.10. Especificação genérica para $E_{MAN4}$ e $E_{MAN5}$ .....	41
Figura 4.11. Especificações (a) $E_{MAN6}$ e (b) $E_{MAN7}$ .....	41
Figura 4.12. Especificações (a) $E_{MAN8}$ e (b) $E_{MAN9}$ .....	41
Figura 4.13. Geradores intermediários para se obter $E_{MAN10}$ e $E_{MAN11}$ .....	42
Figura 4.14. Especificações (a) $E_{ROT1}$ e $E_{ROT2}$ .....	42
Figura 4.15. Especificações $E_{ROT3}$ ( $\langle AGV \rangle = 1$ ) e $E_{ROT4}$ ( $\langle AGV \rangle = 2$ ).....	43
Figura 5.1. Estrutura do sistema de controle (Queiroz, 2000).....	48

Figura 5.2. Estrutura do pseudo-código da implementação .....	49
Figura 5.3. Implementação do supervisor $S_{redMAN2}$ em C .....	51
Figura 5.4. Implementação do gerador $G_{CARGA}$ no nível sistema- produto.....	52
Figura 5.5. Trecho da implementação do gerador $G_{AGV1}$ no nível sistema-produto.....	53
Figura 5.6. Implementação das seqüências operacionais do gerador $G_{CARGA}$ .....	53
Figura 5.7. Ambiente de emulação do controlador .....	54
Figura 5.9. Arquitetura de implementação do sistema de AGV .....	56

## LISTA DE TABELAS

Tabela 2.1. Eventos da Célula Flexível de Manufatura.....	13
Tabela 2.2. Plantas locais para a CFM .....	20
Tabela 2.3. Número de estados dos geradores da síntese dos supervisores.....	20
Tabela 4.1. Plantas locais para o sistema de AGV .....	43
Tabela 4.2. Resumo da síntese dos supervisores do sistema de AGV...	44



# Capítulo 1

## Introdução

Manejo de materiais refere-se à função de transporte entre diferentes pontos, atendendo a requisitos de tempo, quantidade, seqüência e condições para reduzir o custo da produção. O mesmo não deve ser entendido exclusivamente como movimento dos materiais, pois envolve também os processos de empacotamento e armazenamento. Estima-se que o manejo de materiais demande entre 40 a 80 por cento do custo operacional total da produção (Meyers e Stephens, 2002). Diante disto, a automação preocupa-se em fornecer meios para reduzir estes custos sem perder a confiabilidade do sistema. Apesar da redução de custos frequentemente aparecer como o principal motivo para a automatização do manejo de materiais, não se pode excluir as justificativas relacionadas à qualidade, saúde, segurança, etc. (Kent e Williams, 1990).

Dentro do contexto da automação do manejo de materiais, os veículos auto-guiados (AGV) surgem como alternativa importante para a movimentação de materiais. Estes veículos não requerem um operador humano para conduzi-los e são empregados em ambientes que possuem um fluxo repetitivo de deslocamento de materiais entre pontos previamente determinados. A implantação e manutenção de sistemas de AGV são bastante dispendiosas, no entanto, quando o custo é justificado, estes sistemas oferecem uma série de vantagens. Pode-se citar, por exemplo, a flexibilidade oferecida pela estrutura do sistema, que torna fácil e rápida a alteração ou expansão das rotas existentes (HKSYSTEMS, 2009). Outra vantagem é que a falha de um veículo pode ser isolada do sistema e o fluxo de transporte é mantido enquanto o mesmo é reparado. Deve-se lembrar também que os AGVs são mais seguros quando comparados com veículos operados por humanos, já que os AGVs não estão sujeitos aos desgastes físico – diferente de desgaste mecânico – e psicológico sofridos pelos humanos quando operam veículos de transportes. Isto é um fator extremamente relevante, já que os acidentes ocorridos durante o manejo de materiais representam metade dos acidentes industriais (Meyers e Stephens, 2002).

A implantação bem sucedida de sistemas de AGV ao longo das décadas tem expandido sua aplicação em diversos setores da produção

industrial, comércio e serviço. A característica de cada um destes setores exige soluções próprias para lidar com o transporte de materiais. No ambiente industrial os AGVs são aplicados na conexão de diferentes áreas de trabalho, coleta de pedidos, estocagem e linhas de montagem (Schulze, et al., 2008). Em terminais de containers automatizados, os AGVs participam do processo de carregamento e descarregamento de navios (Günther e Kim, 2005). Não se excluem também as aplicações em escritórios e hospitais. Em escritórios estes dispositivos são utilizados para entregas de correspondências e pacotes leves. Já em algumas instalações hospitalares, observa-se o transporte de suprimentos cirúrgicos e alimentação com estes veículos (Groover, 2007). As aplicações não se limitam a estas e tendem a se propagar a outros setores.

Diante da relevância das aplicações dos AGVs, são válidas as pesquisas de novas soluções para o aprimoramento das tecnologias associadas aos mesmos. Este trabalho concentra-se na problemática de controle, a qual se preocupa em estabelecer uma política de controle que satisfaça as demandas de transporte de modo rápido e sem a ocorrência de conflitos entre os AGVs (Vis, 2006). Esta política deve realizar atividades como, por exemplo, despacho dos veículos, roteamento dos veículos, escalonamento das tarefas dos AGVs e posicionamento dos veículos ociosos. Um problema bastante discutido no meio científico, no que diz respeito à política de controle, refere-se ao gerenciamento de tráfego dos AGVs. Neste campo de estudo busca-se métodos de controle que permitam o deslocamento dos AGVs de modo a prevenir colisões entre veículos e *deadlocks*.

No gerenciamento de tráfego é comum a abordagem de controle por zonas, a qual proíbe o acesso mútuo de veículos em uma mesma zona. Evers e Koppers (1996) sugerem a utilização de semáforos para proporcionar o acesso de vários veículos simultaneamente numa mesma zona. De acordo com a quantidade e os locais de entradas e saídas de cada área, é estabelecida uma regra de aumento ou redução do número de vagas de veículos autorizados a transitar pela área por meio de semáforos. Hsieh e Kang (1998) apresentam um método para traduzir um modelo em alto nível do *layout* de percurso de sistemas de AGVs para uma lógica de controle em baixo nível utilizando redes de Petri como interface de controle entre esta lógica e o sistema real. É utilizada uma abordagem modular para resolver problemas de controle de partes específicas do percurso – segmentos, bifurcações, junções e interseções – através da abordagem tradicional de controle de zonas para prevenção de colisões. Com a união destas partes se obtém um modelo completo para o layout proposto.



Moorthy (2003) apresenta um algoritmo para a prevenção de *deadlocks* cíclicos, o qual ocorre quando um veículo *A* necessita acessar uma zona utilizada pelo veículo *B* e este quer acessar a zona acessada pelo veículo *C* que por sua vez espera pela liberação da zona do veículo do solicitante inicial, formando um ciclo. Também é discutida a ocorrência de *deadlocks* cíclicos múltiplos, que consiste na existência de vários *deadlocks* cíclicos conectados pela necessidade do uso de um recurso comum. O algoritmo estima dinamicamente a posição dos veículos antecipadamente em uma zona e verifica a formação de *deadlocks* cíclicos. Reveliotis (2000) propõe um algoritmo de roteamento em tempo real que busca o percurso livre de *deadlock* incrementalmente, a cada mudança de zona. As decisões de roteamento baseiam-se em critérios de segurança e desempenho.

A dinâmica dos sistemas de AGV, analisada sob o ponto de vista de controle, pode ser interpretada como sendo da classe de sistemas a eventos discretos (SED). Os SEDs são sistemas cuja evolução se caracteriza pela ocorrência de eventos que ocorrem abruptamente e, possivelmente, em intervalos de tempo irregulares e desconhecidos (Ramadge e Wonham, 1989).

O controle de SEDs é um campo que vem sendo estudado nas últimas décadas e não apresenta um modelo matemático consolidado para tratar suas especificidades. Várias são as abordagens existentes na literatura. Cadeias de Markov, Redes de Filas e Simulação aparecem como forma de avaliação de desempenho para estes modelos. Lógica Temporal, Redes de Petri e Teoria de Controle Supervisório lidam com a análise e síntese de controle para SEDs (Cassandras e Lafortune, 1999).

A Teoria de Controle Supervisório (TCS) (Ramadge e Wonham, 1989) apresenta uma particularidade especial: oferece um método de síntese automática para obter um supervisor ótimo para o sistema. Em outras palavras isto significa dizer que, a partir das especificações fornecidas, é alcançada uma solução minimamente restritiva e não bloqueante para controlar o sistema. Em resumo, o supervisor age no sistema proibindo a ocorrência de ações que levariam a situações indesejáveis. Como extensão da TCS há a abordagem modular local (Queiroz e Cury, 2002), a qual propõe a síntese de vários supervisores, que atuam cooperativamente, para tratar problemas específicos do sistema. Esta abordagem, quando aplicável, reduz a complexidade computacional para a síntese dos supervisores em relação à TCS. Conforme visto, sistemas de AGV lidam com uma diversidade de problemas de controle, e a TCS, além de apresentar estas boas propriedades, é capaz de abordar vários destes problemas.

O objetivo geral deste trabalho é utilizar a TCS, em sua abordagem modular local, para resolver o problema de controle de sistemas de AGV. Para alcançar este objetivo, conta-se com uma bancada experimental microcontrolada representativa de um sistema de AGV real que é composta por dois veículos e duas estações de transferências de materiais, sendo uma de coleta e outra de entrega.

Para este sistema, são apresentadas soluções para evitar colisões entre os veículos, *deadlocks*, regras de roteamento para os veículos – determinação de percurso seguro que leve o veículo ao seu objetivo, neste caso desconsiderando os requisitos temporais –, gerenciamento da bateria dos veículos e questões de coordenação entre os veículos e as estações de transferência de materiais.

Apesar das soluções propostas serem geradas para este sistema em particular, busca-se obter um conjunto de especificações de controle cujos modelos são bases para a construção de soluções para sistemas de AGV de grande porte.

O trabalho ultrapassa a solução teórica e apresenta a implementação dos supervisores modulares locais em linguagem C para a bancada experimental citada. A implementação é feita de acordo com uma estrutura genérica hierárquica para controle supervisorio (Queiroz e Cury, 2002).

O conteúdo deste documento é organizado como segue. No capítulo 2 a TCS é apresentada a partir da solução do problema real de uma célula flexível de manufatura. No capítulo 3 são apresentados conceitos sobre sistemas de AGV e uma descrição de como a problemática de controle dos mesmos é tratada. No capítulo 4 é apresentado o sistema de AGV estudado e a descrição do processo de obtenção dos supervisores segundo a metodologia da TCS. A implementação dos supervisores sintetizados é descrita no capítulo 5. O trabalho é finalizado no capítulo 6, onde são expostas as conclusões e perspectivas futuras.

## Capítulo 2

# Teoria de controle supervisorio de sistemas a eventos discretos

Neste capítulo são explicados os conceitos da TCS (Ramadge e Wonham, 1989) e da abordagem modular local (Queiroz e Cury, 2002). Para exemplificar a teoria, é solucionado um problema real de controle de uma célula flexível de manufatura (CFM) (Silva e Queiroz, 2009), utilizando a abordagem modular local da TCS. Este problema foi anteriormente solucionado por (Klinge, 2007), porém a solução proposta por esta autora apresentava situações de bloqueio. Para o problema apresentado, são descritos os processos de modelagem e síntese dos supervisores modulares locais ótimos.

### 2.1 Geradores como modelos para sistemas a eventos discretos

Um sistema a eventos discretos (SED), cujo conjunto de eventos (alfabeto) seja  $\Sigma$ , pode ser representado por uma linguagem  $L \subseteq \Sigma^*$ , sendo  $\Sigma^*$  o conjunto de todas as cadeias finitas formadas por elementos de  $\Sigma$ , inclusive a cadeia vazia  $\varepsilon$ . A linguagem de um SED é, portanto, um subconjunto desta totalidade de possibilidades e retrata as seqüências finitas de eventos factíveis de ocorrer.

O gerador é definido por uma quintupla de forma  $G = (\Sigma, Q, \delta, q_0, Q_m)$  (Hopcroft e Ullman, 1979).  $\Sigma$  representa o conjunto de eventos,  $Q$  o conjunto de estados,  $q_0 \in Q$  o estado inicial,  $Q_m \subseteq Q$  o subconjunto de estados marcados e  $\delta: \Sigma \times Q \rightarrow Q$  a função de transição parcial definida em cada estado de  $Q$  para um subconjunto de  $\Sigma$ .

O gerador está associado às linguagens gerada  $L(G)$  e marcada  $L_m(G)$ . A primeira representa todas as seqüências possíveis de ocorrer a partir do estado inicial. A segunda indica as seqüências que alcançam um estado marcado a partir do estado inicial, o que significa conclusão de uma tarefa.

Os geradores podem ser graficamente visualizados por diagramas de transição de estado, que são grafos direcionados. Os vértices simboli-

zam os estados e os arcos, associados com as respectivas funções de transições parciais, a ocorrência dos eventos. Uma flecha aponta para o estado inicial e os estados marcados são desenhados como círculos duplos.

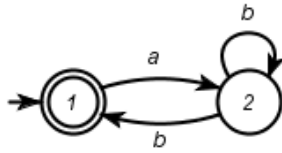


Figura 2.1. Visualização gráfica de um gerador

Se a partir do estado inicial  $q_0$  de um gerador  $G$  existe uma seqüência de eventos que leva a um estado  $q$ , tal estado é chamado de acessível. Diz-se que  $G$  é acessível se todos os seus estados são acessíveis. Tem-se também que o estado  $q$  de um gerador é co-acessível se a partir de  $q$  existe uma seqüência de eventos que leva a um estado marcado.  $G$  é co-acessível se todos os seus estados são co-acessíveis. Sendo acessível e co-acessível, o gerador  $G$  é chamado de aparado ou *trim*.

Um gerador é não bloqueante caso todos os estados acessíveis sejam co-acessíveis. O bloqueio ocorre quando o gerador não consegue progredir até um estado marcado, ou porque o mesmo avançou até um estado não marcado onde não há transições de saída – *deadlock* – ou porque sua evolução está restrita a um conjunto de estados não marcados a partir dos quais não existe uma seqüência de eventos que leve a um estado marcado – *livelock*.

Para representar a evolução paralela de geradores, recorre-se a composição síncrona dos mesmos (Cassandras e Lafortune, 1999). Sejam os geradores  $G_1$  e  $G_2$ , a composição síncrona destes geradores, obtida pela evolução concorrente dos dois geradores sincronizando-se os eventos compartilhados, é simbolizada por  $G_1 \parallel G_2$ . A composição síncrona é uma operação comutativa e associativa e quando se deseja compor vários geradores a operação pode ser feita repetitivamente aos pares.

## 2.2 Teoria de Controle Supervisório

A TCS, proposta por Ramadge e Wonham (1989), oferece um método formal, baseado na teoria de autômatos e linguagens, para gerar um supervisor que rege o sistema de acordo com um mapa de leis de controle. O supervisor acompanha seqüencialmente os eventos que ocorrem na planta, representada de forma abstrata por geradores, e determina quais dos eventos devem ser evitados em certos instantes, a fim de ga-

rantir a segurança e a execução especificada para o sistema. Este método atinge o controle mais flexível para o sistema, visto que permite a ocorrência de todas as ações que não se oponham ao comportamento especificado.

A interação do supervisor com a planta ocorre através da estrutura em malha fechada ilustrada pela figura 2.2. O supervisor é atualizado sempre que ocorre um evento na planta e a partir desta resposta ele desabilita os eventos indesejáveis para o sistema.

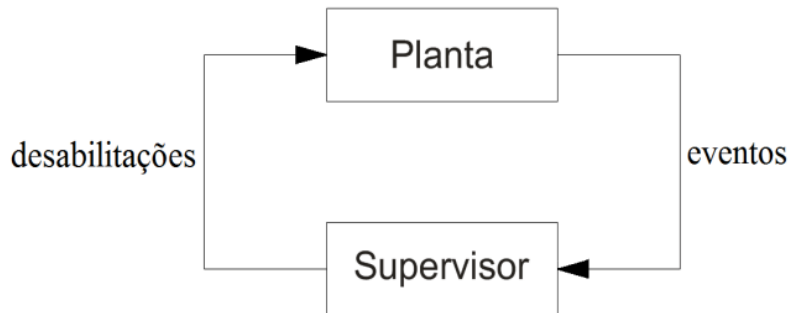


Figura 2.2. Estrutura em malha fechada do controle supervisorio

O supervisor  $S$  é representado por um gerador cujos eventos estejam contidos no alfabeto de  $G$ . Para propósitos de controle, o conjunto dos eventos  $\Sigma$  do gerador é particionado em dois subconjuntos  $\Sigma_c$  e  $\Sigma_u$ , sendo  $\Sigma_c$  os eventos controláveis e  $\Sigma_u$  os não controláveis (Ramadge e Wonham, 1989). Os eventos controláveis podem ser habilitados ou desabilitados por  $S$ , ao passo que os não controláveis estão sempre habilitados. Para desabilitar os eventos controláveis, tais eventos são omitidos nas respectivas transições de estado de  $S$ . Para habilitar, os eventos devem estar explícitos nas transições de estado de  $S$ . Na representação gráfica dos geradores, os eventos controláveis são aqueles que apresentam um traço no arco de transição.

Diz-se que  $S/G$  é o gerador que representa  $S$  controlando  $G$ . Um supervisor  $S$  é não bloqueante se  $\overline{L_m(S/G)} = L(S/G)$ , sendo  $\overline{L}$  o conjunto de todas as cadeias de  $\Sigma^*$  que são prefixos de cadeias de  $L$  (prefixo-fechamento).

No desenvolvimento do supervisor, consideram-se as seqüências de eventos possíveis de ocorrer no sistema real – malha aberta – e, obtém-se uma especificação  $K$ , a partir da composição síncrona das especificações genéricas  $E$  e da planta  $G$ . Deseja-se obter um elemento controlador que não apresente seqüências indesejáveis que se oponham à lógica de controle que se queira alcançar ou que levariam a uma situação de bloqueio.

A condição necessária e suficiente para a existência de um supervisor  $S$  não bloqueante que satisfaça uma dada especificação  $K = L_m(S/G) \subseteq L_m(G)$  é a controlabilidade de  $K$ . Diz-se que  $K$  é controlável e.r.a  $G$  se  $\bar{K} \Sigma_u \cap L(G) \subseteq \bar{K}$ , ou seja, a ocorrência de um evento não controlável em  $G$  não gera uma seqüência de eventos que desrespeite a especificação. Seja  $C(K, G)$  o conjunto de todas as sublinguagens de  $K$  controláveis e.r.a  $G$ . Existe um elemento, chamado  $SupC(K, G)$ , que contém a máxima sublinguagem de  $K$  que é controlável e.r.a  $G$  (Wonham, 2008).

O não bloqueio é tratado durante o cálculo do supervisor, que leva em conta os estados marcados, os quais indicam a conclusão de tarefas. Uma tarefa completa de  $G$  sob a supervisão de  $S$  acontece quando ambos os geradores alcançam um estado marcado.

Diz-se que o controle é monolítico quando há um único supervisor atuando sobre a planta que atenda todas as especificações. Queiroz e Cury (2000) propõe a abordagem modular local, a qual é uma extensão da TCS, que oferece um controle através de vários supervisores e plantas locais. Cada supervisor é responsável por uma especificação comportamental e juntos cooperam para o correto monitoramento e atuação no sistema.

Na abordagem modular local, a especificação restringe as ações da planta local, a qual é composta por todos os geradores dos subsistemas que compartilham ao menos um evento com esta especificação. Suponha que um sistema seja composto pelos subsistemas  $G_k = (\Sigma_k, Q_k, \delta_k, q_{0k}, Q_{mk})$ ,  $k \in K$ . As especificações genéricas  $E_x$ ,  $x \in X$ , representam o comportamento desejável para este sistema e seus alfabetos são definidos por  $\Sigma_x \subseteq \Sigma$ . A planta local  $G_{locx} = (\Sigma_{locx}, Q_{locx}, \delta_{locx}, q_{0locx}, Q_{mlocx})$  referente à especificação  $E_x$  é expressa por  $G_{locx} = \parallel_{k \in I_{locx}} G_x$  para  $I_{locx} =$

$$\{k \in K \mid \Sigma_k \cap \Sigma_x \neq \emptyset\}.$$

O exemplo apresentado na figura 2.3 ilustra como são obtidas as plantas locais. Sejam  $G_0, G_1, G_2, G_3$  e  $G_4$  os geradores que representam os subsistemas da planta e  $E_0, E_1$  e  $E_2$  as especificações genéricas de um determinado sistema. Cada especificação genérica está associada a um conjunto  $I_{locx}$  formado pelos geradores que apresentam ao menos um evento comum à esta especificação. Neste exemplo,  $E_0$  está associado ao conjunto  $I_{locE0} = \{G_0, G_1\}$ ,  $E_1$  ao conjunto  $I_{locE1} = \{G_1, G_2, G_3\}$  e  $E_2$  ao conjunto  $I_{locE2} = \{G_4\}$ . A composição síncrona dos elementos de cada conjunto fornece a planta local referente a cada especificação genérica do sistema.

