

FREDERICO THEODORO ALVES E MELLO

**IMPLEMENTAÇÃO DISTRIBUÍDA DE
CONTROLE SUPERVISÓRIO PARA
SISTEMAS A EVENTOS DISCRETOS**

**FLORIANÓPOLIS
2006**

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

IMPLEMENTAÇÃO DISTRIBUÍDA DE
CONTROLE SUPERVISÓRIO PARA
SISTEMAS A EVENTOS DISCRETOS

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

FREDERICO THEODORO ALVES E MELLO

Florianópolis, fevereiro de 2006.

IMPLEMENTAÇÃO DISTRIBUÍDA DE CONTROLE SUPERVISÓRIO PARA SISTEMAS A EVENTOS DISCRETOS

Frederico Theodoro Alves e Mello

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

Prof. José Eduardo Ribeiro Cury, Dr.
Orientador

Prof. Alexandre Trofino Neto, Dr.
Coordenador do Programa de Pós-Graduação em Engenharia Elétrica

Banca Examinadora:

Prof. José Eduardo Ribeiro Cury, Dr.
Presidente

Prof. Eduardo Alves Portela Santos, Dr.

Prof. Max Hering de Queiroz, Dr.

Prof. Victor Juliano de Negri, Dr.

*No final tudo dá certo.
Se ainda não deu certo, é porque não chegou o final.*

AGRADECIMENTOS

Ao Prof. Cury, pelo empenho e dedicação na orientação deste trabalho e pela amizade durante este período em que trabalhamos juntos.

Aos colegas, professores e funcionários do LCMI pelo companheirismo, ensinamentos e disponibilidade.

Aos colegas do grupo de SEDs, em especial Patrícia, Tatiana e Rodrigo, que colaboraram para a criação de um ambiente de trabalho amistoso no qual a troca de conhecimentos sempre foi realizada de maneira saudável e enriquecedora.

Aos amigos e familiares que, distantes ou próximos, foram os principais incentivadores e fontes de energia para que mais esta etapa fosse vencida.

Aos meus pais, Fernando e Christina, e ao meu irmão Rafael pelo incentivo, suporte e apoio oferecidos incondicionalmente em todos os momentos.

A Deus, pela vida e por possibilitar mais esta conquista.

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

IMPLEMENTAÇÃO DISTRIBUÍDA DE CONTROLE SUPERVISÓRIO PARA SISTEMAS A EVENTOS DISCRETOS

Frederico Theodoro Alves e Mello

Fevereiro/2006

Orientador: Prof. José Eduardo Ribeiro Cury, Dr.

Área de Concentração: Automação e Sistemas

Palavras-chave: sistemas a eventos discretos, controle supervisório, célula flexível de manufatura, implementação distribuída

Número de Páginas: xiv + 107

Sistemas a Eventos Discretos têm sido fonte de muitos estudos nos últimos anos já que podem modelar um grande número de sistemas reais. No entanto, apesar das teorias que tratam de lógicas de controle para resolução dos problemas, como a Teoria de Controle Supervisório (TCS) de Ramadge e Wonham, ainda não foram feitos muitos avanços no sentido da implementação destes sistemas para processos considerados de médio/grande porte. Algumas extensões da TCS, como o Controle Modular Local, propõem novas técnicas que reduzem tanto os esforços computacionais gastos na síntese quanto a memória necessária para a implementação das estruturas do sistema de controle. Mesmo assim pode ser necessário distribuir a implementação deste sistema em mais de um dispositivo de controle para viabilizar sua utilização. Neste trabalho é estudada uma metodologia de distribuição do sistema de controle supervisório que abrange aspectos de comunicação e sincronização. Neste trabalho são levantadas questões práticas e propostas contribuições teóricas à teoria de distribuição utilizada. Esta metodologia é aplicada na resolução de um problema de controle real considerando-se três configuração de distribuição distintas, cada uma delas atendendo a uma gama de problemas específicos.

Abstract of Dissertation presented to UFSC as a partial fulfillment of the requirements for the degree of Master in Electrical Engineering.

DISTRIBUTED IMPLEMENTATION OF SUPERVISORY CONTROL FOR DISCRETE EVENT SYSTEMS

Frederico Theodoro Alves e Mello

February/2006

Advisor: Prof. José Eduardo Ribeiro Cury, PhD.

Area of Concentration: Automation and Systems

Key words: discrete event systems, supervisory control, manufacturing flexible cell, distributed implementation

Number of Pages: xiv + 107

Discrete Event Systems (DES) have been extensively studied due to their capability of modeling actual plants. Despite the significant advances in recent years, the theory hasn't reached the desired level regarding implementation in medium/large scale processes. Some extensions for the Ramadge and Wonham Supervisory Control of DES were proposed, such as the Local Modular Supervisory Control, that decreases both the computational effort spent in the synthesis and the memory needed on the implementation of control systems structures. Nevertheless, it may be required by the nature of the system to distribute the control structure implementation in more than one control device. A control system distribution methodology is studied in this work. This methodology is concerned with communications and synchronization aspects and is used to solve a real control problem considering three distinct distribution configurations. Each one of them is suitable to a group of specific problems. Practical and theoretical aspects are discussed in addition to the distribution theory used. The aim of this work is to show the possibility of applying the SCT and, specially, the Local Modular Control to solve a real problem.

Sumário

1	Introdução	1
1.1	Motivação e Objetivos	2
1.2	Estrutura	4
2	Controle Supervisório de Sistemas Compostos	5
2.1	Linguagens e Autômatos	6
2.1.1	Linguagens	6
2.1.2	Autômatos e Geradores	7
2.2	Sistemas Compostos	9
2.3	Controle Supervisório Monolítico	10
2.4	Controle Modular	13
2.4.1	Controle Modular Clássico	14
2.4.2	Controle Modular Local	16
2.5	Conclusão do Capítulo	20
3	Controle Supervisório de uma Célula Flexível	21
3.1	Descrição da CEBE	21
3.2	Projeto do Controle Supervisório para a CEBE	23
3.2.1	Representação do Comportamento Livre dos Subsistemas Físicos	24
3.2.2	Representação das Especificações Genéricas	24
3.2.3	Síntese de Supervisores Segundo Abordagem Modular Local	28
3.3	Conclusão do Capítulo	31

4	Modelo para Implementação do Controle Supervisório	32
4.1	Algumas Abordagens da Problemática de Implementação	32
4.2	Implementação do Controle Supervisório	35
4.2.1	Formalização do Modelo de Implementação	37
4.2.2	Implementação no Dispositivo de Controle	41
4.3	Aplicação do modelo à CEBE	45
4.3.1	Supervisores Modulares e Módulos do Sistema Produto	45
4.3.2	Seqüências Operacionais	49
4.4	Conclusão do Capítulo	50
5	Controle Supervisório Distribuído	51
5.1	Distribuição Vertical	52
5.1.1	Distribuição Vertical na CEBE	54
5.2	Distribuição Horizontal	58
5.2.1	Seqüência de Processamento Sincronizado	60
5.2.2	Autômatos Sincronizadores	63
5.2.3	Canais de Comunicação	65
5.2.4	Implementação do Sistema de Controle com Distribuição Horizontal	70
5.3	Distribuição Geral	70
5.4	Conclusão do Capítulo	72
6	Implementação Distribuída do Controle Supervisório na CEBE	74
6.1	Configurações de Hardware	75
6.1.1	Configuração Pré-Existente	75
6.1.2	Configuração Atual	76
6.1.3	PROFIBUS	77
6.2	Problemas de Distribuição	79
6.2.1	Implementação 1	81

6.2.2	Implementação 2	83
6.2.3	Implementação 3	86
6.3	Análise para Projeto de Sistemas Distribuídos	90
6.3.1	Critérios de Avaliação	90
6.3.2	Análise das Configurações de Implementação	92
6.3.3	Exemplo de Aplicação da Distribuição	95
6.4	Estrutura do Programa	96
6.5	Conclusão do Capítulo	100
7	Conclusões	101

Lista de Figuras

2.1	Esquema de funcionamento do controle monolítico	10
2.2	Exemplo: (a) M_1 ; (b) M_2	12
2.3	Exemplo: Esquemático do Sistema	12
2.4	Autômatos que geram as linguagens: (a) E_1 ; (b) E_2	12
2.5	Autômato que gera linguagem $Sup C(K, G)$	13
2.6	Supervisor monolítico reduzido	13
2.7	Esquema de funcionamento do controle modular	14
2.8	Autômatos que geram linguagens: (a) $Sup C(K_1, G)$; (b) $Sup C(K_2, G)$	16
2.9	Esquema de funcionamento do controle modular local [23]	16
2.10	RSP: (a) modelo dos subsistemas (b) RSP mais refinada	17
2.11	Constituição das plantas locais	18
2.12	Exemplo: (a) M_1 ; (b) M_2 (b) M_3	19
2.13	Exemplo: Esquemático do Sistema	19
2.14	Especificações Genéricas: (a) E_1 ; (b) E_2	19
2.15	Autômatos que geram linguagens: (a) $Sup C(E_{loc1}, G_{loc1})$; (b) $Sup C(E_{loc2}, G_{loc2})$	20
2.16	Supervisores Reduzidos: (a) S_{red1} ; (b) S_{red2}	20
3.1	Diagrama esquemático da CEBE	22
3.2	Supervisores modulares locais reduzidos: (a) $Sred_a$; (b) $Sred_{b1}$; (c) $Sred_{b2}$; (d) $Sred_{b3}$; (e) $Sred_{b4}$; (f) $Sred_{c1}$; (g) $Sred_{c2}$; (h) $Sred_{c3}$	30
4.1	Esquema de implementação do sistema de controle	36

4.2	Autômato que representa o comportamento livre da mesa giratória	39
4.3	Módulo do sistema produto da mesa giratória representado por máquina de Mealy	39
4.4	Estrutura de implementação de transição do supervisor	42
4.5	Estrutura de implementação de ativação de desabilitação de evento controlável	42
4.6	Estrutura de implementação de ativação de desabilitação de evolução de módulo do sistema produto	42
4.7	Estrutura de implementação de transição relacionada a uma resposta	43
4.8	Estrutura de implementação de transição relacionada a uma negação de desabilitação de evento controlável	43
4.9	Tela <i>Configurações</i> do CGLI [19]	46
4.10	Tela <i>Supervisores</i> do CGLI [19]	46
4.11	Tela <i>Símbolos</i> do CGLI [19]	47
4.12	Supervisor modular local reduzido $Sred_a$	49
4.13	Diagrama ladder do supervisor $Sred_a$	49
5.1	Sistema de controle supervísório com distribuição vertical	53
5.2	Estrutura do sistema de controle: (a) nó raiz; (b) nó folha [35]	55
5.3	Grafo de distribuição vertical da CEBE	56
5.4	Diagrama de blocos do programa das HBs [2]	57
5.5	Fluxograma de funcionamento do programa implementado no CLP	59
5.6	Grafo de distribuição horizontal que representa a seqüência de processamento sincronizado A[BC]DE	63
5.7	Autômatos Sincronizadores: (a) primeiro elemento da seqüência de processamento sincronizado; (b) demais elementos	64
5.8	Autômatos Sincronizadores segundo [35] : (a) primeiro elemento da seqüência de processamento sincronizado; (b) demais elementos	65
5.9	Supervisor S_a	67
5.10	Módulos do sistema produto: (a) g_0 ; (b) g_1	67
5.11	Grafo de comunicação	69