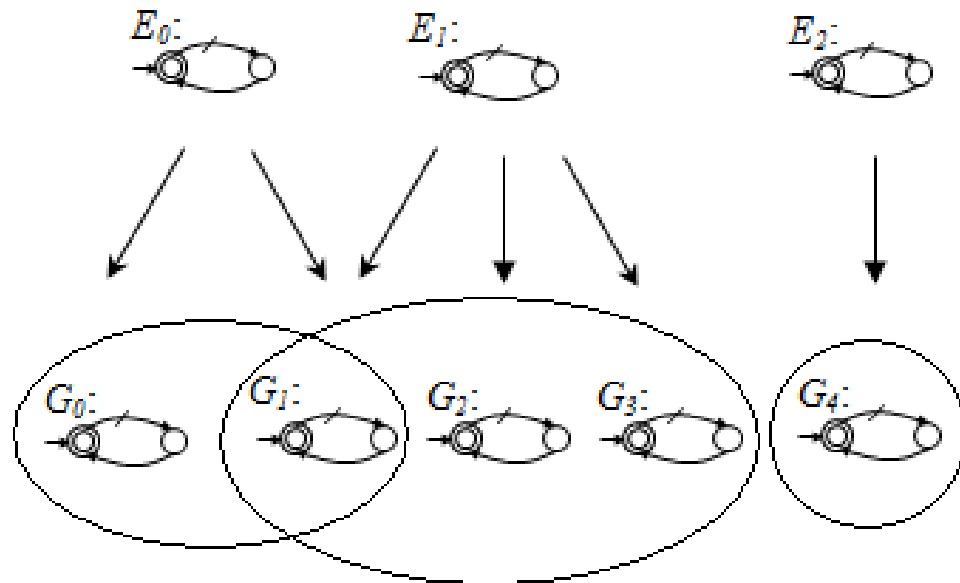


Figura 2.3. Exemplificação de plantas locais

Obtêm-se os supervisores modulares locais  $S_{locx}$  calculando-se  $SupC(K_{locx}, G_{locx})$ , sendo  $K_{locx}$  a especificação local, a qual é definida por  $K_{locx} = G_{locx} \parallel E_x$ .

Para garantir que os supervisores modulares locais atuem no sistema de modo equivalente à abordagem monolítica, se faz necessária a verificação da modularidade local dos supervisores. Tal propriedade é válida se o gerador resultante da composição síncrona de todos os supervisores modulares locais do sistema é *trim*. Caso contrário, há conflito entre os supervisores e, como consequência, não é mais garantida a propriedade de não bloqueio dos supervisores modulares locais. Para solucionar o conflito recorre-se a diferentes alternativas. Uma delas é descartar os supervisores modulares locais e recorrer ao controle monolítico. Pode-se também utilizar o supervisor monolítico em conjunto com os supervisores modulares locais, de modo que o monolítico atue como o coordenador dos supervisores conflitantes. Tal método se torna útil quando se deseja conservar a arquitetura modular de controle (Queiroz, 2004). Pena *et al.*(2009) e Flordal e Malik (2006) também procuram soluções para tornar este problema abordável.

A metodologia para a síntese dos supervisores segundo a TCS segue três etapas: modelagem da planta a ser controlada, modelagem das especificações desejadas para a planta e síntese de uma lógica de controle ótima. Nas próximas seções deste capítulo é apresentada a aplicação da TCS para controlar uma célula flexível de manufatura.

### 2.3 A c3lula flex3vel de manufatura

A CFM 3 uma bancada experimental desenvolvida em parceria entre o Instituto Federal Santa Catarina (IFSC) e o Departamento de Automa33o e Sistemas da Universidade Federal de Santa Catarina. A CFM, figura 2.4, 3 composta por quatro esta33es operacionais que s3o conectadas atrav3s de uma mesa rotativa, que opera sempre no sentido hor3rio, e um manipulador rob3tico. Estas esta33es realizam a fura33o, a soldagem, o teste de qualidade e o retrabalho de pe3as. O objetivo principal da CFM 3 furar e soldar pe3as que chegam ao *buffer* de entrada. Naturalmente, como em uma ind3stria real, as pe3as algumas vezes n3o s3o fabricadas corretamente no ciclo normal de produ33o e necessitam ser reparadas. Desta forma, a esta33o de retrabalho da CFM funciona como entrada de pe3as manualmente restauradas. O sistema n3o apresenta sensores para indicar a chegada de pe3as nas esta33es de fura33o, soldagem e teste. Assim, a presen3a de pe3as sobre a mesa deve ser deduzida a partir do hist3rico de eventos.

H3 tr3s classes de armazenamento poss3veis para as pe3as: aprovadas, reprovadas e danificadas. Quando uma pe3a falha no teste de qualidade, ela 3 reprovada se vier da esta33o de retrabalho ou vai para o dep3sito de pe3as danificadas se for oriunda do *buffer* de entrada. Se a pe3a foi manufaturada corretamente, ela 3 posta classificada como aprovada, n3o importando sua origem.

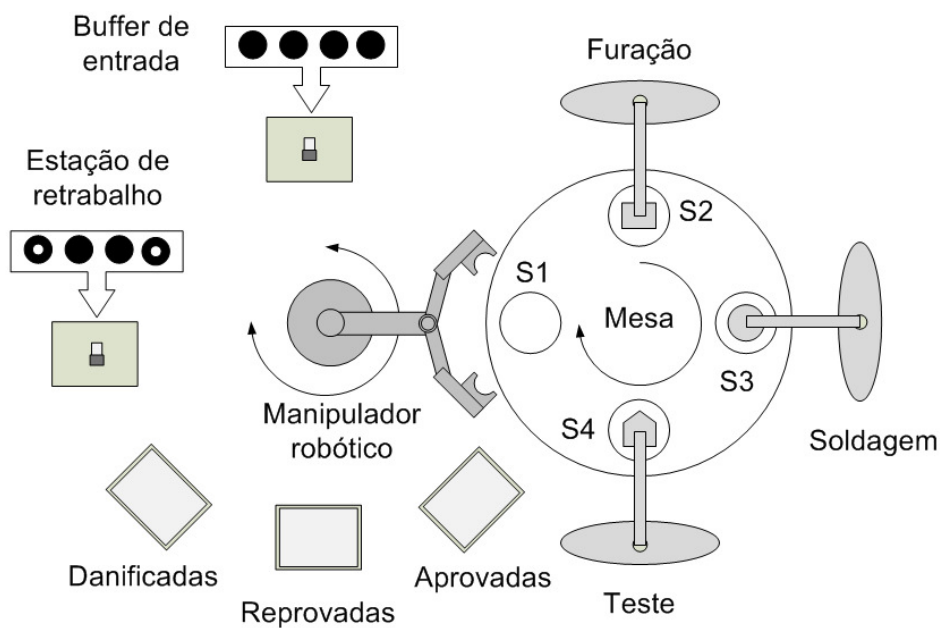


Figura 2.4. C3lula Flex3vel de Manufatura



Como mencionado, as peças podem vir tanto do buffer de entrada como da estação de retrabalho. Quando uma peça bruta chega ao buffer de entrada o manipulador robótico a coloca sobre a posição S1 da mesa. Então, a mesa giratória conduz a peça bruta pelas estações de furação, soldagem e teste de qualidade, e finalmente retorna para a posição S1 para ser retirada. Finalmente, a garra coloca a peça no depósito de peças aprovadas ou danificadas conforme o resultado do teste.

Sempre que uma peça chega à estação de retrabalho, o manipulador robótico a coloca sobre a mesa giratória, a qual faz a peça alcançar a estação de teste de qualidade e posteriormente a posição S1 sem que sejam efetuados os processos de furação e soldagem. Alcançando a posição S1, o manipulador robótico retira a peça da mesa e a coloca no depósito de aprovadas ou reprovadas de acordo com o teste.

É possível implementar uma ampla variedade de leis de controle para a CFM dependendo do comportamento desejado pelo projetista. Poderia ser especificado, por exemplo, uma seqüência de manufatura que permitisse o uso de um único *slot* da mesa giratória. Desde que se pretende alcançar a máxima eficiência e flexibilidade na produção, respeitando as restrições de segurança do sistema, deseja-se que a CFM possa fabricar peças utilizando até o limite máximo dos *slots* da mesa simultaneamente. Para isso, devem-se evitar alguns problemas que poderiam acontecer durante a fabricação de várias peças ao mesmo tempo, como:

- Girar a mesa se não houver pelo menos uma peça sobre ela;
- Girar a mesa enquanto qualquer das estações estiver realizando uma tarefa;
- Girar a mesa caso existam peças esperando para serem furadas, soldadas, testadas ou removidas pelo manipulador robótico;
- Efetuar as operações de furação, soldagem, teste e manipulação robótica se não houver peça esperando pela realização das tarefas citadas no *slot* correspondente;
- Executar a mesma operação por duas vezes consecutivas em uma peça;
- Furar e soldar peças que chegam da estação de retrabalho;
- Sobrepor peças na posição S1;
- Acionar o manipulador robótico para pegar peças nas entradas do sistema sem haver peça nas mesmas.

Respeitando os critérios descritos, pretende-se sintetizar um controlador não bloqueante e minimamente restritivo para a CFM. Assim,

para resolver este problema de controle, será utilizada a TCS (Ramadge e Wonham, 1989).

## 2.4 Modelagem dos subsistemas da CFM

De acordo com a metodologia da TCS, a modelagem da planta inicia-se com a identificação dos subsistemas que a compõem. Cada um deles apresenta um comportamento próprio que quando atuando em conjunto, sob supervisão de um elemento de controle, atuam de forma coordenada. No caso da CFM, são subsistemas da planta: a mesa, o manipulador robótico, o *buffer* de entrada e as estações de retrabalho, furação, soldagem e teste. Estes subsistemas são modelados separadamente por geradores. Como os problemas de controle da CFM referidos na seção 2.3 relatam situações que poderiam ocorrer durante a interação entre os subsistemas, os geradores modelados levam em conta basicamente o início e fim de operações, o que não acarreta perdas no desempenho do controle. Na tabela 2.1 são apresentados os eventos utilizados na modelagem dos geradores dos subsistemas e o significado físico dos mesmos.

O *buffer* de entrada e a estação de retrabalho possuem sensores, cujos modelos –  $G_1$  e  $G_2$  – são vistos na figura 2.5. Os sensores sinalizam a chegada de peças através dos eventos *sen\_ib* e *sen\_rw* respectivamente. Ambos os eventos são modelados como não controláveis e podem ocorrer a qualquer instante no sistema.

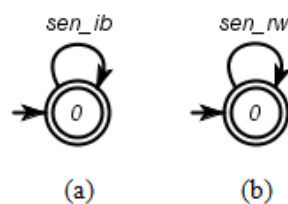
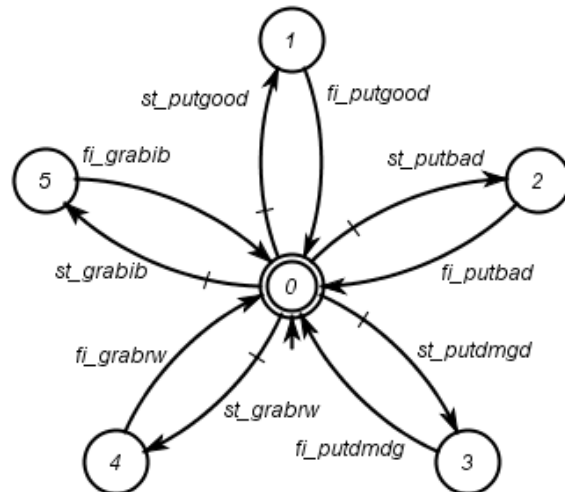


Figura 2.5. Geradores (a)  $G_1$  e (b)  $G_2$

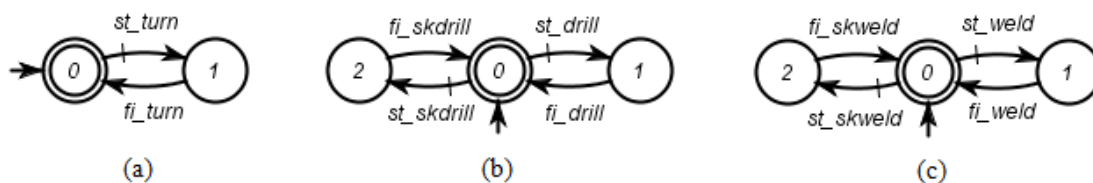
O manipulador robótico, representado pela gerador  $G_3$  na figura 2.6, executa cinco ações diferentes. Ele pode pegar peças do *buffer* de entrada ou da estação de retrabalho, iniciados por *st\_grabib* e *st\_grabrw* respectivamente, e colocá-las no *slot* S1 da mesa. Ele também retira as peças que alcançam o *slot* S1 e as coloca nos três compartimentos de armazenamento de peças, através dos comandos *st\_putgood*, *st\_putbad* e *st\_putdmgd*, que iniciam a operação de estocagem das peças aprovadas, reprovadas e danificadas respectivamente.

Tabela 2.1. Eventos da Célula Flexível de Manufatura

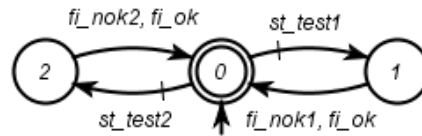
Equipamento	Evento	Descrição
Mesa giratória	<i>st_turn</i> (controlável)	Gira a mesa em 90 graus no sentido horário.
	<i>fi_turn</i> (não controlável)	Sinaliza o fim do giro da mesa.
Furadeira	<i>st_drill</i> (controlável)	Inicia a furação.
	<i>fi_drill</i> (não controlável)	Sinaliza o fim da furação.
	<i>st_skdrill</i> (controlável)	Não realiza a furação.
	<i>fi_skdrill</i> (não controlável)	Sinaliza a não realização da furação.
Soldadeira	<i>st_weld</i> (controlável)	Inicia a soldagem.
	<i>fi_weld</i> (não controlável)	Sinaliza o fim da soldagem.
	<i>st_skweld</i> (controlável)	Não realiza a soldagem.
	<i>fi_skweld</i> (não controlável)	Sinaliza a não realização da soldagem.
Teste	<i>st_test1</i> (controlável)	Inicia o teste das peças brutas.
	<i>st_test2</i> (controlável)	Inicia o teste das peças retrabalhadas.
	<i>fi_ok</i> (não controlável)	Sinaliza a aprovação no teste.
	<i>fi_nok1</i> (não controlável)	Sinaliza a primeira rejeição no teste.
	<i>fi_nok2</i> (não controlável)	Sinaliza a segunda rejeição no teste.
Manipulador Robótico	<i>st_grabib</i> (controlável)	Pega peça do <i>buffer</i> de entrada e coloca na posição S1.
	<i>fi_grabib</i> (não controlável)	Sinaliza o fim desta operação.
	<i>st_grabr</i> (controlável)	Pega peça da estação de retrabalho e coloca na posição S1.
	<i>fi_grabr</i> (não controlável)	Sinaliza o fim desta operação.
	<i>st_putgood</i> (controlável)	Retira peça da posição S1 e coloca no compartimento de peças aprovadas.
	<i>fi_putgood</i> (não controlável)	Sinaliza o fim desta operação.
	<i>st_putbad</i> (controlável)	Retira peça da posição S1 e coloca no compartimento de peças reprovadas.
	<i>fi_putbad</i> (não controlável)	Sinaliza o fim desta operação.
	<i>st_putdmg</i> (controlável)	Retira peça da posição S1 e coloca no compartimento de peças danificadas.
	<i>fi_putdmg</i> (não controlável)	Sinaliza o fim desta operação.
Sensor do <i>buffer</i> de entrada	<i>sen_ib</i> (não controlável)	Sinaliza a chegada de peça no <i>buffer</i> de entrada.
Sensor da estação de retrabalho	<i>sen_rw</i> (não controlável)	Sinaliza a chegada de peça através da estação de retrabalho.

Figura 2.6. Gerador  $G_3$ 

O gerador da mesa –  $G_4$  –, ilustrada pela figura 2.7 (a), inicia a rotação a partir do comando  $st\_turn$  e ao final da operação envia o sinal  $fi\_turn$ . As estações de furação –  $G_5$  – e soldagem –  $G_6$  – iniciam suas respectivas operações sobre as peças que chegam do buffer de entrada através dos comandos  $st\_drill$  e  $st\_weld$  e enviam os sinais  $fi\_drill$  e  $fi\_weld$  após a conclusão das mesmas. De acordo com as restrições apresentadas na seção 2.3, as peças originárias da estação de retrabalho não devem ser furadas e soldadas. Para isso, as plantas  $G_5$  e  $G_6$ , ilustradas na figura 2.7 (b) e (c) respectivamente, são modeladas com os eventos  $st\_skdrill$  e  $st\_skweld$ , que representam a não realização das operações de furação e soldagem. A introdução destes eventos fictícios facilita a modelagem das especificações apresentadas na seção 2.5.

Figura 2.7. Geradores (a)  $G_4$ , (b)  $G_5$  e (c)  $G_6$ 

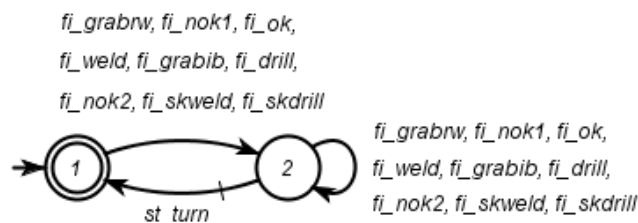
A estação de teste deve alertar o sistema sobre quantas vezes uma peça foi rejeitada. Para isto, o gerador da estação de teste –  $G_7$  –, visto na figura 2.8, é modelado para executar, logicamente, dois tipos de teste. O primeiro teste é aplicado às peças originárias do *buffer* de entrada e responde com  $fi\_ok$  caso a peça tenha sido aprovada ou  $fi\_nok1$  se rejeitada pela primeira vez. O segundo teste verifica as peças que chegam pela estação de retrabalho e a resposta pode indicar aprovação, com  $fi\_ok$ , ou a segunda rejeição, através de  $fi\_nok2$ .

Figura 2.8. Gerador  $G_7$ 

## 2.5 Modelagem das especificações

A composição síncrona dos geradores dos subsistemas da planta representa o comportamento em malha aberta da mesma. Em malha aberta há seqüências de eventos que levariam a situações indesejáveis para o sistema. Um exemplo desta situação na CFM seria a mesa girar enquanto uma peça estivesse sendo furada. O papel das especificações é expressar um conjunto de restrições que evite a ocorrência destas seqüências indesejadas. Em geral, as especificações são modeladas por um conjunto de geradores que determinam como a operação coordenada dos subsistemas deve ocorrer. No caso da CFM, as especificações focam na resolução dos problemas de interação entre os subsistemas mencionados na seção 2.3.

A especificação  $E_1$ , vista na figura 2.9, preocupa-se em não permitir que a mesa gire sem a existência de peças sobre seus *slots*. No estado 1 não há peças sobre a mesa, sendo proibido que a mesa gire – evento *st\_turn*. O evento *st\_turn* somente é habilitado no estado 2, quando é certo que existe pelo menos uma peça sobre a mesa – após ocorrer pelo menos um dos eventos *fi\_grabrw*, *fi\_grabib*, *fi\_drill*, *fi\_skdrill*, *fi\_weld*, *fi\_skweld*, *fi\_ok*, *fi\_nok1* ou *fi\_nok2*. O auto-laço em  $E_1$ , permite que os eventos listados anteriormente possam ocorrer mais de uma vez, possibilitando a execução concorrente destas tarefas.

Figura 2.9. Gerador  $E_1$ 

Durante a execução do sistema, deve haver exclusão mútua entre a mesa e o manipulador robótico, a estação de furação, a estação de soldagem e a estação de teste. Para tratar a exclusão mútua, foram modelados os geradores  $E_{2a}$ ,  $E_{2b}$ ,  $E_{2c}$  e  $E_{2d}$ , que são visualizados na figura 2.10. Através deles exprime-se que uma vez iniciada a rotação da mesa

–  $st\_turn$ , nenhuma operação das estações ou do manipulador robótico deve ser realizada até que o sinal  $fi\_turn$  seja sinalizado e vice-versa.

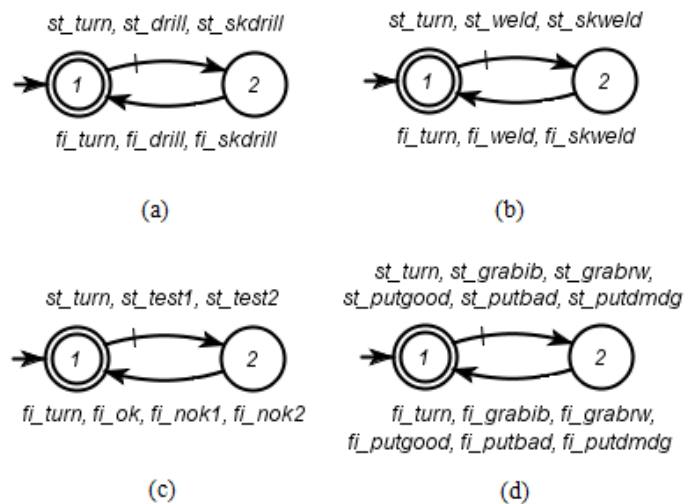


Figura 2.10. Geradores (a)  $E_{2a}$ , (b)  $E_{2b}$ , (c)  $E_{2c}$  e (d)  $E_{2d}$

A coordenação do manipulador robótico e das estações de furação, soldagem e teste em relação ao fluxo de peças é subdividida em quatro especificações, de modo que cada uma delas foca na interação entre dois subsistemas vizinhos. A especificação  $E_{3a}$  responde pelo manipulador robótico e a estação de furação,  $E_{3b}$  pelas estações de furação e soldagem,  $E_{3c}$  pelas estações de soldagem e teste e  $E_{3d}$  pela estação de teste e o manipulador robótico. Um modelo genérico para os geradores  $E_{3a}$ ,  $E_{3b}$  e  $E_{3c}$  é mostrado na figura 2.11. Os estados destes geradores são definidos por um par de números que indicam se nos *slots* dos subsistemas envolvidos na coordenação existem peças esperando pela execução de uma operação. Ressalta-se que o *slot* associado ao manipulador robótico é o S1. Os números 0, 1 e 2 indicam respectivamente que não há peça no *slot* esperando pela execução de uma tarefa, que há uma peça vinda do *buffer* de entrada esperando pela execução de uma tarefa e que há uma peça vinda da estação de retrabalho esperando pela execução de uma tarefa. Os índices numéricos dos estados representam respectivamente o primeiro e o segundo subsistema envolvido na coordenação.

Os eventos  $fi\_<ação\_ib>$  correspondem aos sinais de fim de operação do subsistema anterior sobre uma peça oriunda do *buffer* de entrada. Já o evento  $st\_<prox\_ação>$  representam o início das operações sobre as peças vindas do *buffer* de entrada pelo subsistema seguinte ao que sinalizou o evento  $fi\_<ação\_ib>$ . Análise análoga pode ser aplicada aos eventos  $fi\_<ação\_rw>$  e  $st\_<prox\_ação\_rw>$  em relação às peças originárias da estação de retrabalho. Suponha que o gerador da figura 2.11 represente a especificação  $E_{3a}$ , que coordena o manipulador robóti-

co e a estação de furação. Os eventos  $fi_{\langle ação_{ib} \rangle}$ ,  $st_{\langle prox_{ação_{ib}} \rangle}$ ,  $fi_{\langle ação_{rw} \rangle}$ ,  $st_{\langle prox_{ação_{rw}} \rangle}$  equivalem respectivamente aos eventos  $fi_{grabib}$ ,  $st_{drill}$ ,  $fi_{grabrw}$ ,  $st_{skdrill}$ .

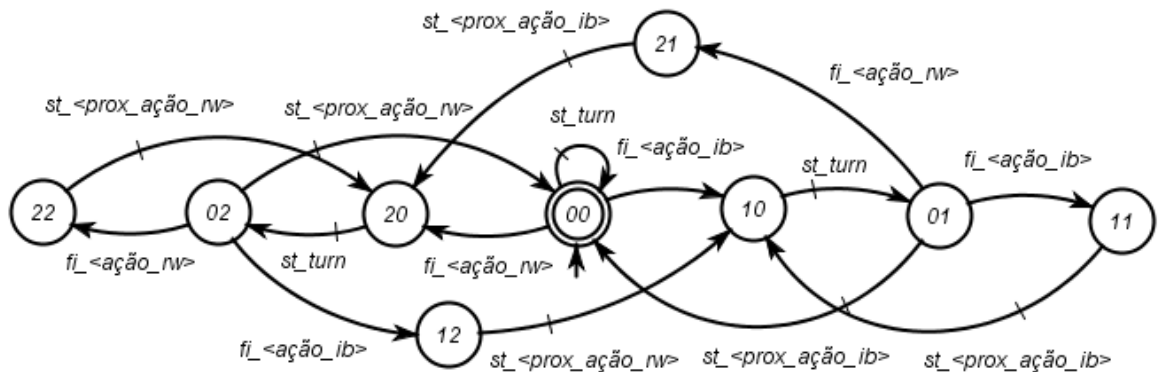


Figura 2.11. Gerador genérico para  $E_{3a}$ ,  $E_{3b}$ ,  $E_{3c}$

Alguns eventos devem ser proibidos de ocorrer em determinados estados do gerador. Um exemplo é o evento  $st_{turn}$ , que não deve ser executado nos estados cujo segundo índice numérico é diferente de zero, pois são estados que indicam a existência de peças sobre as quais não foram realizadas as operações esperadas. A figura 2.12 mostra uma seqüência de eventos a partir da qual é possível acompanhar a evolução da especificação  $E_{3a}$ . Suponha que a mesa esteja inicialmente vazia. Quando o manipulador robótico deposita uma peça vinda do *buffer* de entrada sobre a mesa, figura 2.12 (a),  $E_{3a}$  evolui para o estado 10 e o evento  $fi_{grabib}$  é desabilitado para evitar *overflow* de peças na posição S1. Neste estado somente é possível ocorrer o evento  $st_{turn}$ . Já no estado 01, fisicamente representado pela figura 2.12 (b), existe a possibilidade de iniciar o processo de furação da peça – evento  $st_{drill}$  – ou chegar peça do *buffer* de entrada – evento  $fi_{grabib}$  – ou da estação de retrabalho – evento  $fi_{grabrw}$ . Suponha que chegue uma peça em S1, figura 2.12 (c). Qualquer que seja a peça posta em S1 implica a desabili-

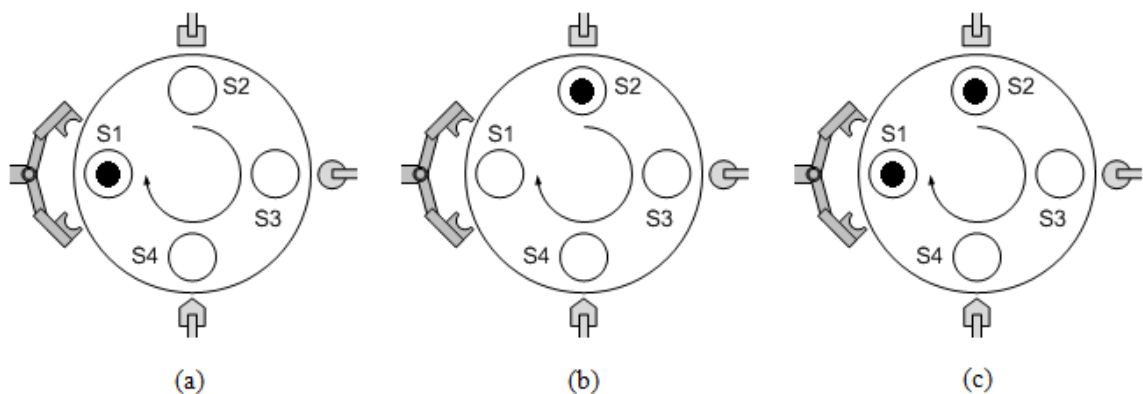


Figura 2.12. Seqüência de operações relacionada à especificação  $E_{3a}$

tação do evento  $st\_turn$  porque a peça no  $slot$  S2 ainda não sofreu a operação de furação.

A especificação  $E_{3d}$ , vista na figura 2.13, lida com a coordenação da estação de teste e do manipulador robótico. Ela segue o mesmo raciocínio adotado para modelar as especificações  $E_{3a}$ ,  $E_{3b}$  e  $E_{3c}$ , no entanto apresenta mais estados do que estas. Isto acontece porque a estação de teste oferece três possibilidades de respostas –  $fi\_ok$ ,  $fi\_nok1$  e  $fi\_nok2$  – ao passo que os modelos anteriormente apresentados sinalizam duas possibilidades –  $fi\_<ação\_ib>$  e  $fi\_<ação\_rw>$ . Em  $E_{3d}$ , os índices dos estados passam a representar estas variações de respostas, significando 1, 2 e 3 respectivamente a existência de peça aprovada, rejeitada pela primeira vez e rejeitada pela segunda vez, que aguardam pela execução da operação do manipulador robótico.

Juntas, estas quatro especificações asseguram que não haverá *overflow* de peças no  $slot$  S1; a mesa não girará antes de a peça ser furada, soldada ou testada; peças remanufaturadas não serão soldadas e furadas; não ocorrerá furação, soldagem ou manipulação de peças se não há peça na estação correspondente; não ocorrerá dupla furação, solda-

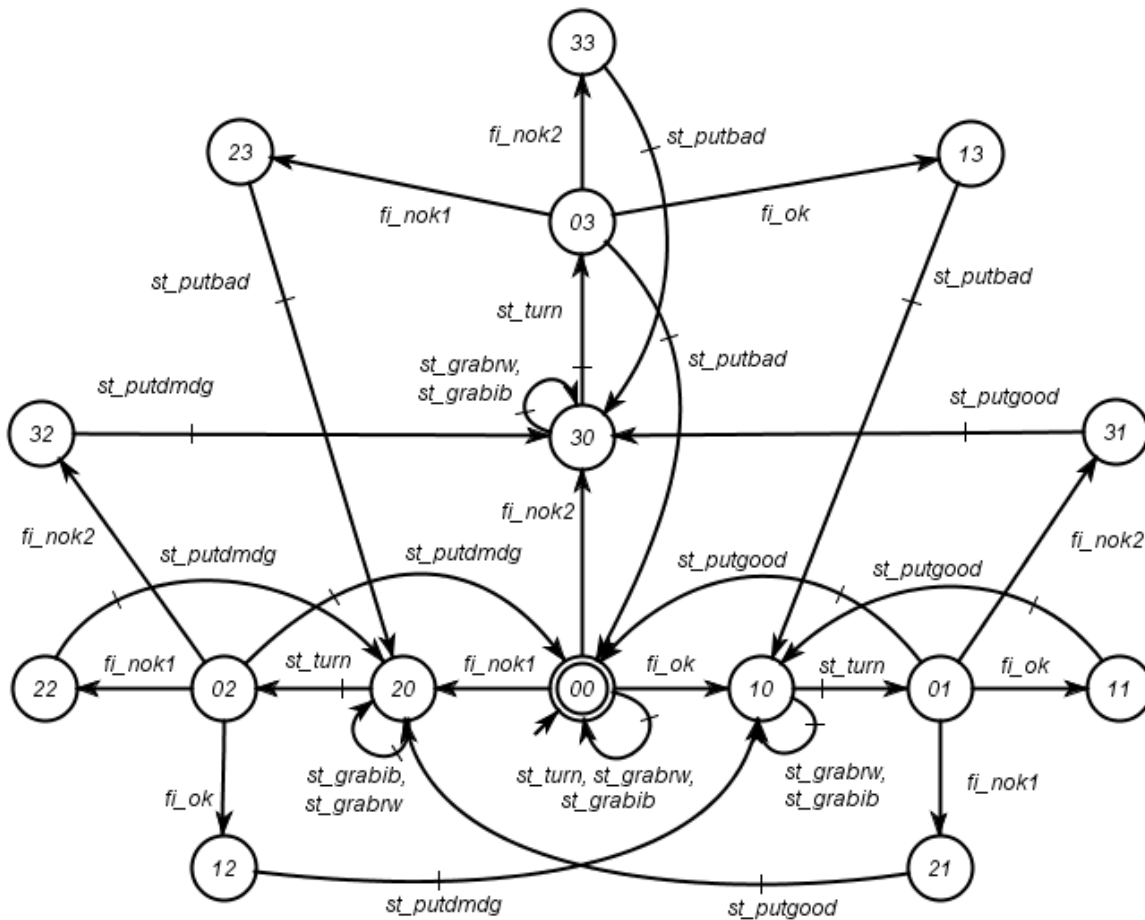


Figura 2.13. Gerador E3d



gem ou teste seqüencial na mesma peça.

Não se pode prever quando as peças chegarão às entradas do sistema, todavia com as especificações  $E_{4a}$  e  $E_{4b}$ , ilustradas pela figura 2.14, se estabelece que o manipulador robótico somente deva pegá-las após os sensores de entrada serem ativados. Os sensores podem ser disparados a qualquer momento do processo de manufatura, desta forma estes eventos aparecem como um auto-laço no estado 2 de  $E_{4a}$  e  $E_{4b}$ .

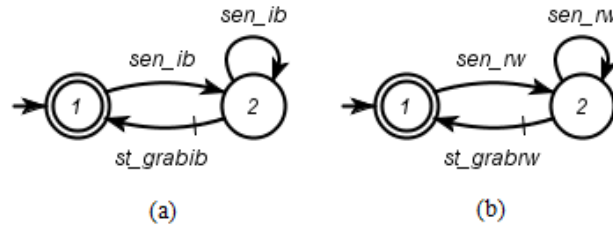


Figura 2.14. Geradores (a)  $E_{4a}$  e (b)  $E_{4b}$

### 2.5.1 Síntese dos supervisores modulares locais da CFM

Conforme foi apresentado na seção 2.2, na abordagem modular local são exploradas as modularidades das especificações e da planta. As plantas locais são obtidas a partir da composição síncrona dos geradores dos subsistemas afetados pelas restrições locais, ou seja, aqueles que compartilham ao menos um evento a especificação.

Tomando como exemplo a construção da planta local  $G_{loc3d}$ , referente à especificação  $E_{3d}$ , tem-se que o alfabeto desta especificação é definido por  $\Sigma_{E_{3d}} = \{fi\_ok, fi\_nok1, fi\_nok2, st\_turn, st\_putgood, st\_putbad, st\_putdmgd, st\_grabib, st\_grabrw\}$ . Comparando este alfabeto com os alfabetos dos geradores dos subsistemas da planta, o qual pode ser obtido a partir da tabela 2.1, nota-se que os geradores  $G_3$ ,  $G_4$  e  $G_7$  apresentam eventos comuns à especificação  $E_{3d}$ . Desta forma, a planta local é a composição síncrona dos geradores  $G_3$ ,  $G_4$  e  $G_7$ . As plantas locais para a CFM são vistas na tabela 2.2.

Conforme visto, a especificação local  $K_{locx}$  é definida por  $K_{locx} = G_{locx} \parallel E_x$ . Aplicando ao exemplo da CFM tem-se que  $x \in X = \{1, 2a, 2b, 2c, 2d, 3a, 3b, 3c, 3d, 4a, 4b\}$ . Computando-se os supervisores modulares locais  $S_{locx}$  para a CFM, sendo  $S_{locx} = SupC(K_{locx}, G_{locx})$ , foram obtidos 11 supervisores  $S_{locx}$ , para  $x \in X$ . A tabela 2.3 resume o número de estados dos geradores envolvidos na síntese dos supervisores modulares locais.  $S_{redx}$  representa os supervisores reduzidos equivalentes aos respectivos  $S_{locx}$ . A redução dos supervisores é discutida na próxima seção.

Tabela 2.2. Plantas locais para a CFM

$G_{locx}$	$\parallel G_x$
$G_{loc1}$	$G_4 \parallel G_5 \parallel G_6 \parallel G_7 \parallel G_3$
$G_{loc2a}$	$G_4 \parallel G_5$
$G_{loc2b}$	$G_4 \parallel G_6$
$G_{loc2c}$	$G_4 \parallel G_7$
$G_{loc2d}$	$G_4 \parallel G_3$
$G_{loc3a}$	$G_4 \parallel G_5 \parallel G_3$
$G_{loc3b}$	$G_4 \parallel G_5 \parallel G_6$
$G_{loc3c}$	$G_4 \parallel G_6 \parallel G_7$
$G_{loc3d}$	$G_4 \parallel G_7 \parallel G_3$
$G_{loc4a}$	$G_1 \parallel G_3$
$G_{loc4b}$	$G_2 \parallel G_3$

Tabela 2.3. Número de estados dos geradores da síntese dos supervisores

$X$	$E_x$	$G_{locx}$	$K_{locx}$	$S_{locx}$	$S_{redx}$
1	2	324	648	648	2
2a, 2b, 2c	2	6	4	4	2
2d	2	12	7	7	2
3a	9	36	324	252	9
3b, 3c	9	18	162	90	9
3d	16	36	576	288	16
4a, 4b	2	6	12	12	2

Os supervisores foram verificados como não conflitantes após certificar que o gerador  $S = \parallel S_{locx}$ , para  $x \in X$ , é *trim*. O gerador  $S$  apresenta 19180 estados e, por ser *trim*, é a solução monolítica para o problema da CFM. Com o teste de não conflito, pode-se afirmar que os supervisores modulares locais atuando em conjunto comportam-se de modo equivalente à solução monolítica. A computação do cálculo de cada supervisor modular local foi realizada na ferramenta TCT (Feng e Wonham, 2006) e levou menos de 1 segundo. Elas foram processadas num computador pessoal AMD Turion™ 64 X2 Dual-Core 1.9 GHz, 2048 MB RAM, Windows Vista® Home Premium 32-bit version.

## 2.6 Redução e implementação de supervisores

Nota-se que alguns dos supervisores da CFM apresentam muitos estados. Pode-se obter a partir dos supervisores locais  $S_{locx}$ , supervisores reduzidos através de algoritmos de redução de supervisores como, por exemplo, o algoritmo de Su e Wonham (2004). A redução dos supervisores é de extrema importância para a implementação pois proporciona redução do uso de memória computacional e torna compreensível a interpretação dos supervisores. A redução atingiu supervisores com número de estados idênticos às especificações modeladas. A redução dos supervisores modulares locais também foi feita através da ferramenta TCT. O tempo de redução dos supervisores modulares locais da CFM variou entre 1 e 2 segundos, com exceção do pior caso,  $S_{loc1}$ , que levou 13 horas. A redução foi realizada no mesmo computador pessoal citado na seção 2.5.1.

Os supervisores modulares locais sintetizados para a CFM foram implementados no controlador lógico programável (CLP) Altus PO 3147 na linguagem *structured text* (IEC, 1993), em suas versões reduzidas, auxiliados por uma ferramenta de geração automática de código (Klinge, 2007). A implementação foi feita através de uma estrutura genérica de implementação (Queiroz e Cury, 2002), que será explicada no capítulo 5. A implementação foi bem sucedida e a CFM operou de modo flexível e sem a ocorrência de *deadlocks*. Detalhes da implementação podem ser vistos em (Silva e Queiroz, 2009).

## 2.7 Conclusão do capítulo

Com a abordagem modular local da TCS foi possível sintetizar supervisores ótimos – minimamente restritivos e não bloqueantes – para coordenar a CFM de acordo com os requisitos exigidos. A modelagem das especificações através de geradores permite tratar os problemas de forma modular, tornando simples a alteração ou adição de novas leis de controle. Além de ser versátil, esta técnica de modelagem se estende a diversas classes de problemas. No capítulo 4, a abordagem modular local é aplicada ao problema de coordenação de AGVs.



## Capítulo 3

# Sistemas de veículos auto-guiados

Os AGVs são veículos que se deslocam de modo autônomo, gerenciados por um sistema de controle, através de um mapa de percurso pré-definido. Numa planta industrial os AGVs lidam com o transporte de matéria-prima, de *pallets*, de produtos ainda sendo manufaturados e de produtos finais.

Neste capítulo é apresentada a estrutura de sistemas de AGV e são descritas as principais características dos seus componentes, além de serem levantados alguns dos problemas de controle relacionados ao mesmo. Este capítulo é a base para formulação do problema de controle de AGVs a ser tratado no capítulo 4.

### 3.1 Estrutura de sistema de AGV

Um sistema de AGV é um sistema de manejo de materiais que utiliza veículos autônomos que se deslocam através de trajetos pré-definidos (Groover, 2007). Ele é basicamente estruturado através dos seguintes componentes: veículos, guias de deslocamento, unidade de controle e interface computacional (Cubberly e Bakerjian, 1989).

O sistema executa suas ações sob monitoramento do sistema de controle, o qual é responsável por coordenar o deslocamento dos veículos e a interação destes com os demais componentes do ambiente – pontos de carregamento e descarregamento, estações de recarga de bateria, etc.

A malha do sistema é integrada através de um mecanismo de comunicação entre os AGVs e o sistema de controle, o que possibilita uma monitoração e supervisão em tempo real do sistema. A supervisão é auxiliada por uma interface gráfica que traduz as informações relevantes que ocorrem durante a evolução do sistema (Mulcahy, 1999).

### 3.1.1 Veículos

Os AGVs são utilizados como solução para manejo de materiais em diversos campos. Para cada finalidade há uma variação nos modelos de AGVs a serem empregados. Algumas aplicações exigem soluções tão únicas que são necessários AGVs personalizados para a realização destas atividades. Estes veículos são construídos considerando o tipo e as dimensões do material a ser manuseado, gerando projetos exclusivos de AGVs. De modo geral, os AGVs são classificados em quatro categorias: reboque (*cart towing*), carga de unidade (*unit load*), carga de *pallet* (*pallet load*) e cargas leves (*miniload* ou *lightload*) (Mulcahy, 1999).

Os rebocadores, figura 3.1 (a), são utilizados quando há grande fluxo de materiais entre pontos distantes do ambiente industrial. Isto é característico nos processos de estocagem, coleta de pedidos e transferência de materiais para uma próxima etapa da manufatura. Este tipo de transporte envolve múltiplas paradas em diferentes locais de entrega e coleta de materiais no decorrer de um ciclo completo do percurso. Para isto, conjuntos de reboques são acoplados ao AGV e o mesmo é despachado para o ponto de coleta ou entrega. Normalmente é necessário um operador para desacoplar ou acoplar os reboques no AGV, além de programar o AGV para o próximo ponto de parada do veículo. Os reboques podem ser carregados ou descarregados por empilhadeiras do tipo auto-guiadas ou não. Existem versões de reboques que possuem mecanismos de descarregamento automático e desta forma é excluído o auxílio das empilhadeiras no descarregamento.