5.12 Grafo de distribuição horizontal do sistema do problema do Exemplo 5.7 segundo Vieira [35]	69
5.13 Implementação do sistema de controle para um nó do grafo de distribuição horizontal [35]	70
5.14 Representação da distribuição geral: (a) 1º nível; (b) 2º nível	72
5.15 Implementação do sistema de controle para um nó do grafo de distribuição horizontal com distribuição vertical das seqüências operacionais [35]	72
6.1 Configuração de <i>hardware</i> pré-existente	76
6.2 Configuração de <i>hardware</i> atual	77
6.3 Grafo de distribuição horizontal da Implementação 1	82
6.4 Grafo de comunicação para Implementação 1	82
6.5 Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 1	83
6.6 Diagrama Implementação 1	83
6.7 Grafo de distribuição horizontal da Implementação 2	84
6.8 Grafo de comunicação para Implementação 2	85
6.9 Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 2	86
6.10 Diagrama Implementação 2	86
6.11 Grafo de distribuição horizontal da Implementação 3	87
6.12 Grafo de comunicação para Implementação 3	88
6.13 Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 3	89
6.14 Diagrama Implementação 3	89
6.15 Representação do sistema da Situação 2	93
6.16 Representação do sistema da Situação 2 com barreira de comunicação	94
6.17 Representação do sistema da Situação 3	94
6.18 Representação do sistema de exemplo da aplicação da distribuição	95
6.19 Fluxograma do algoritmo que realiza a Distribuição Vertical nos CLPs S7-200	97

6.20	Estrutura básica do programa que realiza a Distribuição Horizontal nos CLPs S7-200	98
6.21	Lógica para sincronização	99
6.22	Estrutura básica do programa implementado nos CLPs S7-300	100

Lista de Tabelas

3.1	Gerador G_0 (mesa giratória)	24
3.2	Gerador G_1 (alimentador)	24
3.3	Gerador G_2 (enchedor)	24
3.4	Gerador G_3 (tampador)	25
3.5	Gerador G_4 (manipulador)	25
3.6	Gerador $Egen_a$	25
3.7	Gerador $Egen_{b1}$	26
3.8	Gerador $Egen_{b2}$	26
3.9	Gerador $Egen_{b3}$	26
3.10	Gerador $Egen_{b4}$	27
3.11	Gerador $Egen_{c1}$	27
3.12	Gerador $Egen_{c2}$	28
3.13	Gerador $Egen_{c3}$	28
3.14	Plantas locais	29
4.1	Lista de instruções gerada pelo CGLI (apenas as 6 redes iniciais)	47
4.1	Lista de instruções gerada pelo CGLI (apenas as 6 redes iniciais)	48
6.1	Alfabetos de eventos relevantes e desabilitações efetivas da CEBE	80
6.2	Avaliação das configurações para Situação 1	92
6.3	Avaliação das configurações para Situação 2	94
6.4	Avaliação das configurações para Situação 3	95

Capítulo 1

Introdução

Os processos produtivos industriais têm sofrido uma constante evolução nas últimas décadas. Observa-se uma tendência cada vez maior de se automatizar os processos na tentativa de se obter produtos a custos mais baixos e com maior qualidade. Com a automação, postos de trabalho que se caracterizam pela repetitividade de ações são substituídos pelo trabalho de máquinas, robôs e computadores. É essencial, portanto, que o ser humano crie condições para que estes dispositivos operem adequadamente, realizando exatamente as tarefas desejadas. O controle de processos surge assim como um conjunto de ações através das quais é possível manter o funcionamento de um sistema de acordo com um comportamento desejado.

Dentro do processo de evolução observado nos últimos anos, os sistemas de manufatura vêm se tornando cada vez mais complexos. Neste novo contexto, soluções informais e intuitivas do problema de controle não podem mais ser utilizadas, o que induziu ao estudo e desenvolvimento de técnicas de síntese de lógicas de controle [4]. As técnicas para a realização do controle de sistemas diferem de acordo com o tipo de sistema tratado. Os sistemas encontrados em aplicações práticas podem ser divididos em dois grandes grupos, de acordo com as características das variáveis que possuem.

Os sistemas a estado contínuo, nos quais as variáveis são de natureza contínua, podem ser representados por modelos matemáticos baseados em equações diferenciais e a diferenças. A Teoria de Controle Clássica é a responsável pelo desenvolvimento de técnicas para o controle deste tipo de sistema e seus resultados já vêm sendo amplamente aplicados na indústria.

Os sistemas a eventos discretos (SEDs) são sistemas dinâmicos cuja mudança de estado é orientada pela ocorrência de eventos instantâneos em qualquer instante do tempo [5]. Os SEDs estão presentes em diversos campos como manufatura, robótica, sistemas de comunicação, controle de tráfego e logística. A modelagem de SEDs pode ser realizada por meio de vários formalismos, sendo os mais comuns as Linguagens Formais e Autômatos, Redes de Petri, Cadeias de Markov e Teoria das Filas. Este trabalho trata da implementação da solução do problema de controle para SEDs representados através de Linguagens Formais e Autômatos.

A Teoria de Controle Supervisório (TCS) de Ramadge e Wonham trata da síntese de uma lógica de controle [42] para SEDs modelados por Linguagens Formais e Autômatos. O principal objetivo desta

abordagem é o projeto de estruturas denominadas supervisores que são responsáveis pela habilitação ou desabilitação de eventos conforme o estado em que estes supervisores se encontram. Estes supervisores são projetados de acordo com o sistema tratado e com as especificações de comportamento desejadas para ele. Estas especificações normalmente dizem respeito a roteamento, seqüenciamento, segurança, dentre outros. Segundo Brandin [4], o controle supervisório consiste de:

1. monitoramento do comportamento do sistema por meio de sensores;
2. determinação do controle de acordo com uma lei de controle supervisório que mapeia o comportamento do sistema para as respectivas ações de controle;
3. aplicação da ação de controle através dos programas em execução nos dispositivos de controle.

A escolha da abordagem do problema de controle através da TCS se deve a alguns fatores. Em primeiro lugar, a garantia de que os supervisores e as leis de controle são corretas por construção caso os modelos da planta e das especificações estejam corretos. Isto assegura que o comportamento do sistema controlado não viola as especificações consideradas. Além disso, os supervisores e as leis de controle são sempre minimamente restritivas, permitindo em todos os casos a ocorrência de todos os eventos possíveis que não contradigam as especificações estabelecidas. Por último, foi importante para esta escolha a experiência prévia do grupo de pesquisa no qual este trabalho está inserido, que tem utilizado a abordagem da TCS na maioria das suas pesquisas. As demais abordagens que tratam dos SEDs modelados por outros formalismos são, em sua grande maioria, de cunho meramente analítico, não oferecendo ferramentas de síntese como as oferecidas pela TCS.

Existe a tendência de que os sistemas de manufatura sejam flexíveis. Um sistema de manufatura flexível é aquele composto por várias células de manufatura que operam independente e concorrentemente. Cada uma destas células é responsável por uma ou mais tarefas específicas. Estas células são normalmente conectadas por sistemas de transporte. O sistema escolhido para implementação distribuída do sistema de controle neste trabalho foi uma célula de envasilhamento de bebidas, que nada mais é que um sistema de manufatura flexível com alguns subsistemas que operam independentemente visando realizar o envase de garrafas.

1.1 Motivação e Objetivos

A TCS ainda não é amplamente utilizada na resolução de problemas práticos devido a alguns fatores que limitam a sua aplicação. Um desses fatores limitantes trata-se da explosão de estados observada durante o procedimento de síntese do supervisor que abrange todas as especificações estipuladas. A complexidade computacional para a síntese e a quantidade de memória necessária para implementação dos supervisores em um dispositivo de controle convencional são fatores proibitivos na maioria dos casos. Outro fator importante refere-se à escassez de pessoal qualificado para o projeto e manutenção de sistemas de controle baseados na TCS.

Algumas extensões da TCS visam reduzir o número de estados dos autômatos envolvidos no procedimento de síntese, como é o caso do Controle Modular [43] e do Controle Modular Local [23]. Ambas as extensões propõem a síntese de vários supervisores, sendo um para cada especificação, que, operando em conjunto, atuam como se fossem um supervisor único que atende a todas as especificações simultaneamente. No caso do Controle Modular, para o cálculo de tais supervisores, o sistema é considerado como um todo, enquanto que para o caso do Controle Modular Local apenas os subsistemas afetados pela especificação em questão são considerados.

No entanto, mesmo com a utilização de tais extensões, pode não ser possível implementar todo o sistema de controle em um único dispositivo de controle como um controlador lógico programável (CLP), por exemplo. O principal fator que restringe a implementação concentrada é a característica do dispositivo de controle utilizado, sendo a memória disponível o maior problema. Outros fatores limitantes são: número de canais de entrada e saída de sinal, disponibilidade de recursos internos como variáveis internas, contadores e temporizadores, dentre outros. Outras vezes, pode ser necessário (re)utilizar uma parte do sistema de controle que já esteja implementada e otimizada em um dispositivo de controle específico. Esta é uma situação muito comum já que muitas vezes os fabricantes de equipamentos fornecem o produto com um sistema de controle otimizado. Uma forma simples de se solucionar o problema nos dois casos é a distribuição do sistema de controle em mais de um dispositivo de controle.

Em outros contextos a implementação distribuída do controle supervísório também pode ser extremamente desejável, visto que por meio dela é possível preservar a modularidade do sistema físico e das especificações. Neste caso, uma implementação distribuída pode trazer diversos benefícios, inclusive financeiros, com a redução dos custos para interligação dos equipamentos ao CLP único. Com a distribuição pode-se alocar cada dispositivo de controle próximo aos subsistemas que ele controla, modularizando-se o sistema global. Além destas vantagens, acrescenta-se o fato de ser possível adicionar ou remover subsistemas físicos e especificações com maior facilidade, assim como a possibilidade de operação parcial do sistema em caso de falha ou até mesmo para manutenção.

Existem alguns aspectos intrínsecos à implementação distribuída do controle supervísório para SEDs que devem ser tratados. Não há na literatura muitos trabalhos que lidem com a implementação distribuída do controle supervísório. Vieira [35] aborda em seu trabalho alguns dos aspectos a serem observados durante o procedimento de distribuição de um sistema de controle, sendo a comunicação entre os dispositivos de controle e a sincronização do processamento das estruturas de controle nos dispositivos os mais importantes.

A presente dissertação de mestrado tem como objetivo validar o modelo de Vieira [35] por meio da implementação distribuída do controle supervísório em uma célula flexível real, além de propor critérios para análise de configurações de distribuição de sistemas de controle supervísório. Durante esse processo alguns aspectos de ordem prática e até mesmo teórica que tenham sido tratados de forma diferenciada no referido trabalho também são elucidados. Deste modo é realizada uma adaptação do modelo de Vieira [35] para utilização prática em um sistema com suas peculiaridades. No sistema tratado o controle de baixo nível já estava implementado e otimizado, o que fez com que uma adaptação de cunho prático tivesse que ser realizada, possibilitando a integração deste nível aos de-

mais níveis de controle nestas circunstâncias. No âmbito teórico propõe-se algumas alterações quanto à forma de representação além da generalização de conceitos, contribuindo para a clareza e abrangência do referido trabalho. Alcançados estes objetivos é possível mostrar a adequação da abordagem de controle modular local a situações de implementação distribuída na resolução de problemas com restrição de *hardware*.

1.2 Estrutura

Esta dissertação está organizada da seguinte forma: o Capítulo 2 apresenta os conceitos de linguagens e autômatos, além de uma revisão bibliográfica sobre a Teoria de Controle Supervisório e de duas de suas extensões, o Controle Modular e o Controle Modular Local.

No Capítulo 3 o problema de controle supervisório é resolvido para a célula de envasilhamento de bebidas, sistema utilizado ao longo de todo este trabalho. A metodologia para a síntese dos supervisores é mostrada em detalhes, desde a modelagem do sistema e definição das especificações até a obtenção dos supervisores modulares.

O Capítulo 4 apresenta uma revisão bibliográfica sobre modelos para implementação do controle supervisório, com ênfase para o modelo proposto inicialmente por Queiroz [24] e posteriormente formalizado por Vieira [35]. Este último estabeleceu uma etapa intermediária entre a etapa de modelagem do sistema físico e especificações de comportamento para síntese de supervisores e a etapa de implementação do programa de controle em uma linguagem de programação específica. Este novo formalismo facilita a construção de compiladores para geração automática de código já que define todas as variáveis a serem utilizadas no programa e suas inter-relações. Ainda nesse capítulo este modelo é aplicado às estruturas do sistema de controle encontradas para a célula de envasilhamento de bebidas no Capítulo 3.

No Capítulo 5 os aspectos referentes ao controle supervisório distribuído são apresentados. Em um primeiro momento trata-se da distribuição vertical, que possibilita a distribuição do controle de baixo nível do sistema. Depois é apresentada a distribuição horizontal, complementar à distribuição vertical. Por fim, mostra-se como realizar a distribuição geral do sistema de controle através da associação da distribuição vertical com a horizontal.

O Capítulo 6 traz a implementação distribuída do controle supervisório no sistema de manufatura escolhido. Em um primeiro momento são abordados aspectos relativos às configurações de hardware disponíveis e à tecnologia utilizada na realização da comunicação entre os dispositivos de controle. Em um segundo momento são propostas e realizadas três configurações de implementação distribuída, cada qual com suas particularidades. Em seguida é realizada uma discussão e a proposição dos critérios a serem avaliados na escolha de uma configuração de distribuição na abordagem de um problema real. Por fim, são abordados alguns aspectos considerados relevantes no que diz respeito à estrutura dos programas implementados nos CLPs.

Por fim, no Capítulo 7 são tecidas algumas considerações finais e apresentadas as conclusões do trabalho.

Capítulo 2

Controle Supervisório de Sistemas Compostos

A grande maioria dos sistemas de manufatura existentes pode ser tratada como a composição de diversos subsistemas que trabalham independentemente. Tais sistemas são chamados de Sistemas Compostos. O controle e a coordenação do funcionamento dos diversos subsistemas visando a realização precisa das várias atividades esperadas do sistema composto é um grande desafio. Este capítulo aborda a Teoria de Controle Supervisório (TCS)[42], utilizada para o desenvolvimento da lógica de controle para problemas relacionados a um Sistema a Eventos Discretos (SED).

Um SED é um sistema dinâmico que muda seu estado conforme a ocorrência de eventos instantâneos em qualquer instante do tempo [5]. O fator que determina de que forma o sistema evolui é a seqüência na qual os eventos ocorrem. A Teoria de Controle Supervisório possibilita a síntese automática de controladores (supervisores) ótimos para SEDs. Para isso, faz uso de um modelo da planta e das especificações, que representam o comportamento do sistema sujeito às restrições de projeto. Nesta abordagem assume-se que os eventos não podem ocorrer simultaneamente. São exemplos de eventos o início e o término de operação de uma máquina, a mudança de estado de um sensor, quebra de uma máquina, dentre outros.

A especificação do comportamento de um sistema a eventos discretos consiste no estabelecimento de seqüências ordenadas de eventos que propiciem a realização de determinadas tarefas [30]. O modelo desenvolvido por Ramadge e Wonham [29], baseado na teoria de autômatos e/ou linguagens, tem como grande vantagem o fato de apresentar procedimentos formais para a síntese dos supervisores. Por essa razão, será o modelo explorado neste texto. Mais detalhes acerca deste assunto podem ser encontrados em [41] e [9].

Este capítulo inicia-se com alguns conceitos fundamentais para a modelagem de SEDs sobre os quais se baseia a TCS de Ramadge e Wonham, que também é apresentada. Além disso, algumas extensões desta teoria, como o controle modular clássico e o controle modular local, são abordadas ao final do capítulo.

2.1 Linguagens e Autômatos

Para determinar como o comportamento lógico de um SED pode ser modelado a partir de linguagens e autômatos, são apresentados alguns aspectos fundamentais referentes a estas teorias. A partir de um modelo formal é possível então realizar a síntese do controlador, principal objetivo da Teoria de Controle Supervisório.

2.1.1 Linguagens

Um alfabeto Σ é definido como um conjunto finito de símbolos distintos. Σ^* é o conjunto de todas as cadeias finitas de elementos em Σ , incluindo a cadeia vazia ε .

Definição 2.1.1 *Uma linguagem é um subconjunto de Σ^* .*

Sejam três cadeias $u, v, s \in \Sigma^*$. Se, para alguma cadeia s , $v = us$, u é prefixo de v . O prefixo fechamento de uma linguagem $L \in \Sigma^*$, denotado por \bar{L} , é dado por

$$\bar{L} = \{u \mid \exists v \in \Sigma^* \wedge uv \in L\}. \quad (2.1)$$

\bar{L} é, então, formada por todas as cadeias que são prefixo de L . Se $L = \bar{L}$, a linguagem L é chamada prefixo-fechada, ou seja, qualquer prefixo de qualquer cadeia de L é também uma cadeia de L . Nos casos dos sistemas físicos reais, a linguagem gerada será sempre prefixo-fechada.

Considere dois alfabetos Σ e Σ_i , sendo $\Sigma_i \subset \Sigma$. Muitas vezes é necessário retirar das cadeias de uma linguagem L todos os eventos que não pertencem a Σ_i , ou seja, os eventos que pertencem ao conjunto $(\Sigma - \Sigma_i)$. Para isso, define-se a operação de projeção natural de Σ^* em Σ_i^* , dada por $P_i : \Sigma^* \rightarrow \Sigma_i^*$. Esta é uma operação definida recursivamente por

$$\begin{aligned} P_i(\varepsilon) &= \varepsilon \\ P_i(\sigma) &= \begin{cases} \varepsilon & \text{se } \sigma \notin \Sigma_i \\ \sigma & \text{se } \sigma \in \Sigma_i \end{cases} \\ P_i(s\sigma) &= P_i(s)P_i(\sigma) \quad \text{com } s \in \Sigma^*, \sigma \in \Sigma. \end{aligned} \quad (2.2)$$

Para linguagens regulares, que podem ser expressas através de expressões regulares, o conceito de projeção natural pode ser estendido da seguinte forma:

$$P_i(L) = \{s' \in \Sigma_i^* \mid s' = P_i(s) \text{ para algum } s \in L\} \quad (2.3)$$

Já em outras situações, pode ser necessário inserir os eventos pertencentes a $(\Sigma - \Sigma_i)$ nas cadeias que compõem L . A operação de projeção inversa, $P_i^{-1} : \Sigma_i^* \rightarrow \Sigma^*$, serve a este propósito e é definida como:

$$P_i^{-1}(L) = \{s \in \Sigma^* | P_i(s) \in L\} \quad (2.4)$$

O produto síncrono entre linguagens $L_i \subseteq \Sigma_i^*$, com $i = \{1, \dots, n\}$, é definido como:

$$\prod_{i=1}^n L_i = \bigcap_{i=1}^n P_i^{-1}(L_i) \quad (2.5)$$

Todo SED pode ter seu comportamento determinado por um par de linguagens. O alfabeto nesta representação corresponde a todos os eventos que são considerados na modelagem do sistema. O modelo que descreve o comportamento do sistema é dado pela dupla $D = (L, L_m)$.

Neste modelo, o conjunto de todas as seqüências de eventos fisicamente possíveis é dado pela linguagem $L \subseteq \Sigma^*$, que descreve o comportamento gerado pelo sistema. Para representar o conjunto de seqüências de eventos que determinam a realização de uma tarefa completa, é definida a linguagem L_m , que representa o comportamento marcado do sistema.

Duas propriedades importantes das linguagens L e L_m que determinam um SED podem ser definidas:

- O comportamento gerado de um SED contém o comportamento marcado, ou seja, $L \supset L_m$;
- O comportamento gerado de um SED é prefixo-fechado, ou seja, $L = \bar{L}$.

Todavia, a representação de SEDs por meio de linguagens é limitada computacionalmente. Os autômatos surgem, então, como um modelo mais eficiente para representar esta categoria de sistemas, como pode ser visto na seção subsequente.

2.1.2 Autômatos e Geradores

Toda linguagem regular pode ser definida e representada por um autômato. Na verdade, um autômato é capaz de gerar qualquer linguagem regular. Um autômato é uma quintupla $A = (X, \Sigma, \delta, x_0, X_m)$ onde:

- X é o conjunto, não-vazio e finito, de estados;
- Σ é o alfabeto de eventos;
- $\delta : X \times \Sigma \rightarrow X$ é a função de transição de estados;

- $x_0 \in X$ é o estado inicial do autômato;
- $X_m \subseteq X$ é o conjunto de estados finais ou marcados.

Para os casos em que a função de transição de estados é uma função parcial, ou seja, que é definida apenas para alguns elementos do domínio $X \times \Sigma \rightarrow X$ o termo correto a ser utilizado para esta representação é gerador, e não mais autômato.