Os veículos de carga de unidade, figura 3.1 (b), transportam as cargas sobre uma plataforma equipada com mecanismos que permitem o carregamento e descarregamento automático dos materiais como, por exemplo, sistemas de rolagem, correias e plataformas de elevação. Estes veículos não são capazes de coletar cargas do chão. Nesta categoria de veículos, incluem-se também os AGVs de linha de montagem, figura 3.1 (c), que percorrem diferentes estações de trabalho nas quais são acrescentados componentes a unidade transportada e ao alcançar a última estação é obtido um produto final (Groover, 2007).

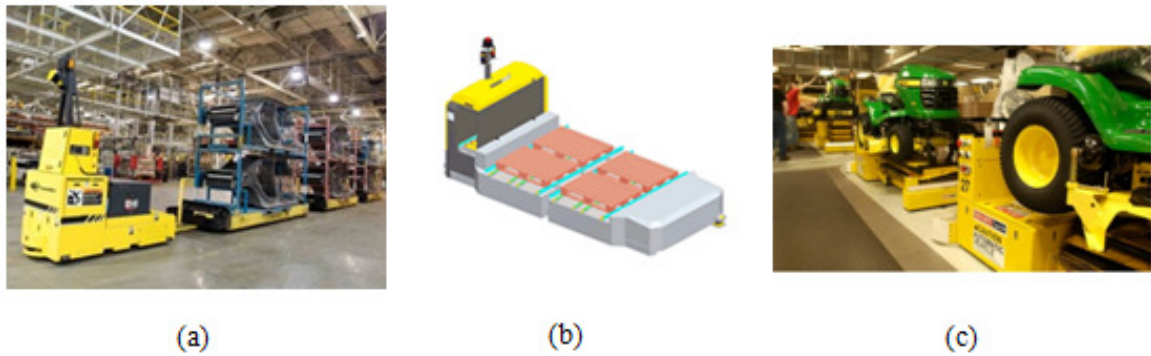


Figura 3.1. Tipos de AGVs: (a) reboque (JBT, 2009), (b) carga de unidade (ELETTRIC80, 2009), (c) linha de montagem (JBW, 2009)

Os AGVs para carga de *pallet*, figura 3.2 (a), substituem as tradicionais empilhadeiras operadas por humanos. Eles podem interagir com vários elementos do ambiente de manufatura, além de serem capazes de colocar e retirar *pallets* do chão. Devido ao seu grande potencial de elevação vertical, são bastante utilizados nos processos de estocagem.

O transporte de cargas leves e de tamanhos reduzidos é feito através dos AGVs para cargas leves, figura 3.2 (b). As configurações existentes para estes veículos variam de acordo com o método utilizado no manejo das cargas, que podem ser do tipo rebocador, plataforma, múltiplas prateleiras ou carregamento e descarregamento por elevação. Eles são ideais para operações em plantas de tamanhos reduzidos dedicadas à manufatura leve e escritórios.



Figura 3.2. Tipos de AGVs: (a) carga de pallet (ELETTRIC80, 2009), (b) cargas leves (AUTOMOTION, 2009)

### 3.1.2 Sistema de navegação

Os veículos possuem sistemas de navegação próprios que permitem que eles se desloquem a pontos específicos da planta de acordo com as rotinas programadas e as permissões do controlador central. A nave-

gação é auxiliada por marcações em alguns pontos do percurso. Estes pontos indicam ao AGV a sua localização e a partir desta informação o mesmo consegue perceber as tarefas que devam ser cumpridas a partir dali. Como exemplos de ações a serem alertadas pelas marcações têm-se a indicação de pontos de parada, de mudança da velocidade e pontos de decisão de caminhos. São várias as tecnologias empregadas nos sistemas de navegação dos AGVs: condutores elétricos, fita ou pintura magnética, ótica, feixe de laser, inercial, etc (Groover, 2007).

A navegação através de guias de condutores elétricos surgiu com os primeiros AGVs e é bastante confiável, apresentando vida útil prolongada e manutenção mínima ao longo do uso (Mitchell, 1998). Este sistema é formado por um percurso de condutores elétricos embutidos no chão através de uma fenda que são ligados a uma fonte de energia. Os sensores dos AGVs captam o campo magnético irradiado pelas guias, o qual servirá de referência para as instruções de controle de direção do veículo. Este tipo de navegação não oferece flexibilidade na mudança das rotas dos AGVs, pois há implicações construtivas na implantação das guias de navegação. Observa-se que a maioria dos novos sistemas de navegação são guiados sem a utilização de condutores elétricos como guias.

AGVs que utilizam a fita ou pintura magnética como método de navegação são equipados com sensores que detectam o pequeno fluxo magnético gerado por estes meios (Mitchell, 1998). O sistema de controle de direção dos AGVs, a partir das informações coletadas pelo sensor, fará com que o mesmo se desloque de modo centralizado ao percurso magnético. Essa tecnologia é bastante flexível e caso seja necessário modificar a rota dos AGVs, é possível fazê-lo em pouco tempo e sem alterações construtivas na planta industrial. No entanto, esse método não apresenta alta durabilidade. Os percursos onde existe grande tráfego são bastante danificados com o tempo e solicitam freqüentes reparos. Sujeira nas guias magnéticas também pode ocasionar problemas de leitura para os sensores. A aplicação deste método de navegação é recomendada para escritórios e indústrias leves, onde o percurso é menos provável de sofrer danos.

A navegação ótica é similar à feita por fita magnética, diferenciando-se apenas no tipo de sensor utilizado para aquisição de dados. Dos métodos de navegação ótica utilizados pode-se citar: alto contraste entre o chão e a linha, fita de materiais reflexivos, filme químico sensível a luz ultravioleta, etc. (Katevas, 2001). Ambos os métodos baseiam-se na



navegação a partir da diferença de intensidade luminosa entre o chão e o percurso. Esse método é passível dos mesmos problemas mencionados na navegação por fita ou pintura magnética.

O sistema de navegação a laser não necessita de percursos demarcados sobre o chão. A referência no deslocamento é obtida a partir de um sensor de triangulação a laser (Alavudeen e Venkateshwaran, 2008). O sensor é posto num ponto alto do veículo e emite um feixe de laser que rotaciona 360 graus. Através da reflexão do feixe por alvos reflexivos implantados em pontos estratégicos da planta industrial o controlador, a partir dos ângulos de reflexão, obtém uma posição de referência do veículo. O AGV possui em sua memória um mapa do percurso a seguir e com as informações coletadas pelo sensor o controlador mantém o veículo em curso. Quando o AGV não consegue coletar dados suficientes para determinar sua posição, utiliza-se o método de navegação estimada (*dead reckoning*). Este método determina a posição do AGV tomando como referência a posição anterior conhecida (Groover, 2007). O AGV calcula a sua posição atual considerando a sua velocidade de deslocamento e o percurso de deslocamento, além de recorrer à posição que foi previamente estimada.

Outro meio de navegação que dispensa o uso de guias de deslocamento é o método inercial (Mitchell, 1998). O AGV se desloca de acordo com um mapa pré-programado do percurso na sua memória. Um giroscópio detecta as mínimas variações no plano de deslocamento do AGV. Através de pontos magnéticos embutidos no chão, o AGV consegue informações para fazer as correções necessárias para se manter no percurso.

### 3.1.3 Controlador

Um sistema de AGV pode apresentar mecanismo de controle centralizado ou descentralizado (Evers e Koppers, 1996). No primeiro caso, o veículo segue estritamente as instruções fornecidas pelo controlador central, excluindo qualquer tipo de tomada decisões no nível dos AGVs. Já o controle descentralizado, fornece certa liberdade aos veículos para processar decisões independentemente como, por exemplo, no cálculo da melhor rota para alcançar o ponto de coleta ou entrega de materiais (Ye, et al., 2000).

Ao receber a sinalização para o transporte de materiais entre dois pontos, o controlador do sistema de AGV deve realizar uma série de tarefas para garantir a coordenação do sistema: escolher um veículo para realizar o transporte; determinar a rota de deslocamento do mesmo; planejar os tempos de passagem nos trechos do percurso para evitar colisão; verificar se há bateria suficiente para completar a tarefa; determinar o local de parada do veículo após a conclusão do transporte (Vis, 2006).

### 3.1.4 Sistema de comunicação

Para o perfeito funcionamento do sistema de AGV, é necessário que a comunicação entre as partes envolvidas possa ocorrer de modo confiável e eficiente. A comunicação pode acontecer entre o controlador central e os veículos, entre dois veículos – controle descentralizado – e entre dispositivos locais e veículos – este último exemplificado pela coordenação entre ponto de entrega e coleta de carga e um veículo.

Através de envio de mensagens o sistema de AGV atribui tarefas aos veículos, gerencia o tráfego, avalia condições de falhas e viabiliza a programação dos veículos, etc. (FROG, 2010). A comunicação pode partir tanto dos veículos quanto do controlador central. Numa comunicação padrão são trocadas informações referentes à identificação do veículo, localização do veículo, *status* de carga do veículo e condições de tráfego.

A comunicação via ondas de rádio é a mais comum em sistemas de AGV. Ela oferece flexibilidade e continuidade, além de permitir que os veículos sejam programados sem interromper a execução de todo o processo. Atualmente a comunicação via infravermelho não é mais utilizada para estabelecer contato entre os veículos e o controlador central, no entanto ela é aplicada na comunicação local.

## 3.2 Problemas de controle de sistema de AGV

Conforme visto na seção 3.1.3, o controlador lida com uma série de tarefas para coordenar o sistema. O controle de sistemas de AGV é complexo e abrange áreas distintas de conhecimento como, por exemplo, escalonamento, prevenção e impedimento de *deadlocks* e colisões, roteamento de veículos e regras de despacho (Vis, 2006).

### 3.2.1 Despacho e roteamento dos veículos

O despacho do veículo para transportar um material normalmente busca a alternativa mais eficiente em relação à menor distância (Le-Anh e De Koster, 2006). Quando o controle do sistema é centralizado, o controlador central monitora todas as informações referentes à evolução dos veículos no sistema. Ele sabe as posições dos AGVs, os pontos de entrega dos materiais, a agenda de eventos dos AGVs, etc. A partir destas informações, o controlador central determina, através de regras específicas, qual é o veículo mais vantajoso para transportar determinada carga. No despacho descentralizado, o sistema central não especifica qual veículo deve se deslocar para receber uma carga. Os AGVs baseiam-se em informações locais e podem se deslocar a qualquer estação de transferência.

Após o veículo completar uma tarefa e não receber nenhum outro pedido de coleta, o mesmo deve ser retirado da malha principal a fim de evitar transtornos a outros veículos em operação (Vis, 2006). Para o posicionamento dos veículos ociosos utiliza-se a regra de zona central, do laço circulatório ou do ponto de entrega. A primeira consiste em estabelecer um local centralizado de alocação dos veículos, onde todos os AGVs ociosos são direcionados. A segunda opção sugere a existência de laços circulatórios em alguns pontos do percurso de maneira que o veículo permaneça neste trajeto até que receba um novo pedido de coleta de material. A terceira regra determina que o AGV simplesmente permaneça no ponto de entrega da carga até receber outra tarefa.

### 3.2.2 Gerenciamento de tráfego

O gerenciamento de tráfego é responsável pelo deslocamento seguro dos AGVs através do mapa de percurso do sistema, evitando *deadlocks* e colisões entre os veículos. De acordo com a literatura, o gerenciamento de tráfego é estudado a partir de três pontos de vista: configuração de layout da rede de percursos, controle por zonas e estratégias de roteamento (Vis, 2006). Em ambas a preocupação de se evitar *deadlock* se mantém.

O percurso convencional é uma rede de caminhos conectando todas as estações que necessitam de deslocamento de material, podendo haver junções, interseções e atalhos. Alternativamente a esta proposta,

podem ser gerados projetos de *layout* que possibilitem regras de controle mais simples, o que é o caso dos *layouts* de laço único e *tandem*. No laço único não há interseções e só existe um circuito de deslocamento para o veículo. O *layout* do tipo tandem é uma combinação de várias zonas não sobrepostas que se conectam através de estações de transferências. Cada zona é servida por um AGV e durante o transporte de um material, pode ser necessária a passagem por vários AGVs até ser alcançado o destino final.

O controle por zona é bastante utilizado na prática. Este método divide a rede de deslocamento em zonas, de modo que somente é permitida a entrada de um veículo por vez em cada zona. Caso um AGV precise entrar em uma zona já ocupada, o mesmo permanece em espera até que a zona seja liberada. Em contraste com este método, existem técnicas de controle por zona de sistemas de grande porte que tornam possível o acesso de vários AGVs simultaneamente numa mesma zona (Evers e Koppers, 1996).

Um dos objetivos do estudo de estratégias de roteamento é reduzir o tempo de transporte dos AGVs. Os algoritmos de roteamento podem ser estáticos ou dinâmicos. O primeiro grupo de algoritmo define antecipadamente a melhor – normalmente a menor distância – rota para transportar cargas entre dois pontos. Sempre que for solicitado o transporte entre estes pontos, é usada esta rota. Os algoritmos dinâmicos calculam a rota de entrega baseado em informações de tempo real das condições do tráfego, podendo existir várias rotas para os mesmos pontos de deslocamento.

### **3.2.3 Gerenciamento da carga da bateria dos AGVs**

A recarga das baterias dos AGVs é feita em estações especiais, normalmente localizadas nos locais centrais de estacionamento (Le-Anh e De Koster, 2006). Existem três modalidades de recarga de bateria sem a intervenção humana: recarga de oportunidade, recarga automática e uma combinação das duas anteriores. Na recarga de oportunidade o AGV aproveita o seu tempo ocioso para efetuar tal operação. Já na recarga automática, a bateria do AGV deve atingir um valor mínimo para poder ser recarregada. Existe também a opção manual de troca de baterias que segue a mesma regra da recarga automática, porém com auxílio de um operador para realizar a troca.

Numa abordagem de controle centralizado, dentre as mensagens transmitidas pelo AGV são enviadas informações situando o controlador central sobre o nível de carga da bateria. O controlador central programa o desvio do veículo para a estação adequada de acordo com a regra adotada.

### 3.2.4 Sincronização entre operações

A sincronização no sistema de AGV é necessária para o estabelecimento de relações seguras entre os seus componentes. Durante o carregamento e descarregamento de materiais, deve existir um acordo entre as partes envolvidas e um protocolo que deva ser respeitado durante a realização destas operações. Não se deseja, por exemplo, que um veículo possa se movimentar enquanto estiver descarregando uma carga em uma estação.

## 3.3 Conclusão do capítulo

Sistemas de AGV oferecem enorme flexibilidade como alternativa para o transporte de materiais. No controle destes sistemas, deseja-se que o transporte seja feito de forma rápida e sem colisões e bloqueios. As pesquisas têm permitido o desenvolvimento de sistemas de rápida reestruturação e manutenção, no entanto o estudo do controle de sistemas de AGV é complexo e exige a interação de diversos campos de conhecimentos.

Neste capítulo foram expostas as características gerais de sistemas de AGV e os problemas de controle relacionados. No próximo capítulo, o estudo do controle de sistemas de AGV é direcionado para a resolução dos problemas de gerenciamento de tráfego – prevenção de colisões e bloqueios –, gerenciamento da bateria dos veículos, regras de roteamento para os veículos e sincronização entre as operações dos veículos e das estações de transferências.

A solução destes problemas é feita utilizando a TCS e toma como exemplo um sistema de AGV, em *layout* de percurso convencional, que contém dois veículos e duas estações de transferência. O controle deste sistema é centralizado e o gerenciamento de tráfego é feito através da abordagem por zonas, com auxílio de regras de roteamento. Não foram especificadas regras de despacho para os veículos e os mesmos sempre

permanecem em circulação pelo percurso. Também se lida com o gerenciamento da bateria dos veículos, que são direcionados para a estação de recarga, por meio de regras de roteamento, onde é feita a troca manual das baterias.

## Capítulo 4

### Aplicação da TCS a um sistema de AGV

Como visto no capítulo 2, a abordagem modular da TCS possibilita a síntese de supervisores locais ótimos. Neste capítulo é descrito o processo de síntese e implementação de um conjunto de supervisores modulares locais para controlar um sistema de AGV de pequeno porte (Silva, et al., 2010). Os supervisores atuam no impedimento de colisões entre os veículos, no roteamento dos veículos, no gerenciamento da bateria dos veículos e na interação entre os veículos e as estações de carga e descarga. O problema a ser tratado foi inspirado pela publicação de (Hsieh e Kang, 1998) e a metodologia adotada para a solução foi a mesma usada em (Silva e Queiroz, 2009).

#### 4.1 O sistema de AGV

O sistema de AGV, mostrado na figura 4.1, corresponde a uma bancada experimental desenvolvida em parceria, entre o Instituto Federal Santa Catarina (IFSC) e o Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina. O desenvolvimento da bancada foi inspirado no problema apresentado por Hsieh e Kang (1998). No *layout* de percurso há ímãs identificando os pontos importantes do trajeto como, por exemplo, cruzamentos e estações de carga e descarga de *pallets*.

O sistema é composto por dois veículos – AGV1 e AGV2 – e pelas estações de carga – ponto B –, descarga – ponto G – e reabastecimento de bateria – ponto L. Cada AGV é controlado por um microcontrolador interno e possui sensores para monitorar o nível da bateria, navegar pelas guias do percurso e identificar os ímãs do trajeto. Os AGVs são equipados com sistema de rolagem para efetuar operações de carga e descarga de *pallets*. O deslocamento dos AGVs é bidirecional entre o trecho HKL e conforme o sentido indicado pelas setas nos demais trechos. O sistema é inicializado com o AGV1 no trecho AB e o AGV2 no trecho DE, e os pontos de parada dos veículos coincidem com a posição dos ímãs no percurso.

As estações de carga e descarga são equipadas com sistema de rolagem para efetuar suas respectivas operações. Os AGVs são carregados com *pallets* na estação de carga, a qual possui um sensor para identificar a chegada dos *pallets*. A estação de reabastecimento de bateria é o local para a troca manual de bateria dos veículos e comporta um AGV. Os veículos devem se deslocar ao ponto L ao atingir o nível crítico de bateria e aguardar até que a mesma seja trocada.

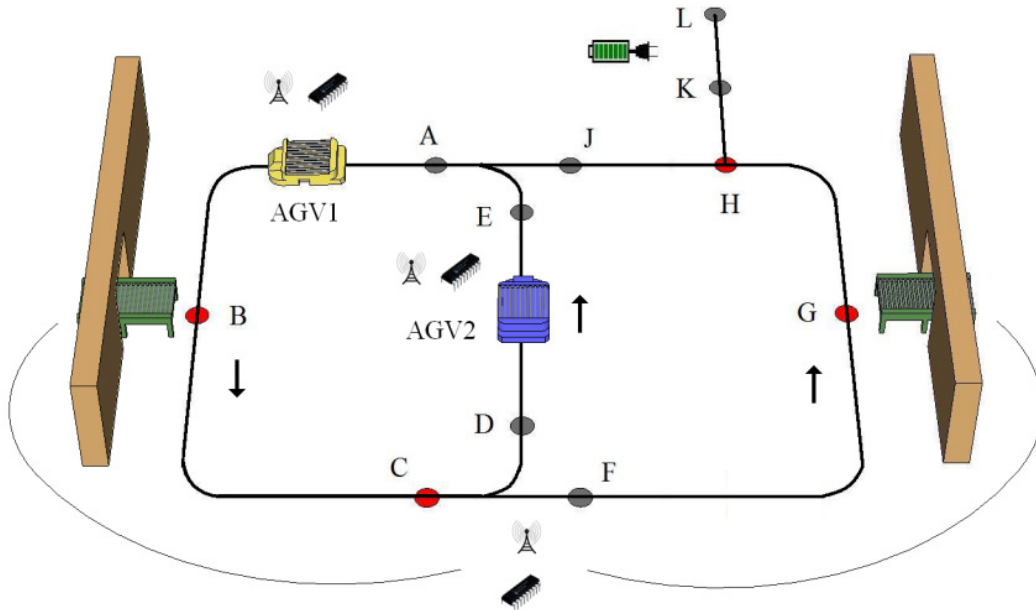


Figura 4.1. Sistema de veículos auto-guiados

No sistema há um microcontrolador central que comanda as estações de carga e descarga, além dos microcontroladores dos AGVs já mencionados. A comunicação entre os AGVs e o controlador central é feita por radiofrequência. A função dos AGVs é transportar os *pallets* que chegam à estação de carga *B* até a estação de descarga *G*. Para isso, o sistema deve seguir algumas especificações que estão separadas por categorias. Os nomes entre parênteses associam a descrição em linguagem natural da especificação com os geradores que a descrevem, os quais serão apresentados na seção 4.3.

A) Impedimento de colisões:

- Os AGVs não podem ocupar simultaneamente o mesmo trecho ( $E_{COLi}$ , para  $i = 1, \dots, 9$ ).