A função de transição de estados δ pode ser estendida para operar sobre cadeias, tornando-se a função $\hat{\delta}$. Neste caso, o domínio da função passa a ser $\hat{\delta} : X \times \Sigma^*$ e ela é definida da seguinte forma:

$$\hat{\delta}(x, \epsilon) = x; \quad (2.6)$$

$$\hat{\delta}(x, \sigma) = \delta(x, \sigma), \sigma \in \Sigma; \quad (2.7)$$

$$\hat{\delta}(x, s\sigma) = \delta(\hat{\delta}(x, s), \sigma) \text{ para } s \in \Sigma^*, e \sigma \in \Sigma \quad (2.8)$$

Por simplicidade de notação, $\hat{\delta}$ será representada apenas por δ .

Graficamente, um autômato pode ser representado por um grafo dirigido, no qual os nós são os estados e os ramos etiquetados são as transições de estado. Círculos com linhas duplas representam os estados finais e uma seta aponta para o estado inicial do autômato.

Um autômato é capaz de reconhecer as duas linguagens associadas à representação de um SED: a linguagem que representa o comportamento gerado e a que representa o comportamento marcado. A linguagem gerada por $A = (X, \Sigma, \delta, x_0, X_m)$ é:

$$L(A) := \{s \in \Sigma^* | \delta(x_0, s) \text{ é definida}\} \quad (2.9)$$

A linguagem marcada de A é:

$$L_m(A) := \{s \in L(A) | \delta(x_0, s) \in X_m\} \quad (2.10)$$

Isso significa que partindo do estado inicial, a linguagem $L(A)$ é formada pelas seqüências de eventos que podem ocorrer seguindo-se as transições do autômato. Já a linguagem $L_m(A)$ representa todas as seqüências de eventos que, partindo do estado inicial, chegam em algum estado marcado.

Em um autômato, um estado é chamado de acessível quando existe uma cadeia de eventos que, a partir do estado inicial, leva ao estado considerado. Isto é o mesmo que dizer que $x \in X$ é acessível se existe $s \in \Sigma^*$ tal que $\delta(x_0, s) = x$. Um estado é co-acessível se existe uma cadeia de eventos a partir dele que leva a um estado marcado, ou seja, para $x \in X$, existe $s \in \Sigma^*$ tal que $\delta(x, s) \in X_m$.

Um autômato é acessível quando todos os seus estados são acessíveis e é co-acessível quando todos os seus estados são co-acessíveis. Um autômato acessível e co-acessível é dito aparado ou *trim*.

A operação de composição síncrona de geradores realiza a evolução em paralelo de dois (ou mais) geradores. Nesta operação, a transição entre estados causada por um evento presente em mais de um gerador só é possível caso este evento esteja definido em todos os geradores que o possuem em seus alfabetos.

A composição síncrona de n geradores $G_i = (X_i, \Sigma_i, \delta_i, x_{0i}, X_{mi})$ com $i = \{1, \dots, n\}$ resulta em um gerador G , com alfabeto $\Sigma = \cup_{i=1}^n \Sigma_i$, descrito pelas seguintes linguagens:

$$L(G) = \prod_{i=1}^n L(G_i); \quad L_m(G) = \prod_{i=1}^n L_m(G_i) \quad (2.11)$$

2.2 Sistemas Compostos

Um SED real pode ser analisado segundo uma abordagem global ou segundo uma abordagem local. Na abordagem global, um único gerador representa todas as seqüências de eventos possíveis e determina todas as tarefas a serem realizadas. Já na abordagem local, o sistema é considerado como sendo a composição de vários sub-sistemas. Sistemas deste tipo são chamados Sistemas Compostos, nos quais a composição do modelo de todas as sub-plantas resulta no sistema global. Na indústria, os Sistemas Compostos são muito comuns, uma vez que um processo normalmente é constituído por diversos equipamentos (sub-sistemas) que funcionam independentemente e devem atuar de forma coordenada e sincronizada para a realização de determinadas tarefas.

O tratamento de Sistemas Compostos apresenta algumas vantagens em relação à abordagem global, ou monolítica. A modelagem global de um SED de grande porte é, normalmente, uma tarefa de bastante complexidade. Já o modelo considerando-se os subsistemas é mais intuitivo, já que explora a modularidade e distribuição natural dos mesmos. Além disso, quando alguma alteração no modelo ou na lógica de controle é necessária, na abordagem global, todo o modelo precisa ser reformulado. Para o caso da representação por Sistemas Compostos, a alteração é feita apenas no modelo especificamente afetado, uma vez que as restrições de coordenação impostas pelo controlador também são projetadas com relação às subplantas.

Caso os alfabetos dos geradores que representam os subsistemas sejam disjuntos, o Sistema Composto é concorrente, porém assíncrono. Quando a intersecção dos alfabetos não é vazia e os eventos pertencentes a esta intersecção causam transições de estados nos geradores em questão, o Sistema Composto é concorrente e síncrono [38]. Neste caso, há sincronização por meio dos eventos comuns, ou seja, estes eventos ocorrem simultaneamente em todos os geradores que os possuem em seus alfabetos.

Formalmente, um sistema composto por n sub-plantas $G_i = (X_i, \Sigma_i, \delta_i, x_{0i}, X_{mi})$, com $i = \{1, \dots, n\}$, tem sua planta global dada pela composição síncrona dos geradores das sub-plantas:

$$G = \prod_{i=1}^n G_i \quad (2.12)$$

Essa forma de se representar um SED foi denominada Representação por Sistemas Compostos (RSC) por Queiroz [23], na qual o alfabeto da planta global é dado por $\Sigma = \cup_{i=1}^n \Sigma_i$.

2.3 Controle Supervisório Monolítico

Como foi discutido, existem maneiras formais de se representar o comportamento livre de um SED. No entanto, muitas vezes é necessário restringir de alguma forma este comportamento para, por exemplo, evitar um bloqueio do sistema ou, ainda, alguma cadeia de eventos possível mas indesejada. Estas restrições de comportamento e coordenação do sistema definem uma especificação. De posse do modelo global da planta e da especificação, encontra-se a linguagem que representa o comportamento controlado, ou em malha fechada, do sistema.

Segundo Ramadge e Wonham [30], em sua abordagem de linguagens controláveis, o alfabeto de eventos do gerador que modela a planta é particionado em dois subconjuntos disjuntos Σ_c e Σ_u ($\Sigma = \Sigma_c \cup \Sigma_u$) de eventos controláveis e não controláveis, respectivamente. Um evento é considerado controlável quando pode ter sua ocorrência impedida por algum tipo de agente externo. Um evento é não controlável quando sua ocorrência não pode ser evitada.

Para que o SED comporte-se conforme a linguagem do sistema controlado, introduz-se um agente de controle denominado supervisor que observa os eventos que ocorrem na planta e, a partir da cadeia de eventos observada, permite ou não a ocorrência dos eventos controláveis fisicamente possíveis. Esta estrutura de controle pode ser vista na Figura 2.1.

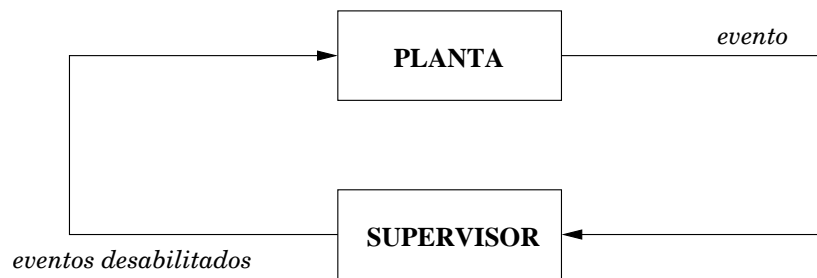


Figura 2.1: Esquema de funcionamento do controle monolítico

Um supervisor é um par $s = (A, \psi)$, onde A é um autômato que gera uma linguagem sobre o mesmo alfabeto Σ da planta G e ψ é um mapeamento entre o par (eventos de Σ , estados de A) e o conjunto {habilitado, desabilitado}. O supervisor s pode ser representado por um gerador S que implementa a ação de controle sobre a planta G implicitamente na sua estrutura de transição. O comportamento do sistema sob supervisão é determinado pelo autômato S/G cuja linguagem gerada $L(S/G)$ é formada pelas cadeias aceitas por S e G que possuam todos os eventos habilitados por ψ . A função do supervisor é desabilitar na planta os eventos que, após uma cadeia observada, não estejam habilitados no supervisor. Já a linguagem marcada do sistema sob a ação do supervisor é $L_m(S/G) = L(S/G) \cap L_m(G)$. A composição síncrona de S e G , $S \parallel G$, determina o SED que descreve o comportamento do sistema sob supervisão se S for admissível.

Um supervisor é não bloqueante (ou próprio) se $\overline{L_m(S/G)} = L(S/G)$ [41].

Outro conceito importante no estudo do controle supervisório é o de controlabilidade de uma linguagem. Uma linguagem $K \subseteq \Sigma^*$ é controlável em relação a (e.r.a) uma linguagem G se:

$$\overline{K}\Sigma_u \cap L(G) \subseteq \overline{K} \quad (2.13)$$

Em palavras, se uma linguagem é controlável, qualquer cadeia de eventos fisicamente possível pertencente a G , seguida de um evento não controlável, é uma cadeia que também pertencente a \overline{K} . O objetivo do controle supervisório é encontrar um supervisor que implemente uma linguagem K de acordo com as especificações de controle e que seja minimamente restritiva. A classe de linguagens controláveis contidas numa linguagem L é dada por $C(L, G) = \{K | K \subseteq L \text{ e } K \text{ é controlável e.r.a } G\}$, que é um conjunto não vazio e fechado em relação à operação de união de conjuntos. Por esta razão, existe um elemento supremo denominado $Sup C(L, G)$ que é a máxima linguagem controlável contida em L [42]. Um algoritmo para o cálculo da máxima linguagem controlável é descrito detalhadamente em [9].

Considere a linguagem $K \subseteq L_m(G)$ como sendo a linguagem que descreve o comportamento desejado do sistema G em malha fechada. Existe um supervisor não bloqueante S que implementa K se e somente se K é controlável e K é L_m prefixo-fechada.

Mesmo que a linguagem desejada K não atenda alguma das condições de existência do supervisor, ainda assim é possível encontrar um supervisor não-bloqueante S que implemente a máxima linguagem controlável contida na especificação K . Se a restrição no comportamento desejado imposta por $Sup C(K, G)$ for aceitável, o problema de controle tem solução e $L_m(S/G) = Sup C(K, G)$. Caso contrário, o problema é dito sem solução.

O procedimento para o projeto de um supervisor monolítico consiste dos seguintes passos:

1. modelar o comportamento livre da planta G , que pode também ser obtido pela composição das subplantas G_i , para $i = \{1, \dots, n\}$;
2. modelar a especificação monolítica E , que pode também ser obtida pela composição de todas as especificações E_j , com $j = \{1, \dots, m\}$;
3. compor E com $L_m(G)$;
4. obter a linguagem desejada K para o sistema sob supervisão, encontrando a componente trim do autômato que representa a linguagem $E || L_m(G)$;
5. calcular o supervisor S que gera a máxima linguagem controlável contida em K .