B) Manipulação de *pallets*:

- A operação de carga é proibida se:
  - Não houver chegado *pallet* na estação de carga ( $E_{MAN1}$ );



- Não houver veículo no ponto B ( $E_{MAN8}$ );
  - Não houver bateria suficiente no veículo em B ( $E_{MAN2}$ ,  $E_{MAN3}$ );
  - A operação de descarga é proibida se não houver veículo na posição G ( $E_{MAN9}$ );
  - Uma vez iniciada a carga (descarga), não é permitido que os AGVs deixem o ponto B (G) até que esta operação seja finalizada ( $E_{MAN6}$ ,  $E_{MAN7}$ );
  - Não deve ocorrer *underflow* ou *overflow* de *pallets* no compartimento de carga dos AGVs ( $E_{MAN4}$ ,  $E_{MAN5}$ );
  - Caso haja *pallet* na estação de carga, o AGV com bateria e livre não pode passar por B sem carregar ( $E_{MAN10}$ ,  $E_{MAN11}$ ).
- C) Roteamento dos AGVs:
- É proibido o deslocamento de AGVs entre o trecho HKL, salvo na condição de bateria com pouca carga ( $E_{ROT1}$ ,  $E_{ROT2}$ );
  - Não é autorizada a passagem do AGV pelo trecho CD e GH se estiver transportando um *pallet* ( $E_{ROT3}$ ,  $E_{ROT4}$ ).

Deseja-se também que o sistema não apresente bloqueio (*deadlock* ou *livelock*). *Deadlock* refere-se ao travamento do sistema devido à ausência de recurso que possa dar continuidade à operação do sistema. Pode ocorrer, por exemplo, quando um veículo A necessita acessar o trecho do veículo B e o veículo B necessita acessar o trecho do veículo A. Já o *livelock* refere-se à impossibilidade de alcançar os objetivos do sistema, apesar de não ocorrer um travamento que impeça a evolução do sistema. Ocorre, por exemplo, quando um AGV carregado não consegue acessar um percurso que leve à estação de descarga.

Com base nos critérios descritos, deseja-se sintetizar e implementar um controlador não bloqueante e minimamente restritivo para este sistema de AGV. O esquema visto na figura 4.2 resume a metodologia utilizada para obter o sistema de controle. Esta metodologia é uma combinação da metodologia já conhecida para a síntese de supervisores ótimos (Ramadge e Wonham, 1989), apresentada no capítulo 2, e da utilização de ferramentas de emulação para minimizar os erros de modelagem e codificação.

Conforme já mencionado, na TCS há três passos para se obter os supervisores: modelagem da planta, modelagem das especificações de controle e síntese dos supervisores ótimos. Falhas durante a modelagem da planta e especificações implicam na síntese de supervisores incorre-

tos. A ferramenta DESEM, discutida na seção 4.5, permite acompanhar a atuação dos supervisores na planta seqüencialmente, possibilitando identificar erros ocorridos na modelagem. Esta ferramenta recebe como entrada de dados a planta e os supervisores na forma de geradores e responde com os eventos possíveis de ocorrer conforme a planta evolui.

O ciclo de modelagem, síntese e emulação dos supervisores, figura 4.2, se mantém até que o sistema se comporte conforme desejado. Posteriormente passa-se para a implementação em linguagem C destes supervisores. O código dos supervisores é emulado num ambiente gráfico, discutido na seção 5.3, que responde com animações gráficas para o que seriam rotinas operacionais do sistema real. Após encerrar o ciclo de emulações e codificação, parte-se para a implementação do código na bancada experimental do sistema de AGV.

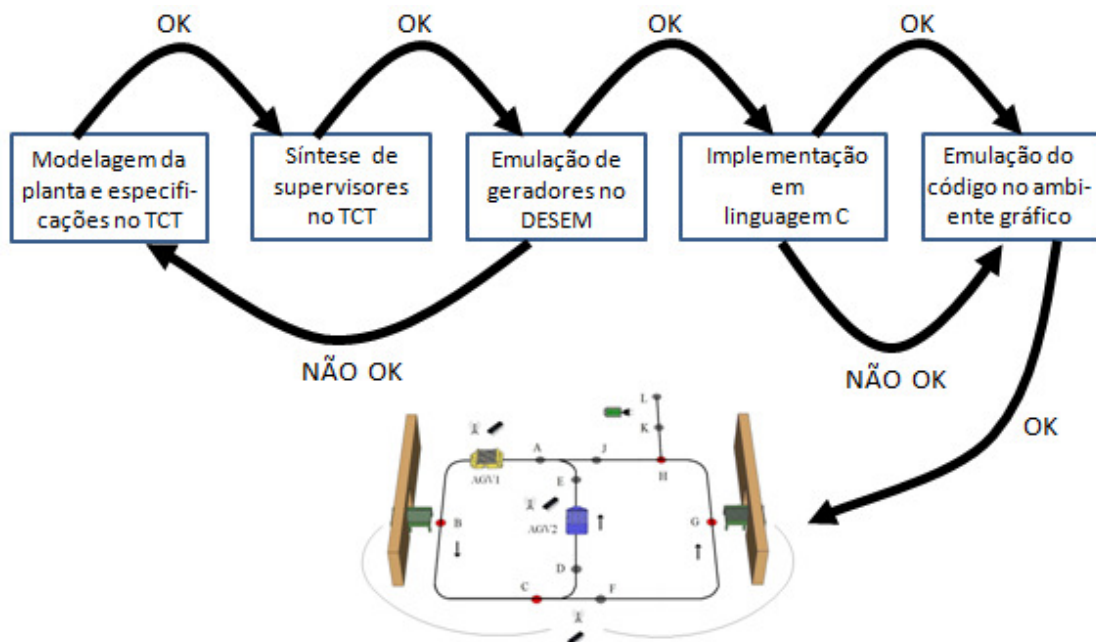


Figura 4.2. Metodologia para obtenção do sistema de controle

Este capítulo concentra-se na obtenção dos supervisores modulares locais para o sistema, destacando os processos de modelagem, síntese e emulação. A descrição da implementação dos supervisores locais, juntamente com o ambiente gráfico para emulação do código dos supervisores, é abordada no capítulo 5.

## 4.2 Modelagem do sistema de AGV

Os AGVs são modeladas considerando os eventos que representam a passagem entre os trechos do *layout* de percurso e a chegada nos

ímãs onde há estação de carga ou descarga. Os eventos controláveis são representados por  $x_{\langle AGV \rangle}ly$ , para  $x,y \in \{a,\dots,h, j, k,l\}$  e  $\langle AGV \rangle \in \{1,2\}$ , sendo  $x$  o ímã de origem do deslocamento,  $\langle AGV \rangle$  a identificação do veículo,  $l$  o índice que indica a passagem do sensor magnético do AGV para o nível baixo (*low*), que ocorre assim que o AGV deixa o ímã, e  $y$  o ímã de destino do deslocamento. Os eventos não controláveis são expressos por  $y_{\langle AGV \rangle}h$ , sendo  $h$  o índice que indica a passagem do sensor magnético do AGV para o nível alto (*high*), que ocorre quando o AGV chega ao ímã. Observando o gerador do AGV1 –  $G_{AGV1}$  –, ilustrada na figura 4.3, interpreta-se o evento  $a1lb$  como o início do deslocamento do AGV1 entre o trecho AB. Já os eventos não controláveis  $b1h$  e  $g1h$  sinalizam a chegada do AGV1 sobre os ímãs  $B$  e  $G$  respectivamente.

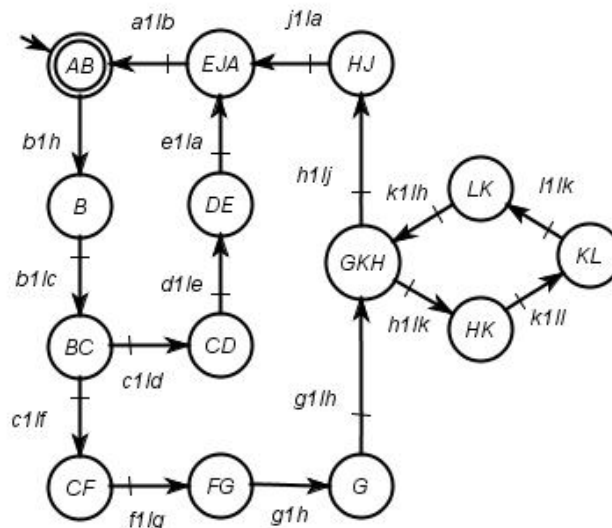


Figura 4.3. Gerador  $G_{AGV1}$

Os estados dos geradores dos AGVs indicam o segmento em que os veículos se localizam no percurso, com exceção dos estados alcançados por eventos não controláveis, que informam quando os AGVs estão nas posições  $B$  e  $G$ . Estas posições aparecem no modelo do gerador porque para coordenar as operações de carga e descarga é necessário saber quando os AGVs as atingem. Os trechos que formam uma junção de percurso como, por exemplo, GH e KH são representados por um único estado – GKH –, já que a saída da junção é a mesma para ambos. O estado inicial do modelo do AGV1 representa o trecho AB. Esta opção de modelo, que omite a chegada dos AGVs nos pontos não significativos para o controle do sistema, oferece economia de estados e conseqüentemente reduz o custo computacional na síntese dos supervisores em relação a um modelo mais completo. No entanto, esta forma de modelo deve ser complementada no nível de implementação com a infor-

mação da chegada dos veículos nos ímãs não representados pelos modelos dos AGVs – A, C, D, E, F, H, J, K, L – para que dois eventos controláveis subsequentes não sejam disparados antes da chegada dos veículos aos respectivos pontos. O gerador do AGV2 –  $G_{AGV2}$  –, figura 4.4, apresenta modelo análogo ao gerador  $G_{AGV1}$ , alterando-se apenas os índices numéricos dos eventos para 2 e a posição inicial do veículo, que passa a ser o trecho DE. Os AGVs devem poder retornar a posição inicial, desta forma marcam-se estes estados.

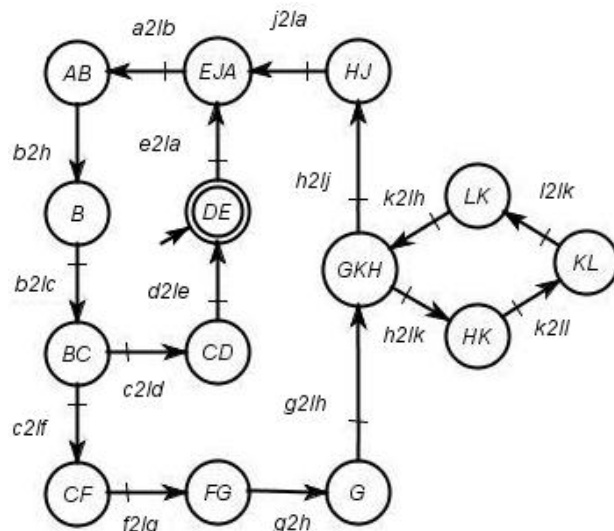


Figura 4.4. Gerador  $G_{AGV2}$

As estações de carga e descarga são modeladas considerando o início e fim de suas operações, pois estes são os eventos relevantes para a coordenação do sistema. O gerador da estação de carga -  $G_{CARGA}$  – é exibida na figura 4.5 (a). O modelo considera operações de carga diferenciadas para cada veículo. Os eventos  $ac1$  e  $bc1$  equivalem ao início e fim da operação de carga do AGV1 respectivamente. Interpretação similar é feita para os outros eventos em relação ao gerador do AGV2. O gerador da estação de descarga –  $G_{DESCARGA}$  –, figura 4.5 (b), segue a mesma lógica do gerador  $G_{CARGA}$ , sofrendo alterações somente no índice  $c$ , que passa a ser  $d$ . Marcam-se os estados iniciais destas plantas porque a operação de carga ou descarga deve ser finalizada após o seu início.

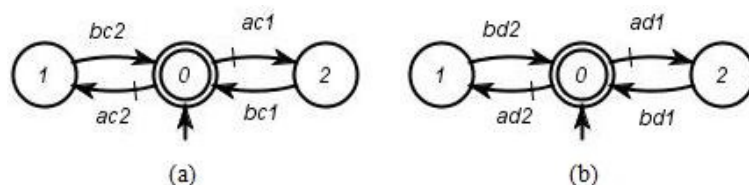


Figura 4.5. Geradores (a)  $G_{CARGA}$  e (b)  $G_{DESCARGA}$

As plantas dos sensores de chegada de *pallets* –  $G_{SENT}$  – e nível de carga da bateria dos AGVs –  $G_{SBAT1}$  e  $G_{SBAT2}$  – são exibidos na figura 4.6. Os eventos dos sensores são modelados como não controláveis e podem ocorrer a qualquer instante no sistema. A marcação dos mesmos representa a idéia de tarefa concluída.

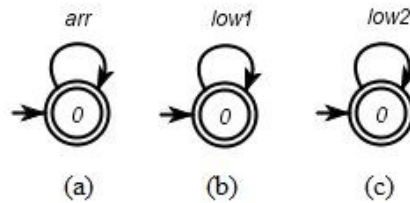


Figura 4.6. Geradores (a)  $G_{SENT}$ , (b)  $G_{SBAT1}$ , (c)  $G_{SBAT2}$

### 4.3 Modelagem das especificações do sistema de AGV

As especificações do sistema de AGV se concentram na resolução dos problemas citados na seção 4.1. Para o impedimento de colisões é utilizada a estratégia de controle por zona, a qual divide o layout de percurso em várias zonas de controle. Como explicado no capítulo 3, cada zona permite o deslocamento de um veículo por vez e caso a zona esteja ocupada, o AGV permanece em espera até que a mesma seja liberada (Mulcahy, 1999).

Para o layout do sistema foram definidas nove zonas de controle: AB, BC, CDF, DE, EJA, FG, GKH, HJ, HKL. É definida uma especificação de exclusão mútua entre os AGVs para cada zona –  $E_{COLi}$ ,  $i = 1, \dots, 9$  – conforme o modelo genérico da figura 4.7, com exceção da zona HKL. No entanto, quando se especifica que um AGV inicia no interior de uma zona, deve-se alterar o estado inicial da especificação de exclusão mútua da zona para o estado que indica a presença de um veículo naquela zona. Exemplificando com a zona AB, os eventos de acesso seriam  $b1lc$  e  $b2lc$  e os de saída  $a1lb$  e  $a2lb$ . Ocorrendo  $a1l$ , fica proibida a execução de  $a2l$  até que  $b1l$  aconteça. Como o AGV já acessou a zona, o estado 2 passa a ser o estado inicial desta especificação.

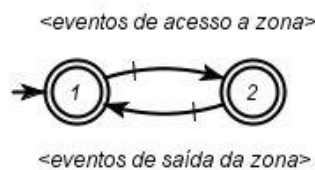


Figura 4.7. Especificação de exclusão mútua genérica

Utilizando a especificação da figura 4.7 para a zona HKL, caso um AGV entre nesta zona o evento  $h_{\langle AGV \rangle lj}$  para o outro AGV é desabilitado até que o AGV saia da zona HLK. No entanto, deseja-se que, enquanto um AGV estiver em HLK, o percurso HJ possa ser acessado pelo outro AGV. Desta forma, a especificação para a zona HKL –  $E_{COL9}$  – foge do padrão da figura 4.7. A especificação para esta zona toma a forma da figura 4.8. Nela a exclusão mútua é tratada sem impedir o tráfego pelo trecho HJ, de modo que caso o AGV1 encontra-se na zona HKL – estado 1 – o AGV2 pode seguir pela zona HJ – evento  $h2lj$  habilitado. Para o AGV2 em HKL, constrói-se um modelo análogo observando os eventos habilitados no estado 2.

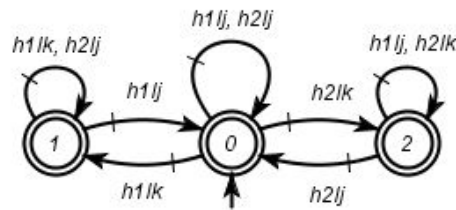


Figura 4.8. Especificação de exclusão mútua  $E_{COL9}$  para a zona HKL

Nas especificações relativas à manipulação dos *pallets* são tratados os problemas de interação entre as estações de carga, descarga e os AGVs. A especificação  $E_{MAN1}$ , figura 4.9 (a), determina que as operações de carga dos veículos – eventos  $ac_{\langle AGV \rangle}$  – não podem ocorrer enquanto não houver chegada *pallet* na estação de carga – evento  $arr$ . Conforme as especificações  $E_{MAN2}$  e  $E_{MAN3}$ , ilustradas genericamente na figura 4.9 (b), a operação de carga não é autorizada quando os AGVs atingem o nível crítico de bateria – evento  $low_{\langle AGV \rangle}$ .

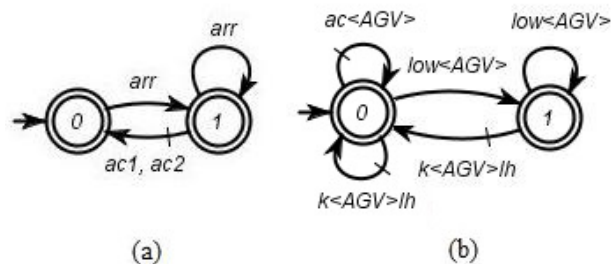


Figura 4.9. Especificação (a)  $E_{MAN1}$  e especificação genérica para (b)  $E_{MAN2}$  e  $E_{MAN3}$

Visando evitar *overflow* e *underflow* de *pallets*, conforme visto na figura 4.10,  $E_{MAN4}$  e  $E_{MAN5}$  desabilitam a operação de descarga dos AGVs se eles não foram carregados anteriormente e desabilitam um novo carregamento até que os mesmos sejam descarregados.

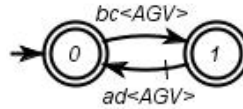


Figura 4.10. Especificação genérica para  $E_{MAN4}$  e  $E_{MAN5}$

Preocupando-se com a sincronia da operação de carga, a especificação  $E_{MAN6}$  da figura 4.11 (a) desabilita os eventos  $b_{<AGV>lc}$  após o início da operação de carga – evento  $ac_{<AGV>}$  –, não permitindo que os veículos se desloquem antes de finalizar a operação de carga – evento  $bc_{<AGV>}$ . Para a operação de descarga, é modelada a especificação  $E_{MAN7}$ , figura 4.11 (b), que age segundo a mesma idéia.

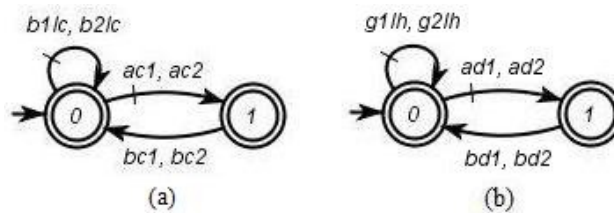


Figura 4.11. Especificações (a)  $E_{MAN6}$  e (b)  $E_{MAN7}$

$E_{MAN8}$ , figura 4.12 (a), desabilita  $ac1$  e  $ac2$  se os AGVs não esti-

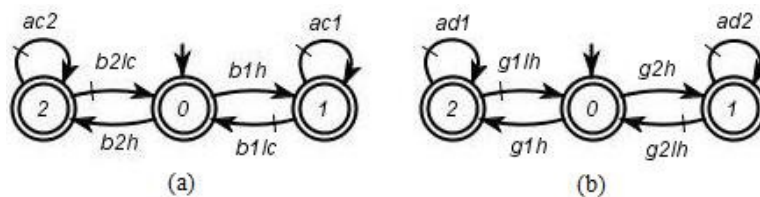


Figura 4.12. Especificações (a)  $E_{MAN8}$  e (b)  $E_{MAN9}$

verem sobre o ímã B – evento  $b_{<AGV>h}$ . Especificação similar,  $E_{MAN9}$ , figura 4.12 (b), é feita para a estação de descarga.

Quando houver *pallet* na estação de carga, as especificações  $E_{MAN10}$  e  $E_{MAN11}$  proíbem o AGV que apresente bateria suficiente e esteja livre passar para o trecho seguinte – evento  $b_{<AGV>lc}$  até que o mesmo seja carregado. Para obter estas especificações, modelaram-se os geradores da figura 4.13. O gerador da figura 4.13 (a) indica se há *pallet* – estado 2 – ou não na estação de carga. Já o da figura 4.13 (b) mostra se a bateria do AGV é suficiente – estado 3 – ou atingiu o nível crítico. Por fim o gerador visto na figura 4.13 (c) representa se o AGV transporta um *pallet* ou está livre – estado 5. Fazendo a composição síncrona destes geradores, e inserindo um auto-laço com o evento  $b_{<AGV>lc}$  em todos os seus estados, com exceção daquele que representa estar simultaneamente nos estados 2, 3 e 5 dos geradores descritos obtêm-se as especificações relacionadas a cada AGV.

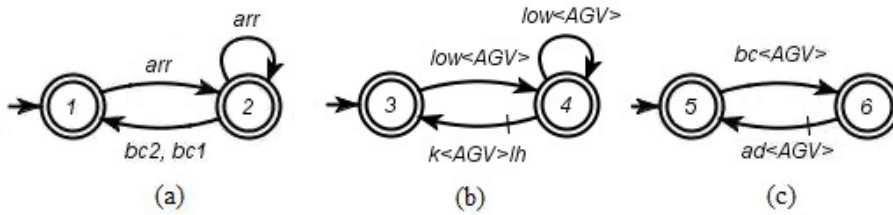


Figura 4.13. Geradores intermediários para se obter  $E_{MAN10}$  e  $E_{MAN11}$

As especificações de roteamento são responsáveis por desabilitar alguns percursos do sistema de modo que os caminhos autorizados conduzam de modo seguro e eficiente ao objetivo desejado. A especificação de roteamento  $E_{ROT1}$ , apresentada na figura 4.14 (a), desabilita os eventos  $c1ld$  e  $h1lj$  e habilita  $h1lk$ , forçando o AGV1 se deslocar para a estação de abastecimento de bateria quando sua carga estiver baixa. Esta especificação também se preocupa com a possibilidade de existir um AGV na zona HKL e desabilita o evento  $g1lh$ , fazendo que o AGV1 aguarde no ímã G até que a zona esteja livre. Uma especificação equivalente a esta –  $E_{ROT2}$  –, figura 4.14 (b), é feita para o AGV2.

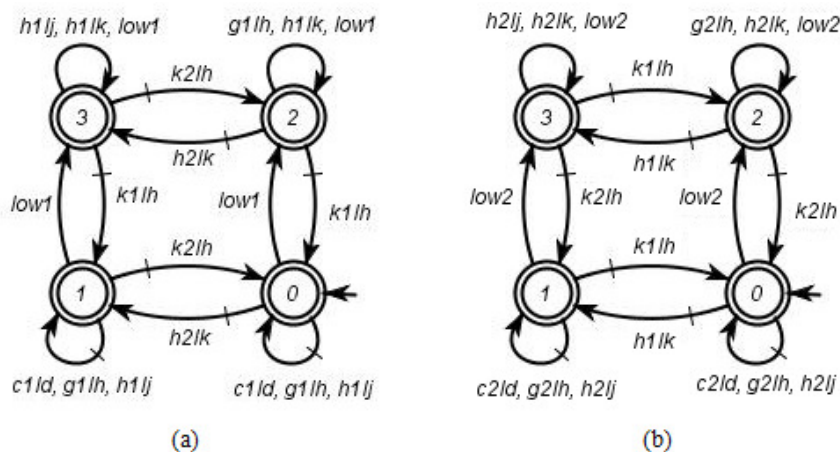


Figura 4.14. Especificações (a)  $E_{ROT1}$  e  $E_{ROT2}$

Para lidar com o transporte dos *pallets* entre as estações de carga e descarga, têm-se as especificações de roteamento  $E_{ROT3}$  e  $E_{ROT4}$ , expressas genericamente na figura 4.15. Elas desabilitam os eventos  $c_{<AGV>ld}$  e  $g_{<AGV>lh}$  após os AGVs serem carregados – evento  $bc_{<AGV>}$  –, conduzindo-os diretamente à estação de descarga. Estes eventos são habilitados novamente após a descarga do *pallet* – evento  $ad_{<AGV>}$ . Nota-se que esta especificação força o AGV que apresente nível de bateria baixo e esteja transportando um *pallet* descarregar a carga antes de prosseguir a estação de recarga da bateria.



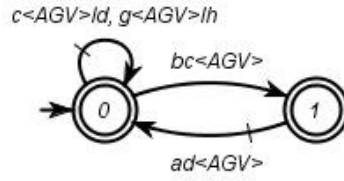


Figura 4.15. Especificações  $E_{ROT3}$  ( $\langle AGV \rangle = 1$ ) e  $E_{ROT4}$  ( $\langle AGV \rangle = 2$ )

As restrições de não bloqueio são tratadas durante o cálculo dos supervisores, que leva em conta os estados marcados, os quais indicam a conclusão de tarefas.

Como o percurso não é extenso, não foram determinadas regras de deslocamento para os AGVs ociosos. Os veículos podem circular livremente pelo percurso, salvo no trecho HKL.

Tabela 4.1. Plantas locais para o sistema de AGV

$G_{locx}$	$\parallel G_x$
$G_{locCOLi}, i = 1, \dots, 9$	$G_{AGV1} \parallel G_{AGV2}$
$G_{locMAN1}$	$G_{CARGA} \parallel G_{SENT}$
$G_{locMAN2}$	$G_{AGV1} \parallel G_{CARGA} \parallel G_{SBAT1}$
$G_{locMAN3}$	$G_{AGV2} \parallel G_{CARGA} \parallel G_{SBAT2}$
$G_{locMAN4}, G_{locMAN5}$	$G_{CARGA} \parallel G_{DESCARGA}$
$G_{locMAN6}, G_{locMAN8}$	$G_{AGV1} \parallel G_{AGV2} \parallel G_{CARGA}$
$G_{locMAN7}, G_{locMAN9}$	$G_{AGV1} \parallel G_{AGV2} \parallel G_{DESCARGA}$
$G_{locMAN10}$	$G_{AGV1} \parallel G_{CARGA} \parallel G_{DESCARGA} \parallel G_{SENT} \parallel G_{SBAT1}$
$G_{locMAN11}$	$G_{AGV2} \parallel G_{CARGA} \parallel G_{DESCARGA} \parallel G_{SENT} \parallel G_{SBAT2}$
$G_{locROT1}$	$G_{AGV1} \parallel G_{AGV2} \parallel G_{SBAT1}$
$G_{locROT2}$	$G_{AGV1} \parallel G_{AGV2} \parallel G_{SBAT2}$
$G_{locROT3}$	$G_{AGV1} \parallel G_{CARGA} \parallel G_{DESCARGA}$
$G_{locROT4}$	$G_{AGV2} \parallel G_{CARGA} \parallel G_{DESCARGA}$

Ao todo, modelaram-se 24 especificações e as plantas locais, obtidas pela composição dos subsistemas afetados por elas, são expressas na tabela 4.1. Para  $x \in X = \{ COLi, i = 1, \dots, 9; MANj, j = 1, \dots, 11; ROTk, k = 1, \dots, 4 \}$  as especificações locais são definidas por  $K_{locx} = G_{locx} \parallel E_x$ .

#### 4.4 Síntese dos supervisores do sistema de AGV

Das especificações locais  $K_{locx}$  são extraídas as máximas linguagens controláveis através da operação  $SupC(K_{locx}, G_{locx})$ . Conforme visto no capítulo 2, para certificar que os supervisores modulares locais com-

portam-se de modo equivalente à solução monolítica, deve-se verificar se os mesmos são não conflitantes. No teste de não conflito, realiza-se a operação de composição síncrona entre os supervisores modulares locais e caso o gerador resultante seja não bloqueante, os supervisores modulares locais não são conflitantes. Para  $x \in X = \{ \text{COL}_i, i = 1, \dots, 9; \text{MAN}_j, j = 1, \dots, 11; \text{ROT}_k, k = 1, \dots, 4 \}$ , obteve-se  $S = \|S_{locx}$ . O gerador  $S$  é não bloqueante – portanto idêntico ao monolítico – e possui 4098 estados. Logo os supervisores sintetizados são não conflitantes.

A tabela 4.2 apresenta um resumo da síntese dos supervisores modulares locais quanto ao número de estados.

Tabela 4.2. Resumo da síntese dos supervisores do sistema de AGV

$X$	$E_x$	$G_{locx}$	$K_{locx}$	$S_{locx}$	$S_{redx}$
COL1, COL3, COL6	2	196	192	192	2
COL2, COL4, COL5, COL8	2	196	195	195	2
COL7	2	196	192	153	2
COL9	2	196	209	209	4
MAN1	2	3	6	6	2
MAN2, MAN3	2	42	84	84	2
MAN4, MAN5	2	9	18	15	2
MAN6, MAN7	2	588	588	588	2
MAN8	3	588	585	576	6
MAN9	3	588	585	576	4
MAN10, MAN11	8	126	1008	588	8
ROT1, ROT2	4	196	350	350	4
ROT3, ROT4	2	126	252	210	2

O número de estados dos supervisores reduzidos  $S_{redx}$  é próximo ou idêntico ao das especificações genéricas  $E_x$ . Isto facilita a interpretação da lógica de controle imposta pelos supervisores  $S_{redx}$ .

A síntese e a redução dos supervisores modulares locais foram feitos através da ferramenta TCT (Feng e Wonham, 2006). No processo de síntese, os cálculos não ultrapassaram os 2 minutos para o pior caso. Já a redução, o pior caso precisou de aproximadamente 2 horas para ser finalizada. Estas operações foram processadas num computador Xeon® quad core 2.33 GHz, 4 Gb RAM, Windows Vista® Home Premium 32-bit version. No apêndice deste documento é possível visualizar os 24 supervisores modulares locais do reduzidos do sistema de AGV.

## 4.5 Emulação dos supervisores

A síntese automática garante que os supervisores são controláveis, não bloqueantes e minimamente restritivos em relação às especificações, no entanto é necessário certificar que o modelo proposto está correto e completo. Neste trabalho foi utilizada uma ferramenta de emulação antes de implementar os supervisores. A emulação ajuda a visualizar a lógica proposta e permite realizar mudanças no projeto antecipadamente, evitando os problemas causados por alterações tardias durante a implementação.

Apesar da emulação dos supervisores não ser exaustiva, ela ajuda a validar a solução. Ela funciona como *feedback* no processo de síntese dos supervisores modulares locais. Quando se verifica a ausência de uma especificação ou é violada alguma regra imposta para o sistema, pensa-se em uma nova especificação que satisfaça a lógica do sistema.

DESEM é um emulador de sistemas a eventos discretos desenvolvido no Departamento de Automação e Sistemas da Universidade Federal de Santa Catarina. Com ele é possível emular a atuação dos supervisores locais gerados sobre a planta, inclusive acompanhar as ações tomadas pelos supervisores progressivamente.

No DESEM, blocos representam os subsistemas do sistema físico. Os blocos são carregados com seus respectivos geradores. Se nenhum supervisor é carregado, o sistema é emulado em malha aberta. Conforme se adiciona supervisores no DESEM, os geradores passam a atuar respeitando as restrições impostas por eles.

Até se alcançar os supervisores finais, várias especificações tiveram de ser revisadas, especialmente  $E_{ROT1}$ ,  $E_{ROT2}$ ,  $E_{MAN10}$  e  $E_{MAN11}$ , que contêm regras de controle mais complexas.

## 4.6 Expansão para sistemas de grande porte

Para expandir os modelos apresentados para sistemas de AGV de maior porte, utiliza-se a mesma idéia de modelagem aqui aplicada, levando em conta a particularidade de cada sistema. O modelo dos geradores dos AGVs é dependente do layout de percurso definido para o sistema. Já os modelos das estações de transferências são influenciados pelo número de veículos do sistema e os modelos dos sensores não sofrem alterações.

As especificações de controle também são dependentes de fatores envolvendo número de veículos e *layout* de percurso, no entanto a idéia

apresentada para a modelagem se mantém. Uma especificação de exclusão mútua, por exemplo, manteria a mesmo número de estados, porém os eventos associados a ela dependeriam do número de veículos do sistema.

A síntese de supervisores para modelos de grande porte solicitará maior demanda computacional, podendo ser impraticável quando o percurso envolver muitas zonas de controle ou elevado número de veículos.

#### **4.7 Conclusão do capítulo**

O conjunto de supervisores não bloqueantes, conforme verificado pelo teste de não conflito, obtidos através da abordagem modular local da TCS permite o funcionamento do sistema de AGV sem perda da flexibilidade operacional. Este funcionamento é ótimo no sentido de que os AGVs podem realizar todas as seqüências de operações que não são prejudiciais ao sistema – citadas na seção 4.1. A característica distribuída dos supervisores locais não só contribui para a redução da complexidade computacional no cálculo dos mesmos, em relação à abordagem monolítica, como também torna versátil a alteração ou adição de regras de controle e a implementação, a ser tratada no próximo capítulo.

O controle supervisório permitiu tratar problemas de gerenciamento de tráfego, gerenciamento do nível de bateria dos AGVs, roteamento estático dos veículos e interação entre os AGVs e os pontos de coleta e distribuição de carga. No entanto, ressalta-se que a abordagem modular local é dependente do teste de não conflito dos supervisores modulares locais para ser aplicada e o teste se torna impraticável para modelos muito complexos – *layout* de percurso extenso e elevado número de AGVs –, sendo necessários meios alternativos para tratar este problema (Pena, et al., 2009) e (Flordal e Malik, 2006).

## Capítulo 5

# Implementação de supervisores em sistema de AGV

Este capítulo descreve a implementação para microcontrolador, em linguagem C, dos supervisores locais sintetizados para o sistema de AGV estudado no capítulo 4. A implementação dos supervisores locais é feita num único microcontrolador a partir das versões reduzidas destes. No entanto as rotinas operacionais estão distribuídas nos diversos microcontroladores do sistema. Utiliza-se uma estrutura genérica de implementação (Queiroz e Cury, 2002), na qual são preservadas as características originais da TCS. A implementação desta estrutura de controle em programadores lógicos controláveis pode ser vista em (Klinge, 2007), a qual desenvolve um processo automatizado de tradução das máquinas de estados dos supervisores em linguagem de texto estruturado. Por fim, este capítulo descreve o ambiente de emulação gráfica desenvolvido para testar o código dos supervisores e sobre as considerações finais da implementação do projeto.

### 5.1 Estrutura genérica de controle

O controle exercido pelos supervisores age diretamente sobre as plantas, que representam o sistema real de forma abstrata. A TCS supõe que os eventos gerados pelas plantas, quando não desabilitados, ocorrem espontaneamente sem a interferência de controle externo. No entanto, os eventos controláveis são, em geral, disparados por comandos enviados pelo sistema de controle. Como exemplo, para o AGV1 iniciar o deslocamento para a zona BC é necessário que seja enviado o comando *bllc*, ao invés de simplesmente desabilitá-lo. Também é necessário traduzir os eventos abstratos dos modelos em seqüências de operações sobre os atuadores e sensores.

Para manter as características originais da TCS, implementa-se, conforme proposto por (Queiroz e Cury, 2002), uma interface, ilustrada pela figura 5.1, entre os supervisores e o sistema real. A função desta interface é comandar o sistema real conforme os modelos abstratos dos

subsistemas da planta e traduzir os comandos destes modelos em sinais de entrada do sistema real, e as saída do sistema real em respostas para os modelos abstratos. O conjunto de geradores que representam os subsistemas da planta aparece no nível chamado sistema-produto, cuja função é enviar comandos para executar as seqüências operacionais referentes aos eventos não desabilitados e receber as respostas enviadas pelas seqüências operacionais. Sempre que o sistema-produto envia comandos ou recebe respostas, os supervisores são atualizados.

As seqüências operacionais correspondem ao conjunto de ações de baixo nível que devem ser executadas, segundo determinada lógica, para realizar a ordem do comando enviado pelo sistema-produto, ou seja, as seqüências operacionais representam a tradução dos comandos lógicos do sistema-produto em sinais de entrada para o sistema real.

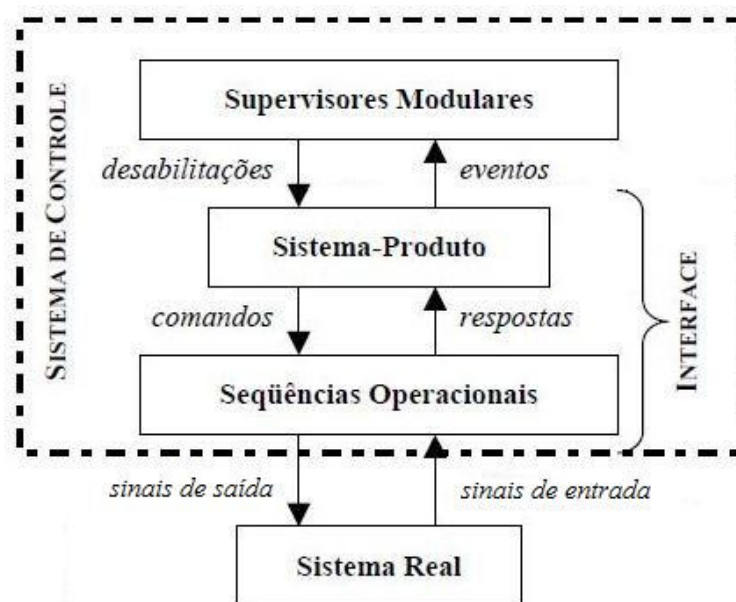


Figura 5.1. Estrutura do sistema de controle (Queiroz, 2000)

## 5.2 Estrutura de controle aplicada ao sistema de AGV

Para implementar a estrutura genérica de controle nos microcontroladores do sistema de AGV estudado, utilizou-se a linguagem de programação C. Os supervisores modulares locais, o sistema-produto e as seqüências operacionais das estações de carga e descarga foram implementados no microcontrolador central. Já as seqüências operacionais dos AGVs estão embarcadas nos microcontroladores dos mesmos. Detalhes da implementação das seqüências operacionais fogem do objetivo deste trabalho.

Cada nível da estrutura é expresso por um conjunto de funções, e a interação entre estes níveis se dá através do aninhamento destas funções. O pseudocódigo da implementação desta estrutura de controle é organizado segundo a figura 5.2. Conforme é visto na função *main()*, o controlador começa chamando a função *atualizaDesabilitações()*, que define o conjunto de eventos desabilitados para os estados iniciais dos supervisores. Os estados iniciais dos supervisores são inicializados na declaração das variáveis que os representam. Posteriormente o programa entra num *loop* infinito que executa as funções *executaEventosControláveis()* e *atualizaSeqüênciasOperacionais()*.



Figura 5.2. Estrutura do pseudo-código da implementação

A função *atualizaSeqüênciasOperacionais()* constitui o nível mais baixo na hierarquia de controle. Sabe-se que as ações de controle dos supervisores são reações aos sinais enviados pelo sistema real, os quais devem ser constantemente acompanhados. A função *atualizaSeqüênciasOperacionais()* monitora a ocorrência destes sinais e responde ao sistema real com sinais de baixo nível de acordo com os comandos recebidos do sistema-produto. Quando ocorre um sinal de resposta do sistema real, a função *atualizaSeqüênciasOperacionais()* aciona o nível imediatamente superior – sistema-produto – a partir da função *recebeRespostas(resposta)*, a qual recebe como argumento a resposta representativa da ação ocorrida no sistema real.

A função *executaEventosControláveis()*, juntamente com a função *recebeRespostas(resposta)*, constituem o nível do sistema-produto. Segundo a TCS, os eventos controláveis são executados quando não estiverem desabilitados. Para isso, o sistema-produto deve sempre verificar se há algum evento controlável que deva ser disparado, o que é feito pela função *executaEventosControláveis()* no corpo da função *main()*. Quando há dois ou mais eventos não desabilitados, implanta-se uma heurística de prioridade na execução destes eventos. No sistema de AGV há momentos em que os eventos  $c_{<AGV>ld}$  e  $c_{<AGV>lf}$  estão habilitados num mesmo instante. Para cada veículo, optou-se por executar estes eventos alternadamente quando ambos estão habilitados. Fisicamente isto faz com que os veículos se desloquem alternando entre o menor e maior laço do percurso.

A função *recebeResposta(resposta)* é responsável por tratar as respostas recebida do nível das seqüências operacionais. Quando o sistema-produto dispara um evento controlável ou recebe sinalizações das seqüências operacionais, os estados das plantas do sistema-produto são atualizados e o evento ocorrido é repassado ao nível dos supervisores modulares locais através da função *atualizaSupervisores(evento)*.

A função *atualizaSupervisores(evento)* e *atualizaDesabilitações()* compõem o nível dos supervisores modulares locais. Os estados dos supervisores são atualizados ao ser chamada a função *atualizaSupervisores(evento)*, a qual atualiza a lista de eventos controláveis que devem ser desabilitados nos estados atuais através da função *atualizaDesabilitações()*. Quando ocorre uma sinalização do sistema real, somente devem ser atualizados os supervisores cujos estados sejam alterados por aquele evento.

Para cada supervisor local obtido há uma função *atualizaSupervisor<sup>(evento)*, sendo  $<sup>$  o índice do supervisor local. Nestas



funções, os supervisores são implementados respeitando o modelo de suas respectivas máquinas de estados. No entanto, como os eventos em auto-laços não ocasionam mudança de estado nos supervisores e consequentemente não alteram o conjunto de eventos desabilitados, os auto-laços podem ser omitidos no código dos supervisores. Alternativamente a esta forma de implementação, Teixeira *et al.* (2006) apresentam uma forma que dispõe o supervisor através de grupos de dados em uma tabela. A implementação deste autor também é feita em microcontroladores, porém na linguagem *assembly*.

Nos geradores dos supervisores modulares locais, os eventos desabilitados em cada estado são representados por auto-laços tracejados. Na figura 5.3 observa-se o supervisor  $S_{redMAN2}$  e o código em linguagem C associado ao mesmo. A função *atualizaSupervisorMan2()* somente é chamada quando ocorre algum evento que causaria mudanças de estado do supervisor  $S_{redMAN2}$ . Ao ser acionada, a função *atualizaSupervisorMan2()* executa a mudança de estados e chama a função *atualizaDesabilitacoes()* para alterar a lista de eventos desabilitados devido à esta mudança de estado.