Cabe neste ponto ressaltar que existem procedimentos para a redução de supervisores, como o proposto por Vaz e Wonham [34], que possibilitam uma diminuição, muitas vezes considerável, do número de estados do supervisor encontrado. O exemplo que segue ilustra o procedimento de síntese de um supervisor monolítico.

Exemplo 2.1 Considere um sistema constituído por duas máquinas M_1 e M_2 , cujos modelos são mostrados na Figura 2.2. Cada máquina apresenta 3 estados, sendo eles: máquina parada (0), máquina funcionando (1) e máquina com defeito (2). Os eventos nesse modelo são, para $i = \{1, 2\}$:

- $s_i \in \Sigma_c$: início de operação da máquina M_i ;
- $f_i \in \Sigma_u$: fim de operação da máquina M_i ;
- $b_i \in \Sigma_u$: defeito na máquina M_i ;
- $r_i \in \Sigma_c$: fim do conserto da máquina M_i .

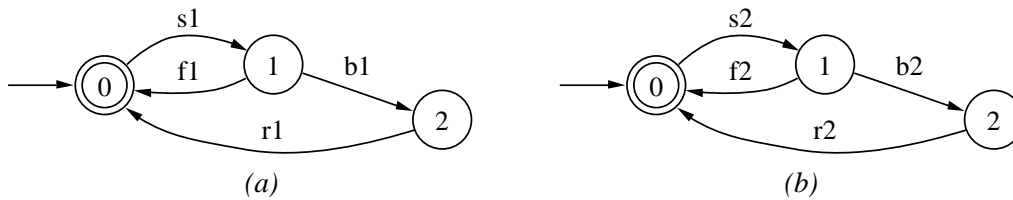


Figura 2.2: Exemplo: (a) M_1 ; (b) M_2

O sistema possui ainda um buffer B com capacidade unitária entre as máquinas, como mostrado na Figura 2.3. Deseja-se que (1) não haja "underflow" nem "overflow" de B ; e (2) M_2 tenha prioridade de reparo caso as 2 máquinas quebrem. Estas especificações são mostradas na Figura 2.4.

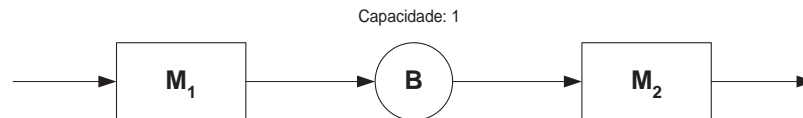


Figura 2.3: Exemplo: Esquemático do Sistema

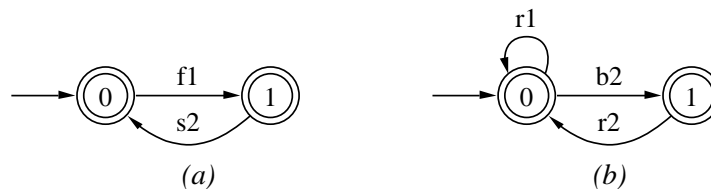


Figura 2.4: Autômatos que geram as linguagens: (a) E_1 ; (b) E_2

Para a resolução deste problema calcula-se inicialmente $G = M_1 || M_2$. Em seguida, calcula-se a especificação monolítica $E = E_1 || E_2$ que, composta com a planta G , resulta na linguagem desejada K . A síntese do supervisor monolítico S é realizada por meio do cálculo de $Sup C(K, G)$, cujo autômato gerador é mostrado na Figura 2.5. Com o auxílio do algoritmo de redução de supervisores implementado na ferramenta CTCT disponibilizada pelo Systems Control Group - University of Toronto, chega-se ao supervisor reduzido S_{red} mostrado na Figura 2.6.

◇

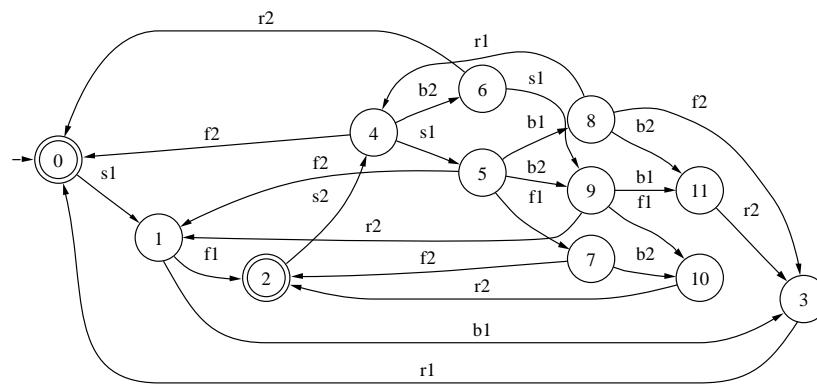


Figura 2.5: Autômato que gera linguagem $Sup C(K, G)$

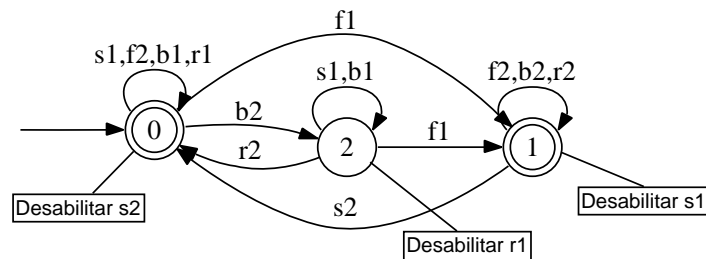


Figura 2.6: Supervisor monolítico reduzido

2.4 Controle Modular

Apesar de apresentar uma solução para o problema de controle supervisório de SEDs, a teoria proposta por Ramadge e Wonham apresenta uma grande limitação prática. Trata-se do procedimento de síntese do supervisor monolítico, cuja complexidade computacional é polinomial no tamanho dos modelos da planta e da especificação. Para modelos com muitos estados, a síntese do supervisor pode levar a uma explosão no número de estados. Por isso, para os sistemas reais, que normalmente são de grande porte, a aplicação deste método fica inviabilizada.

Na tentativa de resolver esta questão, Wonham e Ramadge [43] propõem o projeto de vários supervisores em oposição ao supervisor único considerado no controle monolítico. Para isso, cada subtarefa ou restrição do sistema é considerada uma especificação, a partir da qual é projetado cada um dos supervisores. Estes supervisores, se satisfeita uma propriedade de modularidade, irão atuar concorrentemente implementando a solução para o problema de controle original.

Como já foi dito na seção 2.2, sistemas de grande porte reais são, normalmente, constituídos por diversos subsistemas que realizam suas atividades paralelamente. A idéia do controle modular local apresentada por Queiroz [23], explora exatamente aspectos relacionados à modularidade dos sistemas. Desta forma, muitas vezes não é necessário compor todas as subplantas para o cálculo do supervisor modular, bastando apenas considerar as subplantas que sofrem influência direta da especificação em questão.

As duas seções subseqüentes detalham o controle modular clássico e o controle modular local.

2.4.1 Controle Modular Clássico

No intuito de reduzir a complexidade de síntese de um supervisor único e aumentar a flexibilidade de projeto, a teoria de controle supervisório de Ramadge e Wonham foi estendida. No controle modular clássico, as especificações de controle são divididas em várias subtarefas, que são tratadas individualmente. Os subcontroladores, ou supervisores modulares, encontrados operam concorrentemente para implementar uma solução para o problema de controle original. Um evento é desabilitado se pelo menos um dos supervisores modulares o desabilitar. A Figura 2.7 ilustra um sistema sob a ação de supervisores modulares.

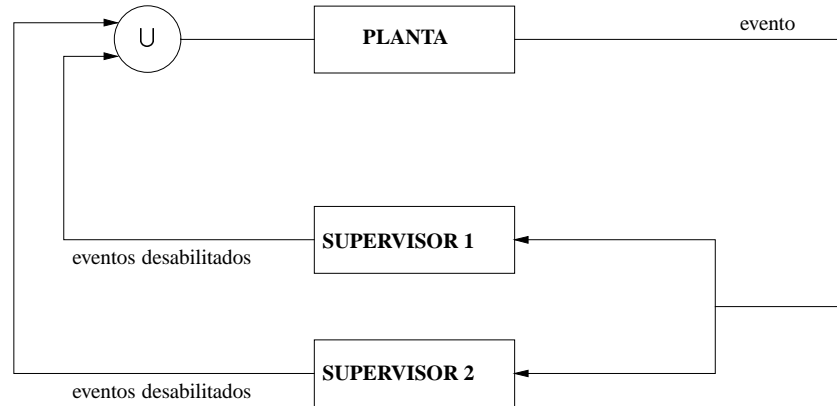


Figura 2.7: Esquema de funcionamento do controle modular

Além de ser mais facilmente sintetizado, um supervisor modular é também mais facilmente modificado, atualizado e corrigido [3]. Se for necessário, por algum motivo, modificar apenas uma das especificações será preciso reprojeter apenas o supervisor correspondente, o que confere maior flexibilidade à síntese modular. A exclusão ou inclusão de qualquer especificação também não interfere nos demais supervisores. Por possuírem também um menor número de estados, os supervisores modulares podem ser interpretados com maior facilidade pelo projetista além de possibilitarem as suas implementações em um dispositivo de controle como, por exemplo, um controlador lógico programável (CLP).

Entretanto, como os supervisores modulares possuem apenas uma visão local do sistema, as ações de controle implementadas por eles podem ser conflitantes em um âmbito global. Por esta razão, a solução modular de um problema pode ser degradada em relação à solução monolítica. Neste contexto, para que a solução modular seja equivalente à solução monolítica, a seguinte propriedade, denominada modularidade, deve ser satisfeita:

$$\bigcap_{j=1}^m \overline{L_m(\text{Sup } C(E_j, G_j))} = \overline{L_m(\text{Sup } C(E_j, G_j))} \quad (2.14)$$

Em palavras, esta propriedade consiste em testar se todo prefixo de uma cadeia marcada comum a todos supervisores S_j , com $j = \{1, \dots, m\}$, é também prefixo de pelo menos uma cadeia da composição das linguagens geradas por cada supervisor S_j . A modularidade das linguagens duas a duas não implica na modularidade global das linguagens. Assim sendo, é necessário calcular a linguagem

que representa a interseção das linguagens geradas por todos os supervisores modulares. Essa é uma tarefa que pode representar elevados custos em termos computacionais.

Gohari e Wonham [11] fazem um estudo da complexidade computacional para resolução do problema de controle supervisório para SEDs e chegam à conclusão de que a abordagem modular não oferece ganhos em relação à abordagem monolítica. O problema da explosão de estados é inevitável nos dois casos.

Não existe nenhuma desvantagem em relação ao tempo de computação quando é considerada a computação monolítica em oposição à computação modular. A grande vantagem da abordagem modular é a potencial redução exponencial na quantidade de informação a ser armazenada que ocorre quando o sistema em questão é eficientemente transformado de um sistema monolítico para um sistema modular [31].

Em suma, os passos para o cálculo dos supervisores modulares são:

1. modelar o comportamento livre da planta G , que pode também ser obtido pela composição das subplantas G_i , para $i = \{1, \dots, n\}$;
2. modelar todas as especificações E_j , com $j = \{1, \dots, m\}$;
3. compor cada E_j com $L_m(G)$;
4. obter as linguagens desejadas K_j para o sistema sob supervisão modular, encontrando a componente trim de $E_j || L_m(G)$;
5. calcular cada supervisor modular S_j que gera a máxima linguagem controlável contida em cada K_j ;
6. realizar o teste de modularidade entre as linguagens geradas por todos os supervisores modulares obtidos.

O exemplo que segue ilustra o procedimento de síntese modular.

Exemplo 2.2 *Considere o mesmo sistema apresentado no Exemplo 2.1. Como as restrições são dadas por meio de 2 especificações, 2 supervisores modulares deverão ser sintetizados. Os autômatos mostrados na Figura 2.8 geram a máxima linguagem controlável contida em K_1 e K_2 , que são as linguagens desejadas formadas pela composição da planta G com as especificações E_1 e E_2 , respectivamente.*

O teste de modularidade é realizado para checar se a ação dos supervisores modulares não é conflitante. Neste exemplo os supervisores são modulares e portanto S_1 e S_2 são soluções para o problema de controle.

◇

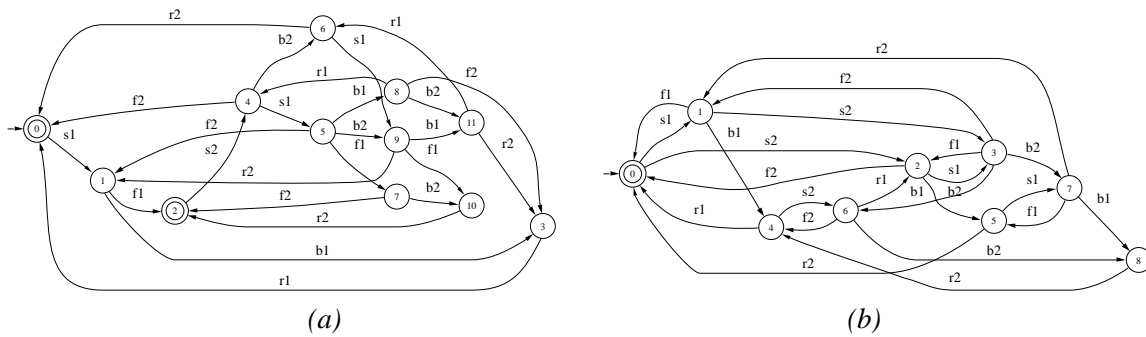


Figura 2.8: Autômatos que geram linguagens: (a) $Sup C(K_1, G)$; (b) $Sup C(K_2, G)$

2.4.2 Controle Modular Local

Uma grande limitação para a implementação prática tanto do controle monolítico quanto do modular é que em ambos os casos a síntese do(s) supervisor(es) depende do modelo global da planta. Conforme foi discutido na seção 2.2, a maioria dos sistemas de manufatura reais são de grande porte e normalmente podem ser modelados como uma RSC.

A abordagem de Controle Modular Local apresentada por Queiroz [23], [26] [25] [27], propõe uma abordagem do problema de controle supervisório que tira proveito da característica descentralizada do modelo em malha aberta de sistemas compostos. Com isso, são reduzidos os esforços computacionais no processo de síntese e implementação do controle. A Figura 2.9 mostra o diagrama de funcionamento de um sistema de controle projetado segundo a abordagem de Controle Modular Local.

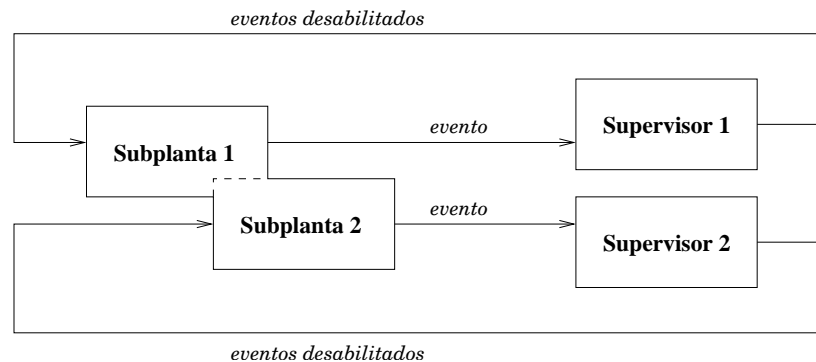


Figura 2.9: Esquema de funcionamento do controle modular local [23]

Na abordagem modular local o sistema deve estar modelado sob a forma de uma Representação por Sistema Produto (RSP), como mostrado na seção 2.4.2.1.

2.4.2.1 Sistema Produto

Um sistema produto [30] é uma RSC na qual todas as subplantas são modeladas por geradores assíncronos. Compondo-se as subplantas síncronas de qualquer sistema composto é possível obter um sistema produto equivalente.

A RSP mais refinada é obtida quando são compostos apenas os subsistemas síncronos resultando no maior número de novos subsistemas assíncronos possível. A Figura 2.10 ilustra o procedimento para a obtenção da RSP mais refinada. Em um primeiro momento são modelados os subsistemas, representados por círculos, como mostrado na Figura 2.10(a). Os círculos com sobreposição indicam que os subsistemas têm eventos em comum, ou seja, são síncronos. A Figura 2.10(b) mostra a RSP mais refinada após a composição dos sistemas síncronos, que origina quatro sistemas assíncronos.

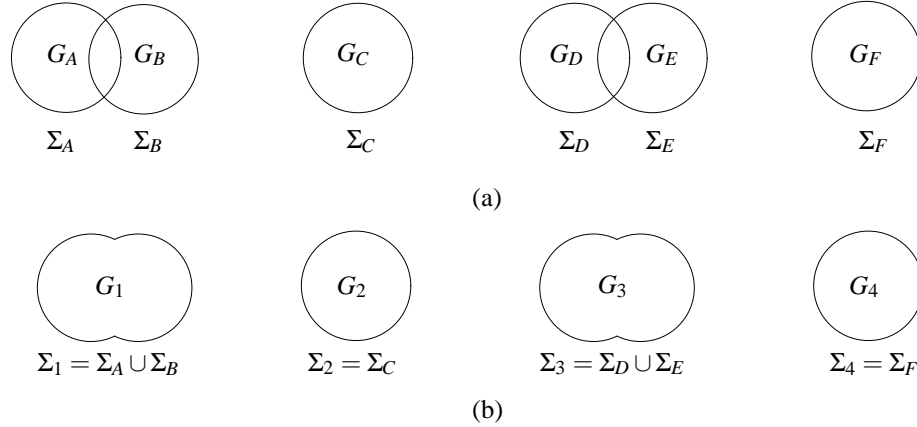


Figura 2.10: RSP: (a) modelo dos subsistemas (b) RSP mais refinada

Na abordagem modular local, quanto mais refinada a RSP melhores os resultados obtidos. No pior caso, quando todas as subplantas são síncronas, os resultados obtidos são os mesmos da abordagem modular clássica.

2.4.2.2 Metodologia

A grande vantagem do controle modular local em relação ao controle modular clássico é que o primeiro leva em conta na síntese de um supervisor modular apenas os subsistemas da RSP que possuem eventos em comum com a especificação em questão. Considere o sistema produto G formado pelas subplantas $G_i = (\Sigma_i, X_i, \delta_i, x_{0i}, X_{mi})$, $i \in I = \{1, \dots, n\}$, e as especificações genéricas locais E_{x_j} definidas sobre $\Sigma_{x_j} \subseteq \Sigma$, para $j \in J = \{1, \dots, m\}$. Nestas especificações são levados em conta apenas os eventos relevantes para a representação do comportamento desejado do sistema em malha fechada.

Uma planta local G_{loc_j} , para $j \in J = \{1, \dots, m\}$, é constituída pela composição de todos os componentes da RSP que possuem eventos em comum com a especificação genérica local E_{x_j} . Na Figura 2.11 a planta local G_{loc_a} é formada pelas subplantas G_1 e G_2 e a planta local G_{loc_b} é formada pelas subplantas G_2 e G_3 . A subplanta G_4 não faz parte de nenhuma planta local, já que não possui eventos em comum nem com E_a nem com E_b .

Definidas as plantas locais, deve-se encontrar a linguagem de cada uma das G_{loc_j} associada à especificação genérica local correspondente E_{x_j} . Estas linguagens encontradas são denominadas especificações locais E_{loc_j} . Os supervisores modulares locais S_{loc_j} são definidos pela máxima linguagem controlável contida em cada especificação local em relação à sua planta local, ou seja, $SupC(E_{loc_j}, G_{loc_j})$.

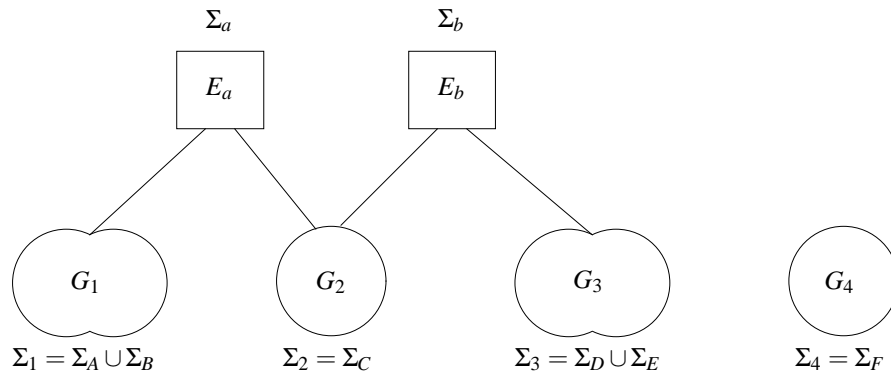


Figura 2.11: Constituição das plantas locais

Entretanto, o procedimento descrito acima não garante a resolução do problema de controle, uma vez que a atuação conjunta dos supervisores modulares locais pode gerar conflitos. Neste caso, em algumas situações, o sistema não é capaz de completar qualquer tarefa, ou seja, existem cadeias de eventos a partir das quais não é possível chegar a estados marcados. Por essa razão, é necessário assegurar que a linguagem gerada pelos supervisores satisfaça a propriedade de modularidade local, que exige que a seguinte igualdade seja satisfeita:

$$\overline{\|_{j=1}^m L_m(\text{Sup } C(E_{loc_{xj}}, G_{loc_{xj}}))} = \overline{\|_{j=1}^m L_m(\text{Sup } C(E_{loc_{xj}}, G_{loc_{xj}}))} \quad (2.15)$$

Segundo Queiroz e Cury [26], caso os supervisores modulares locais atendam à propriedade acima, é possível assegurar que a atuação conjunta destes supervisores apresenta a mesma performance do controle monolítico. Existem diversos métodos para a resolução de conflitos [7] [8] [39] [40] [1] [18], caso eles ocorram. Uma revisão bibliográfica abordando este assunto pode ser encontrada em [22].