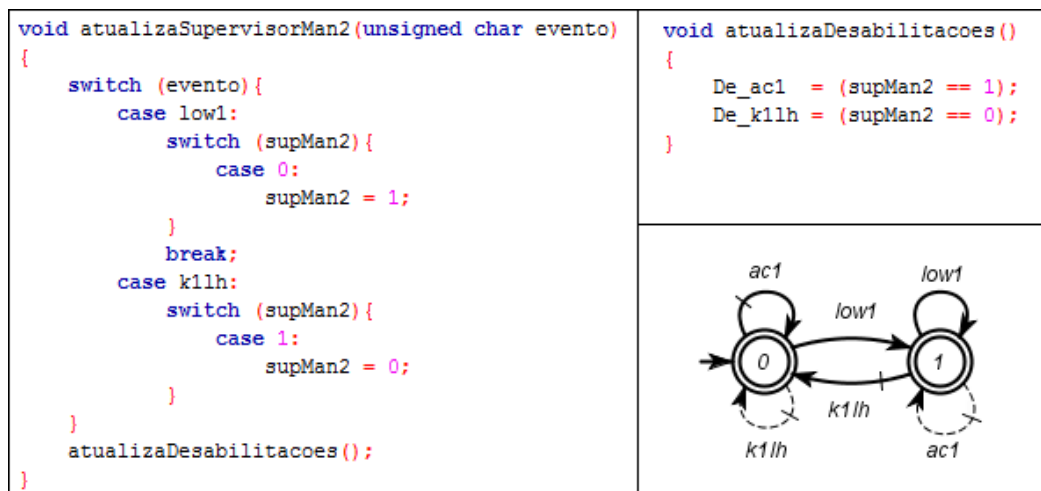


Figura 5.3. Implementação do supervisor  $S_{redMAN2}$  em C

No nível do sistema-produto, os modelos das plantas são implementados do mesmo modo como é feito para os supervisores. Cada subsistema da planta possui as funções *executaControláveis<subsistema da planta>()* e *recebeResposta<subsistema da planta>()*, sendo *<subsistema da planta>* o índice referente ao respectivo subsistema da planta. A figura 5.4 ilustra a implementação do gerador  $G_{CARGA}$ . Os eventos controláveis da planta são tratados por *executaControlaveisCarga()* e os não controláveis por *recebeRespostaCarga()*. Uma mudança no estado da planta não necessariamente ocasiona alteração nos estados dos supervi-

sores. Assim, somente são chamadas as funções de atualização dos supervisores se o evento também produzir mudanças nos mesmos.

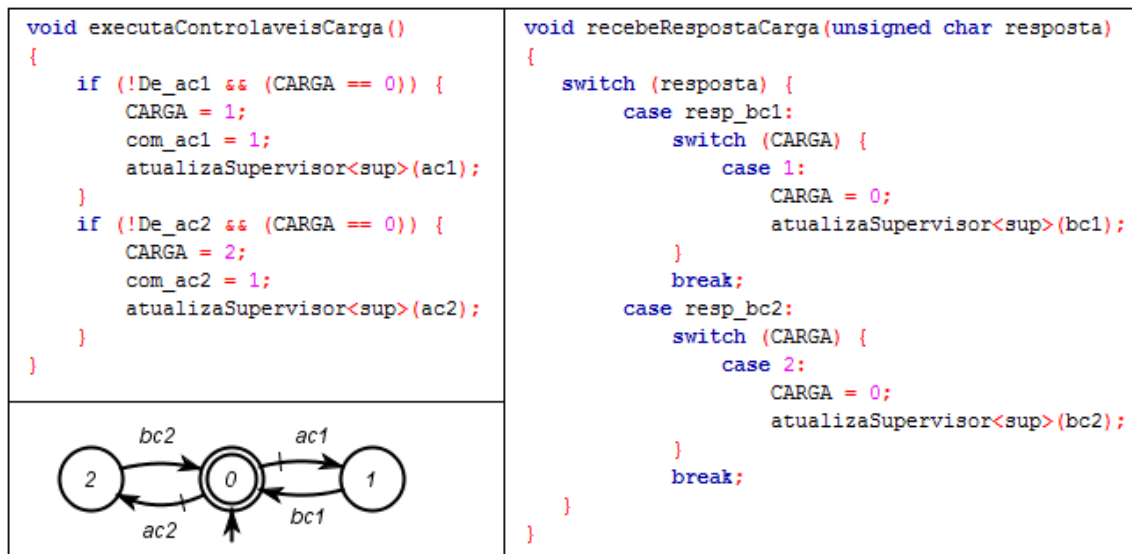


Figura 5.4. Implementação do gerador  $G_{CARGA}$  no nível sistema-produto

Conforme visto na seção 4.2, preferiu-se utilizar um modelo simplificado para os geradores dos AGVs. Como os eventos controláveis são executados automaticamente quando os mesmos estão habilitados, e no modelo omitem-se os eventos de chegada dos veículos em alguns pontos do sistema, pode ocorrer o disparo automático dos eventos controláveis subsequentes. Para resolver este impasse, sempre que os AGVs chegam num ímã não representado por um estado nos geradores dos AGV, é setada uma variável que indica a chegada dos AGVs nestes pontos. Na função *executaControlaveis()*, no nível do sistema-produto, antes de executar os eventos controláveis disparados nos estados que representam trechos, deve ser verificado se a variável que representa a chegada dos AGVs está setada. Desta forma, não há o risco de disparar um evento controlável sem a chegada do veículo no ponto onde o evento deve ser iniciado.

Na figura 5.5 é mostrado um trecho da implementação da função *executaControlaveisAGV1()*. No gerador do AGV1 não há evento de chegada que antecede o evento controlável *allb*. Desta forma, insere-se a variável *cheguei1* como condição para disparar o evento *allb*. O oposto ocorre para o evento *bllc*, que é antecedido pelo evento *bllh*, e não necessita da variável *cheguei1*.

```

void executaControlaveisAGV1(){
    if (!De_allb && (AGV1 == 5) && (chegueil == 1)) {
        AGV1 = 0;
        com_allb = 1;
        chegueil = 0;
        atualizaSupervisorAb(allb);
        atualizaSupervisorEja(allb);
        atualizaSupervisorCargab(allb);
    }

    if (!De_b1lc && (AGV1 == 1)) {
        AGV1 = 2;
        com_b1lc = 1;
        atualizaSupervisorAb(b1lc);
        atualizaSupervisorBc(b1lc);
        atualizaSupervisorCargab(b1lc);
    }

    ...
}

```

Figura 5.5. Trecho da implementação do gerador  $G_{AGV1}$  no nível sistema-produto

A implementação das seqüências operacionais para os eventos controláveis do gerador  $G_{CARGA}$  é mostrada na figura 5.6. As máquinas

```

void atualizaSoCarga()
{
    switch (SO_ac1) {
        case 0:
            if (com_ac1) {
                com_ac1 = 0;
                SO_ac1 = 1;
                r_carregou1 = 0;
                //inicia operação de carga
            }
            break;
        case 1:
            if (r_carregou1) {
                SO_ac1 = 0;
                r_carregou1 = 0;
                recebeRespostaCarga(resp_bc1);
            }
    }
    switch (SO_ac2) {
        case 0:
            if (com_ac2) {
                com_ac2 = 0;
                SO_ac2 = 1;
                r_carregou2 = 0;
                //inicia operação de carga
            }
            break;
        case 1:
            if (r_carregou2) {
                SO_ac2 = 0;
                r_carregou2 = 0;
                recebeRespostaCarga(resp_bc2);
            }
    }
}

```

Figura 5.6. Implementação das seqüências operacionais do gerador  $G_{CARGA}$

de estados das operações *carrega AGV1* e *carrega AGV2* são descritas pelas variáveis *SO\_ac1* e *SO\_ac2* respectivamente. Ao receber os comandos do sistema-produto, estas operações iniciam sua execução e a estrutura de controle espera pela sinalização do término das operações. Chegando os sinais do sistema real, a função *recebeRespostaCarga(resposta)* repassa ao sistema-produto a indicação da finalização da operação.

### 5.3 Ambiente de emulação do controlador

Para testar a implementação do código dos supervisores segundo a estrutura de controle apresentada, criou-se um ambiente gráfico para emulação do controlador. Este ambiente, visto na figura 5.7, apresenta o comportamento do sistema real de AGV de acordo com as ações de controle executadas pelos supervisores. No ambiente de emulação, as rotinas operacionais do sistema real são substituídas por um conjunto de funções que simulam estas rotinas reais. As sinalizações do sistema que não representam término de operação – sinalização de chegada de *pallet* e do nível crítico da bateria dos veículos – são disparadas pelo usuário. As demais sinalizações do sistema são disparadas automaticamente conforme o sistema evolui.

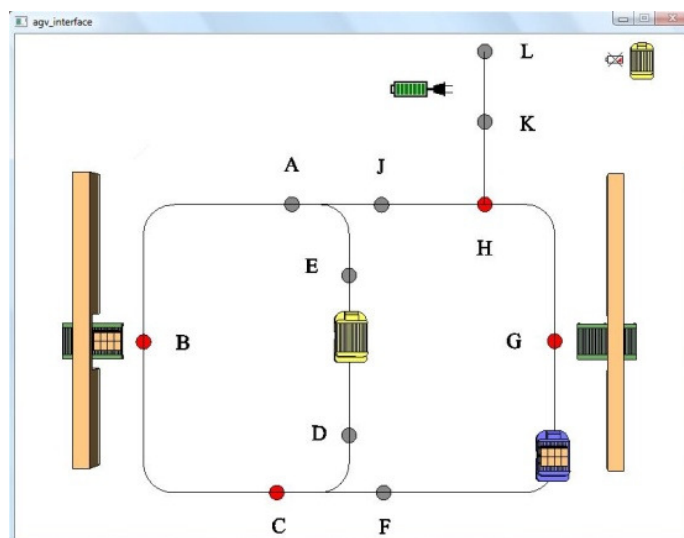


Figura 5.7. Ambiente de emulação do controlador

O emulador recebe os sinais de resposta do sistema e, após serem processados pelo controlador, retorna animações gráficas correspondentes. Para desenvolver este ambiente de emulação, utilizou-se a biblioteca *allegro* para programação de jogos em linguagem C/C++ (ALLEGRO, 2009), que é distribuída gratuitamente e suporta as plataformas *Unix*,

*Windows, MacOS X e Haiku/BeOS*. O executável do ambiente de emulação desenvolvido e o respectivo código fonte podem ser encontrados no endereço <http://www.das.ufsc.br/~ygarcia>.

A principal motivação para o desenvolvimento desta interface foi a possibilidade de tornar o processo de testes do código dos supervisores independente dos componentes de *hardware* do sistema e das rotinas operacionais de baixo nível. A emulação gráfica facilita a visualização da aplicação das leis de controle especificadas, de modo que a identificação dos erros de digitação do código se torna mais eficiente do que a depuração convencional.

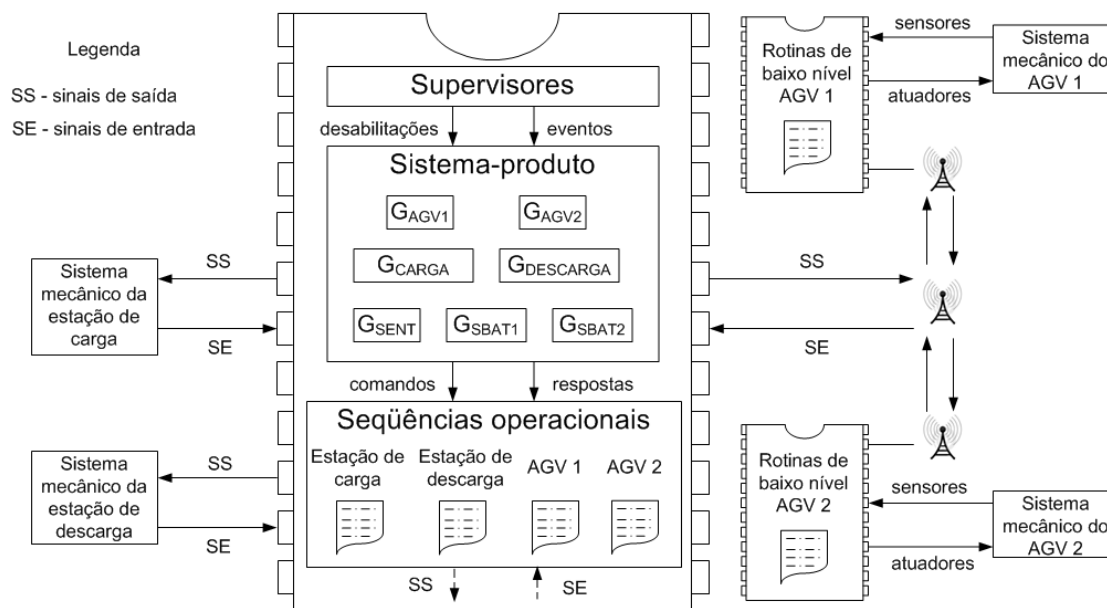
A estrutura do código do ambiente desenvolvido pode ser aproveitada para criar um ambiente de gerenciamento remoto para o sistema de AGV, o qual fornece uma visão global do sistema em funcionamento. Para isto, devem-se coletar informações dos estados das plantas e dos supervisores que traduzem um significado físico para o sistema real.

#### **5.4 Implementação da bancada experimental do sistema de AGV**

A implementação da bancada experimental do sistema de AGV contou com duas frentes de trabalho: implementação dos supervisores e implementação das seqüências operacionais do sistema. A primeira foi desenvolvida pelo autor deste documento e consiste no conteúdo descrito nas seções anteriores deste capítulo. Já a segunda frente e a integração destas implementações em microcontroladores foi realizada por um aluno de graduação do curso de engenharia de controle e automação da UFSC.

A figura 5.9 mostra a arquitetura de implementação do sistema de AGV. Os supervisores modulares locais e o sistema-produto foram implementados num mesmo microcontrolador, juntamente com o nível das seqüências operacionais. Já as rotinas de operacionais de baixo nível dos AGVs estão presentes nos microcontroladores de cada um deles. Desta forma, quando o nível das seqüências operacionais deve iniciar uma rotina dos AGVs, o nível das seqüências operacionais envia a respectiva ordem para o microcontrolador do AGV correspondente via radiofrequência.

Para aplicar o código dos supervisores no microcontrolador 8051, utilizou-se o ambiente integrado de desenvolvimento Keil™ (KEIL, 2010). A linguagem C para microcontroladores, também conhecida como *embedded C*, apresenta características próprias para suportar a capacidade de memória reduzida dos microcontroladores (Parab, et al.,



ntação do sistema de AGV

2007). O ambiente Keil™ permite utilizar a variável *bit* que, como sugere o nome, ocupa 1 bit de memória. Muitas variáveis do código dos supervisores não necessitam mais do que 1 bit de memória e sofreram esta alteração na implementação contida no microcontrolador 8051. A capacidade de memória dos microcontroladores 8051 foi suficiente para suportar a implementação do código.

Apesar dos supervisores e das rotinas operacionais terem sido implementados sem grandes problemas, o sistema de AGV apresenta perdas de mensagem na comunicação por radiofrequência. Isto impossibilitou o funcionamento adequado da bancada experimental até o final da escrita deste documento.

## 5.5 Conclusão do capítulo

O sucesso alcançado na implementação do sistema de controle da bancada experimental dos AGV, conforme visto na emulação, demonstra o poder da TCS para solucionar problemas reais de controle. A estrutura genérica utilizada na implementação, além de preservar as características originais da TCS, quando associada com a linguagem C permite a criação de um código bem organizado.

Na bancada experimental do sistema de AGV, a estrutura de controle é implementada em microcontroladores 8051, os quais apresentam capacidade de memória reduzida. Durante a implementação do código dos supervisores, preocupou-se em gerar um código que se encaixasse

nas limitações de memória dos microcontroladores. No entanto, em sistemas de AGV reais este cuidado pode ser relaxado, visto que o controle supervísório é normalmente executado em um *hardware* dedicado com grande quantidade de memória.

Percebeu-se que a implementação do código dos supervisores apresenta padronização em sua estrutura, o que sugere a possibilidade de desenvolvimento de ferramenta para a geração automática do código em linguagem C para os supervisores.





## Capítulo 6

### Conclusão

Este trabalho apresentou a aplicação da TCS para controlar um sistema de AGV de pequeno porte. Os supervisores que gerenciam o sistema evitam colisões entre os veículos e *deadlocks*, especificam regras de roteamento para os veículos e lidam com a interação entre as estações de transferência de materiais e os AGVs. A implementação dos supervisores é feita segundo uma estrutura de controle que conserva as características originais da TCS, a qual considera que os eventos ocorrem espontaneamente na planta.

A estrutura de controle proposta pela TCS separa o sistema em malha aberta do sistema de controle. Este benefício possibilita a síntese de diferentes soluções de controle para a mesma planta. A síntese dos supervisores é feita de modo automático e concentra-se em achar a solução minimamente restritiva e não bloqueante para as especificações definidas para a planta. Esta propriedade da síntese oferece aos AGVs flexibilidade operacional, permitindo a execução de todos os eventos que não se oponham às especificações do sistema e não ocasione bloqueio (*deadlock* ou *livelock*).

Na solução adotada para o sistema, segundo a abordagem modular local da TCS, são exploradas as modularidades das plantas e especificações. Um benefício que isto traz é que, ao se introduzir ou alterar uma especificação, não é necessário realizar a síntese para todo o conjunto de especificações, mas apenas para a que foi alterada ou introduzida. Isto facilitaria o trabalho de expansão do sistema de AGV, seja para adicionar novos veículos ou trechos de percurso. No entanto, o teste de não conflito deve ser feito entre todos os supervisores modulares locais. Este é um ponto que ainda pesa na aplicação da abordagem modular local da TCS, pois o bloqueio é um problema global do sistema, o que torna necessário a realização da composição síncrona de todos os supervisores do sistema para verificar o não bloqueio.

A estrutura de controle hierarquizada usada na implementação permitiu a elaboração sistemática do código do controlador e ao final chegou-se a um código limpo e bem organizado. Um ponto importante para o sucesso da implementação do código foi o uso de um ambiente

gráfico para a emulação do controlador, onde foi possível visualizar o controlador atuando numa representação gráfica do sistema real. Este ambiente serviu não só para a identificação dos erros de implementação do código que comprometeriam o desempenho do controlador, mas também para validar o desempenho do controlador proposto, visto que não há alterações da lógica estrutural do mesmo para a implementação no sistema real.

O resultado da aplicação da abordagem modular local da TCS no controle do sistema de AGV serve como incentivo para a propagação desta teoria na resolução de problemas de controle de SEDs de modo geral. É certo que as dificuldades do teste de não conflito limitam a aplicação desta abordagem, não sendo possível garantir o não bloqueio para sistemas de grande porte.

Como contribuição deste trabalho junto à comunidade científica, foram escritos dois artigos. O primeiro com o título “*Formal Synthesis, Simulation and Automatic Code Generation of Supervisory Control for a Manufacturing Cell*”, publicado no proceedings of the 20th International Congress of Mechanical Engineering (COBEM’09). O segundo com o título “*Síntese e implementação de controle supervisorio modular local para um sistema de AGV*”, aceito no XVII Congresso Brasileiro de Automática (CBA’10).

O trabalho iniciado por este autor continua com a busca de soluções para os problemas de comunicação e também para a expansão do sistema de AGV. Como perspectivas de trabalhos futuros ressaltam-se:

- Solução dos problemas de comunicação de sistemas de AGV através de redes de sensores sem fio (Akyildiz, et al., 2002);
- Introdução de novas especificações de controle relacionadas à movimentação de AGVs ociosos e à economia da bateria;
- Expansão do número de veículos e trechos de deslocamento da bancada estudada, o qual requer a investigação de técnicas que vissem contornar as limitações da abordagem modular local impostas pelo teste de não conflito, como Pena *et al.*(2009) e Flordal e Malik (2006) têm feito.
- Desenvolvimento de ambiente de gerenciamento remoto SCADA (Daneels e Salter, 1999) para o sistema de AGV.
- Desenvolvimento de ferramenta para traduzir os supervisores em linguagem C, de forma que todo o processo de síntese e implementação seja automatizado. Isto viabilizaria a geração do controlador a

partir do *layout* de percurso do sistema, da mesma forma feita por (Hsieh e Kang, 1998).

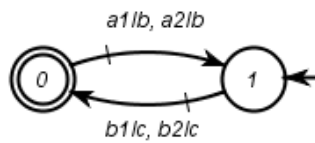


# APÊNDICE

## Supervisores modulares locais reduzidos para o sistema de AGV:

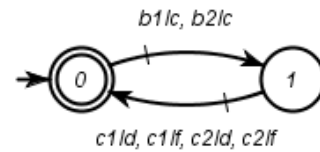
### A) Impedimento de colisões:

$S_{redCOL1}$ :



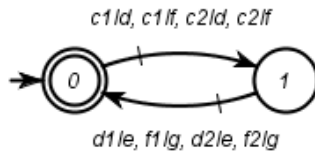
0:  $b1lc, b2lc$   
1:  $a1lb, a2lb$

$S_{redCOL2}$ :



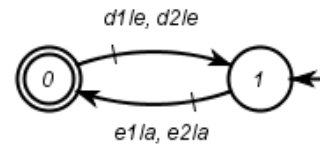
0:  $c1ld, c1lf, c2ld, c2lf$   
1:  $b1lc, b2lc$

$S_{redCOL3}$ :



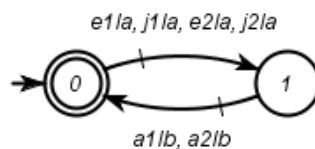
0:  $d1le, f1lg, d2le, f2lg$   
1:  $c1ld, c1lf, c2ld, c2lf$

$S_{redCOL4}$ :



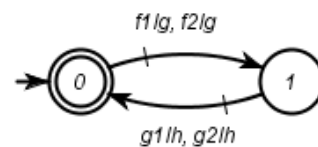
0:  $e1la, e2la$   
1:  $d1le, d2le$

$S_{redCOL5}$ :

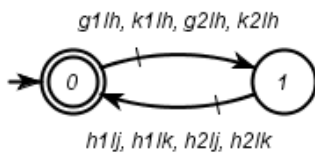


0:  $a1lb, a2lb$   
1:  $e1la, j1la, e2la, j2la$

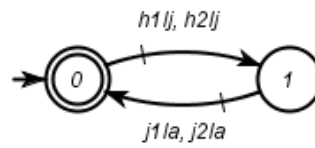
$S_{redCOL6}$ :



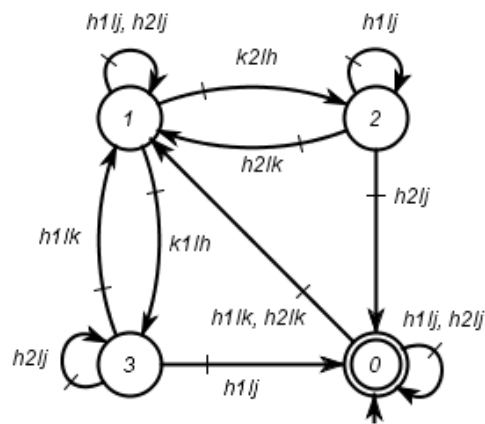
0:  $g1lh, g2lh$   
1:  $f1lg, f2lg$

$S_{redCOL7}$ :

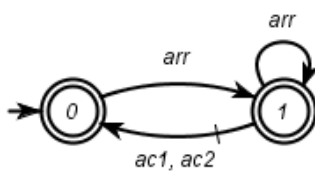
0:  $h1lj, h1lk, h2lj, h2lk$   
 1:  $g1lh, k1lh, g2lh, k2lh$

 $S_{redCOL8}$ :

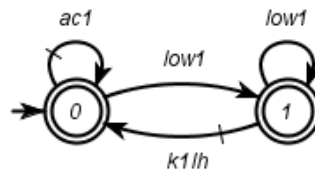
0:  $j1la, j2la$   
 1:  $h1lj, h2lj$

 $S_{redCOL9}$ :

0:  $k1lh, k2lh$   
 1:  $h1lk, h2lk$   
 2:  $h1lk, k1lh, k2lh$   
 3:  $h2lk, k1lh, k2lh$

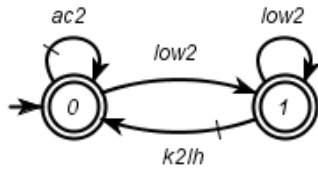
**B) Manipulação de pallets:** $S_{redMAN1}$ :

0:  $ac1, ac2$

 $S_{redMAN2}$ :

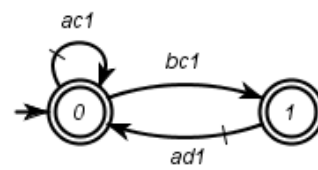
0:  $k1lh$   
 1:  $ac1$

$S_{redMAN3}$ :



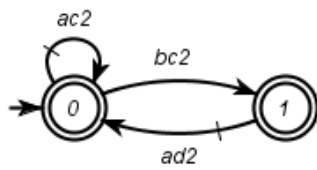
0:  $k2lh$   
1:  $ac2$

$S_{redMAN4}$ :



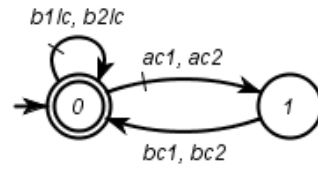
0:  $ad1$   
1:  $bc1, ac1$

$S_{redMAN5}$ :



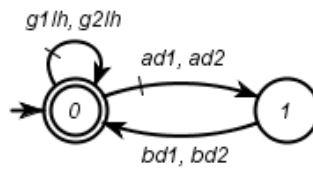
0:  $ad2$   
1:  $ac2, bc2$

$S_{redMAN6}$ :



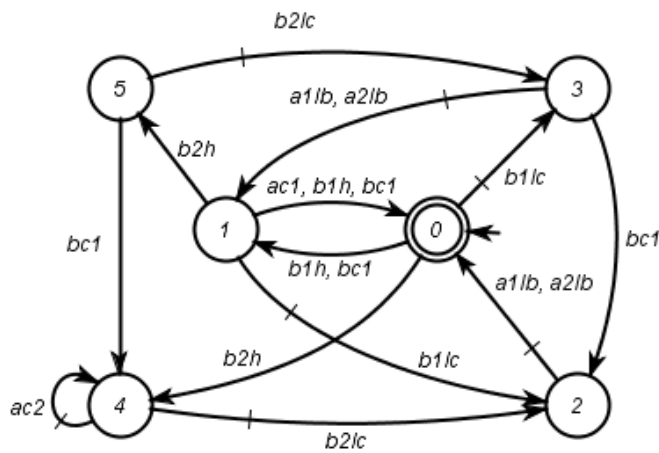
0:  $bc1, bc2$   
1:  $ac1, ac2, b1lc, b2lc$

$S_{redMAN7}$ :



0:  $bd1, bd2$   
1:  $ad1, ad2, g1lh, g2lh$

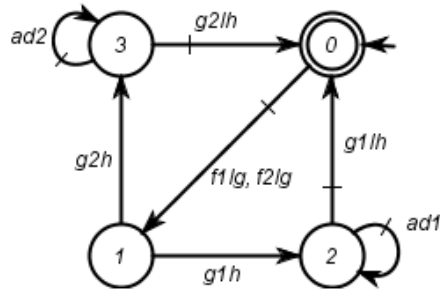
$S_{redMAN8}$ :



0:  $a1lb, a2lb, ac1, ac2, b2h, b2lc$   
1:  $a1lb, a2lb, ac2, b2lc$   
2:  $ac1, ac2, b1h, b1lc, b2h, b2lc, bc1$

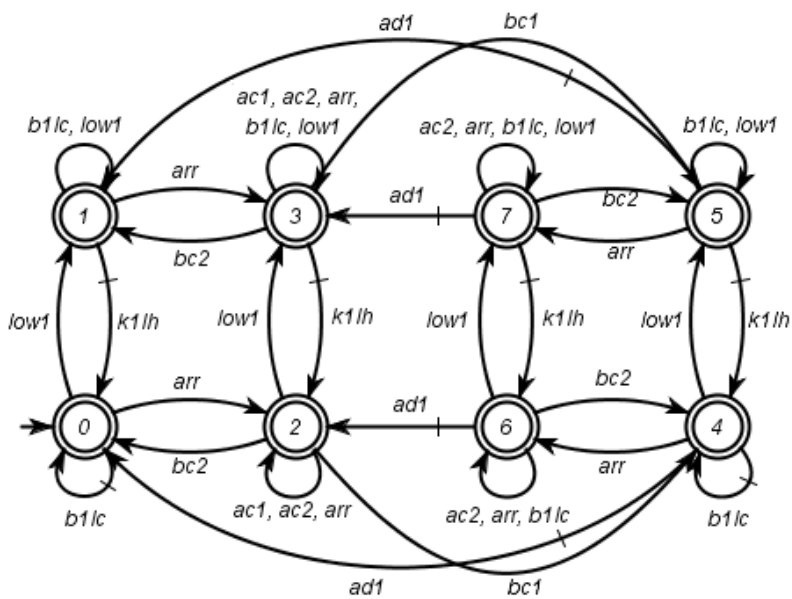
3:  $ac1, ac2, b1h, b1lc, b2h, b2lc$   
 4:  $a1lb, a2lb, ac1, b1h, b1lc, b2h, bc1$   
 5:  $a1lb, a2lb, ac1, ac2, b1h, b1lc, b2h$

$S_{redMAN9}$ :



0:  $ad1, ad2, g1h, g2h, g1lh, g2lh$   
 1:  $ad1, ad2, f1lg, f2lg, g1lh, g2lh$   
 2:  $ad2, f1lg, f2lg, g1h, g2h, g2lh$   
 3:  $ad1, f1lg, f2lg, g1h, g2h, g1lh$

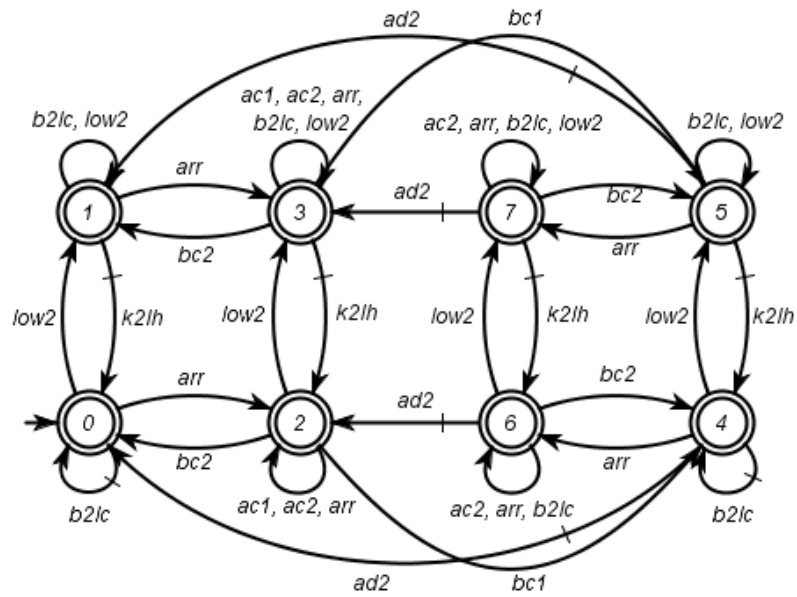
$S_{redMAN10}$ :



0:  $ac1, ac2, ad1, bc1, bc2, k1lh$   
 1:  $ac1, ac2, bc1, bc2$   
 2:  $ad1, b1lc, k1lh$   
 3:  $ad1$   
 4:  $ac1, ac2, bc1, bc2, k1lh$   
 5:  $ac1, ac2, bc1, bc2$   
 6:  $ac1, bc1, k1lh$   
 7:  $ac1, bc1$



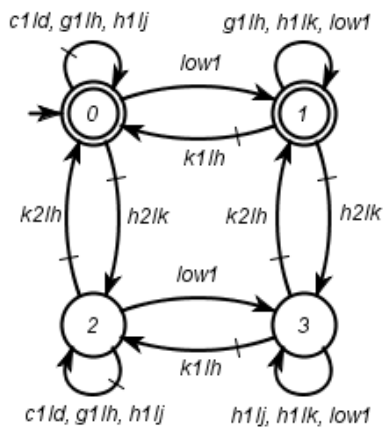
$S_{redMAN11}$ :



- 0:  $ac1, ac2, ad2, bc1, bc2, k2lh$
- 1:  $ac1, ac2, bc1, bc2$
- 2:  $ad1, b2lc, k2lh$
- 3:  $ad2$
- 4:  $ac1, ac2, bc1, bc2, k2lh$
- 5:  $ac1, ac2, bc1, bc2$
- 6:  $ac1, bc1, k2lh$
- 7:  $ac1, bc1$

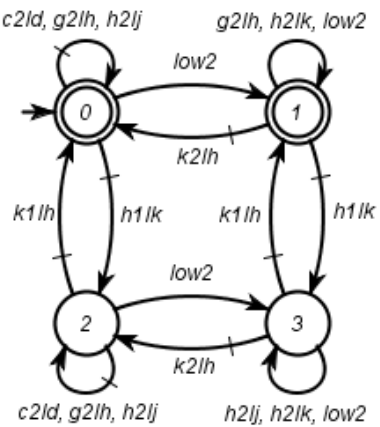
**C) Roteamento dos AGVs:**

$S_{redROT1}$ :



- 0:  $h1lk, k1lh, k2lh$
- 1:  $c1ld, h1lj, k2lh$

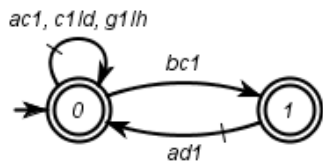
$S_{redROT2}$ :



- 0:  $h2lk, k1lh, k2lh$
- 1:  $c2ld, h2lj, k1lh$

2:  $h1lk, h2lk, k1lh$   
 3:  $c1ld, g1lh, h2lk$

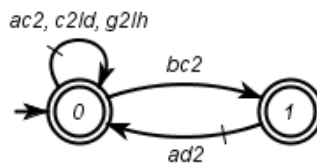
$S_{redROT3}$ :



0:  $ad1$   
 1:  $ac1, bc1, c1ld, g1lh$

2:  $h2lk, h1lk, k2lh$   
 3:  $c2ld, g2lh, h1lk$

$S_{redROT4}$ :



0:  $ad2$   
 1:  $ac2, bc2, c2ld, g2lh$

# REFERÊNCIAS BIBLIOGRÁFICAS

- AKYILDIZ, I., SU, W., SANKARASUBRAMANIAM, Y., e CAYIRCI, E.; 2002, "A survey on sensor networks", IEEE communications magazine, Vol. 40, No. 8, pp. 102-114.
- ALAVUDEEN, A. e VENKATESHWARAN, N.; 2008. "Computer integrated manufacturing". 1st Prentice Hall of India.
- ALLEGRO; 2009, "A game programming library", <http://alleg.sourceforge.net/>, 14/11/2009.
- AUTOMOTION; 2009, "Automated Guided Carts", <http://www.automotion.org/TC-900WP.asp>, 17/05/2009.
- CASSANDRAS, C.G. e LAFORTUNE, S.; 1999. "Introduction to Discrete Event Systems". Kluwer Academic Publishers, USA.
- CUBBERLY, W. e BAKERJIAN, R.; 1989. "DESK EDITION: Tool and manufacturing engineers handbook". Society of manufacturing engineers.
- DANEELS, A. e SALTER, W.; 1999, "What is SCADA?", International Conference on Accelerator and Large Experimental Physics Control Systems, Trieste, pp. 339-343.
- ELETRIC80; 2009, "Laser Guided Vehicles", <http://www.eletric80.com/products-applications/our-products/lgv/index.aspx>, 27/04/2009.
- EVERS, J.J.M. e KOPPERS, S.A.J.; 1996, "Automated guided vehicle traffic control at a container terminal", Transportation Research Part A, Vol. 30, No. 1, pp. 21-34.
- FENG, L. e WONHAM, W.; 2006, "TCT: A computation tool for supervisory control synthesis", Eighth International Workshop on Discrete Event Systems, Ann Arbor, pp. 388–389.
- FLORDAL, H. e MALIK, R.; 2006, "Nonblocking verification using conflict equivalence", Proceedings of the 8th International Workshop on Discrete Event Systems, Ann Arbor, USA, pp. 100 - 106.
- FROG; 2010, "General Technology Description AGV-systems", <http://www.frog.nl>, 31/03/2010.
- GROOVER, M.P.; 2007. "Automation, production systems, and computer-integrated manufacturing". Prentice Hall.
- GÜNTHER, H. e KIM, K.; 2005. "Container terminals and automated transport systems: logistics control issues and quantitative decision support". Springer.
- HKSYSTEMS; 2009, "Why are AGVs the best solution?", <http://www.hksystems.com/agv/index.cfm>, 12/10/2009.

- HOPCROFT, J.E. e ULLMAN, J.D.; 1979. "Introduction to Automata Theory, Languages and Computation". Addison-Wesley, USA.
- HSIEH, S. e KANG, M.Y.; 1998, "Developing AGVS petri net control models from flowpath nets", Journal of Manufacturing Systems, Vol. 17, No. 4, pp. 237-250.
- IEC; 1993. "International Electrotechnical Commission. IEC International Standard 1131-3 Programmable Controllers, Part 3: Programming Languages."
- JBT; 2009, "Automatic Guided Vehicle Systems", <http://www.jbtc-agv.com/solutions/products.aspx>, 24/04/2009.
- JBW; 2009, "John Deere Plant Case Study", <http://www.jerviswebb.com/pdfs.aspx>, 17/05/2009.
- KATEVAS, N.; 2001. "Mobile robotics in healthcare". IOS Press
- KEIL; 2010, "Embedded Development Tools", <http://www.keil.com/>, 21/01/2010.
- KENT, A. e WILLIAMS, J.; 1990. "Encyclopedia of Microcomputers: Volume 7-Evolution of Computerized Maintenance Management to Generation of Random Numbers". CRC.
- KLINGE, S.; 2007. "Supervisory control of a manufacturing cell: modeling and implementation". (Minor Thesis) - Fakultät für Elektrotechnik und Informationstechnik, Otto-von-Guericke-Universität Magdeburg.
- LE-ANH, T. e DE KOSTER, M.; 2006, "A review of design and control of automated guided vehicle systems", European Journal of Operational Research, Vol. 171, No. 1, pp. 1-23.
- MEYERS, F. e STEPHENS, M.; 2002. "Manufacturing facilities design and material handling". 2nd Prentice Hall.
- MITCHELL, P.E.; 1998. "Tool and manufacturing engineers handbook: material and part handling in manufacturing". 4th Society of manufacturing engineers.
- MOORTHY, L.; 2003, "Cyclic deadlock prediction and avoidance for zone-controlled AGV system", International Journal of Production Economics, Vol. 83, No. 3, pp. 309-324.
- MULCAHY, D.E.; 1999. "Materials Handling Handbook". 1st McGraw-Hill Professional.
- PARAB, J., SHELAK, V., KAMAT, R., e NAIK, G.; 2007. "Exploring C for microcontrollers". Springer.
- PENA, P., CURY, J., e LAFORTUNE, S.; 2009, "Verification of Nonconflict of Supervisors Using Abstractions", IEEE Transactions on Automatic Control, Vol. 54, No. 12, p. 2803.
- QUEIROZ, M.H.; 2000. "Controle Supervisório Modular de Sistemas de Grande Porte". Florianópolis. Dissertação (Mestrado em

- Engenharia Elétrica) - Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina.
- QUEIROZ, M.H.; 2004. "Controle supervisorio modular e multitarefa de sistemas compostos". Florianópolis. Tese (Doutorado em Engenharia Elétrica) - Programa de Pós-graduação em Engenharia Elétrica, Universidade Federal de Santa Catarina.
- QUEIROZ, M.H. e CURY, J.E.R.; 2000, "Modular supervisory control of large scale discrete event systems", Proceedings of the 5th International Workshop on Discrete Event Systems, Ghent, Belgium, pp. 103 - 110.
- QUEIROZ, M.H. e CURY, J.E.R.; 2002, "Synthesis and implementation of local modular supervisory control for a manufacturing cell", Sixth International Workshop on Discrete Event Systems, Zaragoza, pp. 377-382.
- RAMADGE, P.J.G. e WONHAM, W.M.; 1989, "The control of discrete event systems", Proceedings of the IEEE, Vol. 77, No. 1, pp. 81-98.
- REVELIOTIS, S.; 2000, "Conflict resolution in AGV systems", IIE Transactions, Vol. 32, No. 7, pp. 647-659.
- SCHULZE, L., BEHLING, S., e BUHRS, S.; 2008, "Automated Guided Vehicle Systems: a Driver for Increased Business Performance", Proceedings of the International MultiConference of Engineers and Computer Scientists, Vol. 2.
- SILVA, Y.G. e QUEIROZ, M.H.; 2009, "Formal synthesis, simulation and automatic code generation of supervisory control for a manufacturing cell", Proceedings of the 20th International Congress of Mechanical Engineering, Gramado, Brazil.
- SILVA, Y.G., QUEIROZ, M.H., e CURY, J.E.R.; 2010, "Síntese e implementação de controle supervisorio modular local para um sistema de AGV", XVII Congresso Brasileiro de Automática (CBA'10), Bonito, Brazil.
- SU, R. e WONHAM, W.M.; 2004, "Supervisor reduction for discrete-event systems", Discrete Event Dynamic Systems, Vol. 14, No. 1, pp. 31-53.
- TEIXEIRA, C.A., LEAL, A.B., e SOUSA, A.H.; 2006, "Implementação de supervisores em microcontroladores: uma abordagem baseada na teoria de controle de sistemas a eventos discretos", XVI Congresso Brasileiro de Automática (CBA'06), pp. 2772 - 2777.
- VIS, I.F.A.; 2006, "Survey of research in the design and control of automated guided vehicle systems", European Journal of Operational Research, Vol. 170, No. 3, pp. 677-709.

- WONHAM, W.M.; 2008, "Notes on control of discrete-event systems", Department of Electrical and Computer Engineering, University of Toronto.
- YE, R., HSU, W., e VEE, V.; 2000, "Distributed routing and simulation of automated guided vehicles", Proceedings of TENCON 2000, Vol. 2, Malaysia, pp. 315-320.