A metodologia descrita nesta seção pode ser sintetizada pelos seguintes passos, propostos por Queiroz e Cury [27]:

1. modelar cada componente elementar do sistema isoladamente, evitando incluir eventos e estados desnecessários;
2. calcular a mais refinada RSP, fazendo a composição dos subsistemas síncronos;
3. modelar cada especificação isoladamente, considerando apenas os eventos relevantes;
4. obter a planta local para cada especificação compondo os subsistemas da RSP que tenham eventos em comum com a especificação;
5. calcular a linguagem de cada planta local que satisfaça a especificação, através do produto síncrono de cada planta local com sua respectiva especificação genérica;
6. calcular a máxima linguagem controlável contida em cada especificação local;
7. verificar a modularidade local das linguagens resultantes;

8. se não forem modulares, resolver os conflitos;
9. se forem modulares, implementar um supervisor reduzido para cada linguagem controlável.

O exemplo que segue traz uma aplicação do controle modular local.

Exemplo 2.3 Considere um sistema constituído por três máquinas M_1 , M_2 e M_3 , cujos modelos são mostrados na Figura 2.12. Cada máquina apresenta 2 estados, sendo eles: máquina parada (0) e máquina funcionando (1). Os eventos nesse modelo são, para $i = \{1, 2, 3\}$:

- $s_i \in \Sigma_c$: início de operação da máquina M_i ;
- $f_i \in \Sigma_u$: fim de operação da máquina M_i .

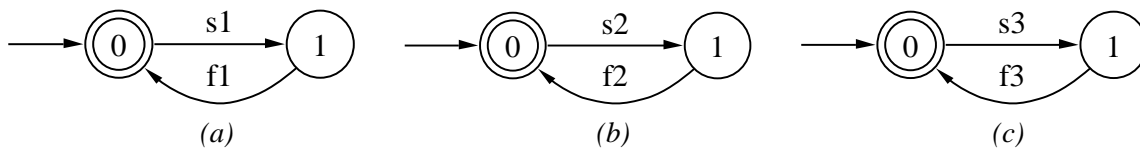


Figura 2.12: Exemplo: (a) M_1 ; (b) M_2 (b) M_3

O sistema possui ainda dois buffers B_1 e B_2 com capacidade unitária entre as máquinas, como mostrado na Figura 2.13. Deseja-se que não haja "underflow" nem "overflow" tanto de B_1 quanto de B_2 . Estas especificações genéricas são mostradas na Figura 2.14.

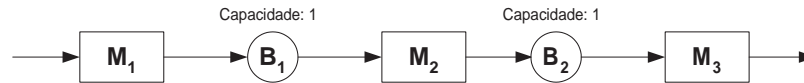


Figura 2.13: Exemplo: Esquemático do Sistema



Figura 2.14: Especificações Genéricas: (a) E_1 ; (b) E_2

Para este sistema, a planta local G_{loc1} é constituída por G_1 e G_2 e a planta local G_{loc2} por G_2 e G_3 . Compondo-se cada especificação genérica com suas respectivas plantas locais são obtidas as especificações locais E_{loc1} e E_{loc2} . Os supervisores S_{loc1} e S_{loc2} geram a máxima linguagem controlável contida em E_{loc1} e E_{loc2} (Figura 2.15), respectivamente.

O teste de modularidade é realizado para checar se a ação dos supervisores locais não é conflitante. Neste exemplo os supervisores são modulares e portanto S_{loc1} e S_{loc2} são soluções para o problema de controle. Pode-se ainda realizar a redução destes supervisores, que se tornam os supervisores S_{red1} e S_{red2} (Figura 2.16). Essa redução foi realizada com o auxílio do algoritmo de redução de supervisores implementado na ferramenta CTCT para controle supervisório disponibilizada pelo Systems Control Group - University of Toronto.

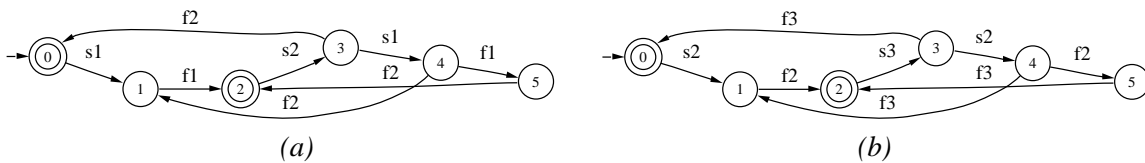


Figura 2.15: Autômatos que geram linguagens: (a) $Sup C(E_{loc1}, G_{loc1})$; (b) $Sup C(E_{loc2}, G_{loc2})$

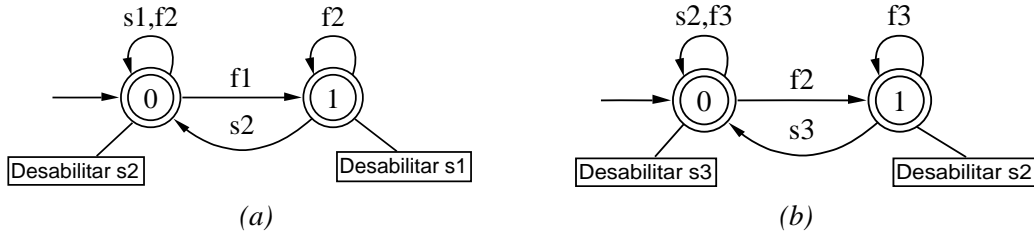


Figura 2.16: Supervisores Reduzidos: (a) S_{red1} ; (b) S_{red2}

2.5 Conclusão do Capítulo

Neste capítulo foram apresentados os fundamentos da teoria de controle supervisório de Ramadge e Wonham, assim como alguns dos conceitos básicos que dão suporte a esta teoria e algumas de suas extensões. Uma delas, o controle modular local, é de grande importância na medida em que possibilita uma grande redução da complexidade computacional na síntese de supervisores, principalmente no que tange a problemas de grande porte, bem como uma redução expressiva na quantidade de memória necessária para o armazenamento das estruturas que representam o modelo do sistema e os supervisores modulares.

Um exemplo de síntese de supervisores por meio da teoria de controle modular local é apresentado no próximo capítulo. Espera-se que a abordagem destes conceitos através de um exemplo auxilie no entendimento dos aspectos envolvidos no procedimento de síntese.

Capítulo 3

Controle Supervisório de uma Célula Flexível

Com o propósito de ilustrar os conceitos apresentados no capítulo 2, este capítulo traz um exemplo de projeto de supervisores por meio da teoria de controle modular local. O processo escolhido para o desenvolvimento de tal sistema de controle é uma célula flexível de manufatura cuja principal função é o envasilhamento de bebidas. Esta célula encontra-se no Laboratório de Automação Industrial (LAI) da Universidade Federal de Santa Catarina e é o sistema experimental adotado a partir de agora nesta dissertação, em que será realizada a implementação distribuída do controle supervisório em uma célula flexível.

Os resultados mostrados neste capítulo foram originalmente obtidos por Queiroz et al. [28], através da resolução do problema de controle para uma célula de manufatura semelhante à célula escolhida. A reprodução destes resultados é importante por se tratar do primeiro passo na construção do sistema distribuído, uma vez que os módulos do sistema produto e os supervisores modulares locais aqui encontrados são os componentes a serem implementados distribuídamente.

O capítulo inicia-se com uma breve descrição do sistema de manufatura escolhido e em seguida é apresentado o procedimento de síntese dos supervisores modulares locais que irão fazer com que o sistema funcione segundo as especificações de funcionamento desejadas.

3.1 Descrição da CEBE

O sistema de manufatura escolhido para o desenvolvimento deste trabalho foi uma planta industrial em escala reduzida que simula uma célula de envasilhamento de bebidas (CEBE). Essa planta, fruto da combinação de peças de kits de montagem tipo LEGO, dispositivos de controle (CLPs) e placas eletrônicas microprocessadas, realiza operações de movimentação, envase e fechamento de garrafas [2].

A CEBE é composta de três partes distintas. A primeira é a planta, que é o sistema físico composto de módulos responsáveis por executar operações específicas. Tem-se ainda o sistema de sensoriamento e acionamento, constituído de sensores, motores e cilindros pneumáticos responsáveis pelo controle interno de cada módulo. A terceira parte constituinte da CEBE é o sistema de controle, responsável por coordenar as operações de cada módulo da planta.

São cinco os módulos que constituem a CEBE, sendo quatro estações de trabalho dispostas ao redor de uma mesa giratória responsável pelo deslocamento das garrafas entre estações adjacentes, como mostrado na Figura 3.1. Estes cinco módulos apresentam as seguintes características:

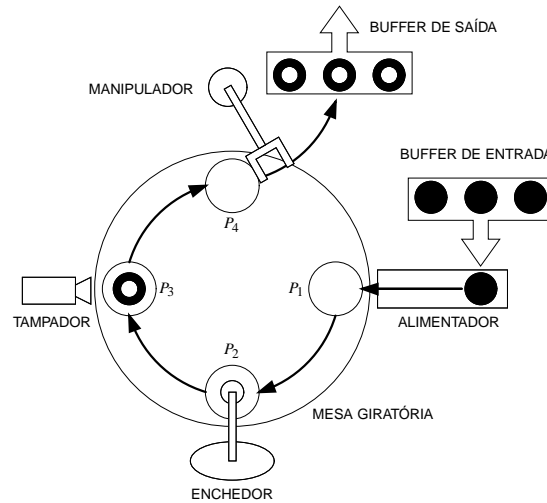


Figura 3.1: Diagrama esquemático da CEBE

1. Mesa giratória: disco responsável pela movimentação das garrafas entre as estações. Possui quatro *pallets* (posições na mesa) que acomodam as garrafas durante o processo. Este módulo opera executando giros de 90° no sentido horário, resultando no deslocamento de cada *pallet* para a estação de trabalho adjacente.
2. Alimentador: módulo responsável pela inserção das garrafas vazias no processo constituído por dois cilindros pneumáticos que, quando acionados, ejetam uma das garrafas do *buffer* de entrada, posicionando-a sobre um dos *pallets* da mesa giratória.
3. Enchedor: simula o processo de envase das garrafas. Possui um cilindro pneumático responsável pelo deslocamento vertical do bico enchedor.
4. Tampador: simula o processo de fechamento das garrafas. Possui um motor de corrente contínua que promove um giro de 720° do aro responsável pelo rosqueamento da tampa na garrafa.
5. Manipulador: módulo responsável por retirar as garrafas prontas do processo. Possui uma haste movida por dois motores de corrente contínua que por meio de uma garra manipuladora, retira a garrafa do *pallet* da mesa giratória e a deposita no *buffer* de saída. Esse módulo possui ainda dois sensores de fim de curso e um sensor de posição angular.

Duas placas microcontroladas do tipo *Handy Board* são responsáveis pelo controle de baixo nível da planta. Promovem a realização em cada um dos módulos das atividades que constituem uma operação, recebendo comandos de início de operação do sistema de controle e enviando respostas a ele sempre que uma operação é completada. Para a realização da comunicação entre as placas microcontroladas que implementam as seqüências operacionais e os demais níveis que compõem o sistema de controle foi estabelecida a troca serial de informações.

3.2 Projeto do Controle Supervisório para a CEBE

O funcionamento da célula sem a ação de um sistema de controle e coordenação pode levar a inúmeros inconvenientes. Apesar de cada módulo independentemente executar sua operação sem maiores problemas, algumas situações indesejáveis, e muitas vezes perigosas, surgem caso não haja um sistema de controle responsável pela coordenação do funcionamento destes subsistemas. Dentre elas, pode-se citar: sobreposição de garrafas, giro desnecessário da mesa giratória, fechamento de garrafa vazia, enchimento de garrafa já cheia, dentre outros.

O programa de controle original da CEBE permite que apenas uma garrafa seja processada por vez, ou seja, o processo só é alimentado com uma nova garrafa depois que o manipulador retira uma garrafa cheia e tampada da mesa. Esse modelo, apesar de evitar os problemas inerentes ao processamento em paralelo de garrafa, é muito ineficiente uma vez que os módulos ficam em repouso na maioria do tempo.

Deseja-se, portanto, que cada um dos módulos que constituem a CEBE operem paralelamente aumentando a produtividade do processo. Esse procedimento seria de certa forma trivial caso houvessem sensores de presença em cada um dos módulos. Neste caso, a lógica de controle faria com que o subsistema só entrasse em operação caso houvesse alguma garrafa no *pallet* sob ele. No entanto, isso não ocorre na CEBE e a condição instantânea do processo deve ser inferida a partir da evolução das operações que ocorreram no sistema a partir do estado inicial, quando a planta se encontrava vazia.

O problema de controle a ser resolvido consiste no projeto de uma lógica de controle a ser implementada em um ou mais CLPs que restrinja a planta o mínimo possível permitindo o processamento simultâneo de 0 a 4 garrafas sem que ocorram os problemas especificados (vide seção 3.2.2). O sistema deve também operar de forma contínua, ou seja, não pode haver situações de bloqueio.

Reproduz-se nesta seção o desenvolvimento do sistema de controle supervisório para a CEBE. Este procedimento consiste de 3 etapas sendo elas: representação do comportamento livre dos subsistemas físicos, representação das especificações genéricas e síntese de supervisores segundo abordagem modular local [28]. Estas etapas são abordadas nas seções subseqüentes.

3.2.1 Representação do Comportamento Livre dos Subsistemas Físicos

As Tabelas 3.1, 3.2, 3.3, 3.4 e 3.5 mostram os geradores que representam o comportamento livre da mesa giratória, alimentador, enchedor, tampador e manipulador respectivamente.

Tabela 3.1: Gerador G_0 (mesa giratória)

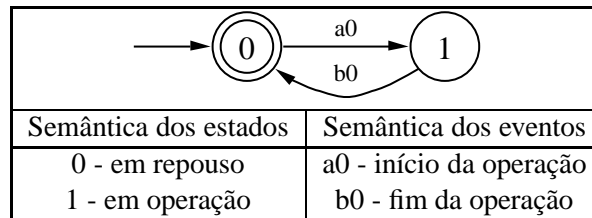


Tabela 3.2: Gerador G_1 (alimentador)

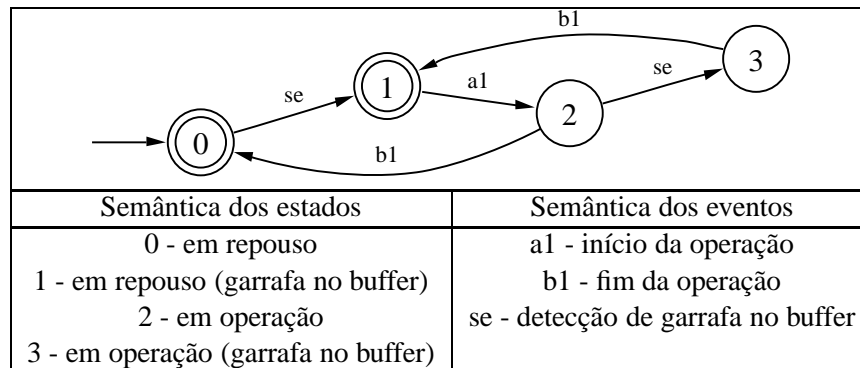
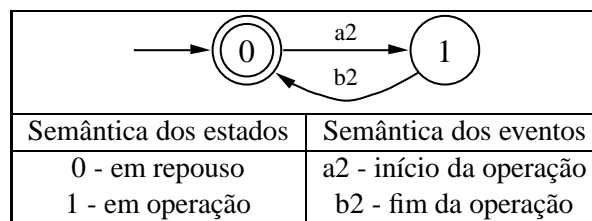


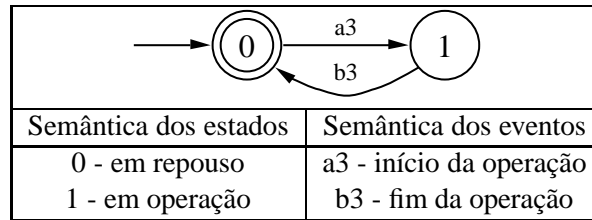
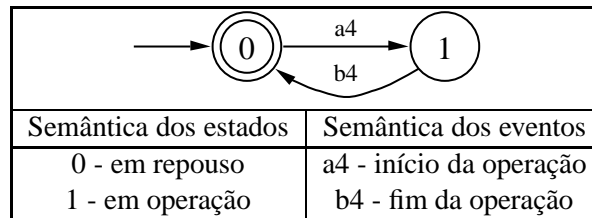
Tabela 3.3: Gerador G_2 (enchedor)



3.2.2 Representação das Especificações Genéricas

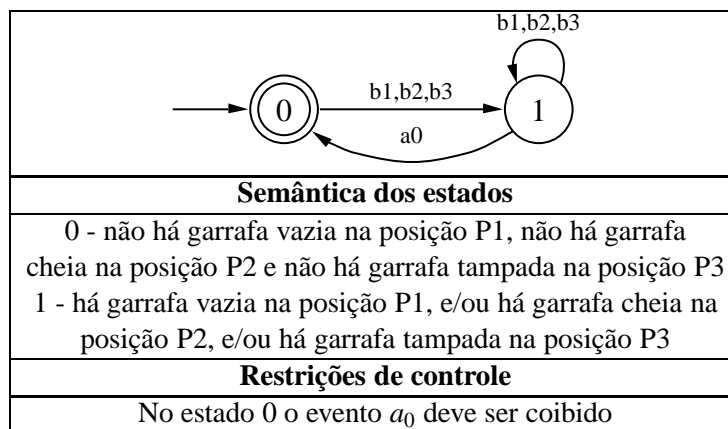
Conforme exposto no início desta seção, para que o sistema funcione adequadamente é necessário definir algumas especificações responsáveis pela descrição da operação desejada do sistema quando seus módulos operam paralelamente. Essas especificações visam coordenar a ação conjunta dos subsistemas possibilitando o perfeito funcionamento da célula, além de evitarem situações de bloqueio.

Cabe ressaltar que a marcação das especificações se deve à suposição de que serão consideradas tarefas completas apenas aquelas nas quais não exista nenhuma garrafa em processamento (todos os pallets vazios). As especificações genéricas para o problema de controle da CEBE são descritas na seqüência.

Tabela 3.4: Gerador G_3 (tampador)Tabela 3.5: Gerador G_4 (manipulador)

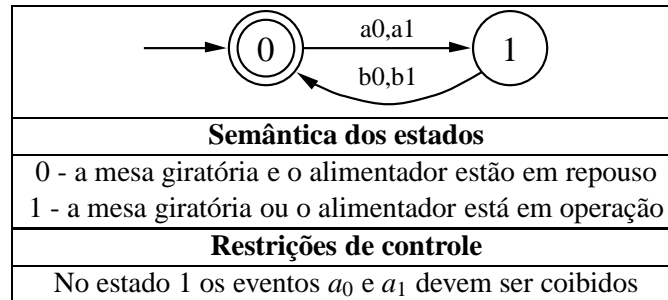
- Especificação Genérica a

Evita que a mesa gire sem necessidade, ou seja, sem garrafa vazia na posição P1, e/ou sem garrafa cheia na posição P2, e/ou sem garrafa tampada na posição P3 (Tabela 3.6).

Tabela 3.6: Gerador $Egen_a$ 

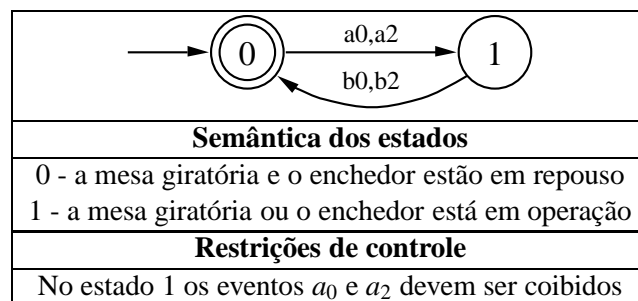
- Especificação Genérica b1

Realiza o intertravamento entre o módulo mesa giratória e o módulo alimentador (Tabela 3.7).

Tabela 3.7: Gerador $Egen_{b1}$ 

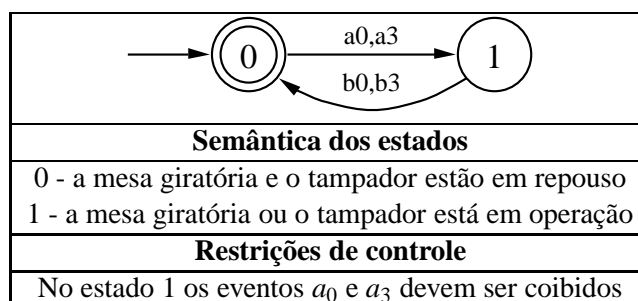
- Especificação Genérica b2

Realiza o intertravamento entre o módulo mesa giratória e o módulo enchedor (Tabela 3.8).

Tabela 3.8: Gerador $Egen_{b2}$ 

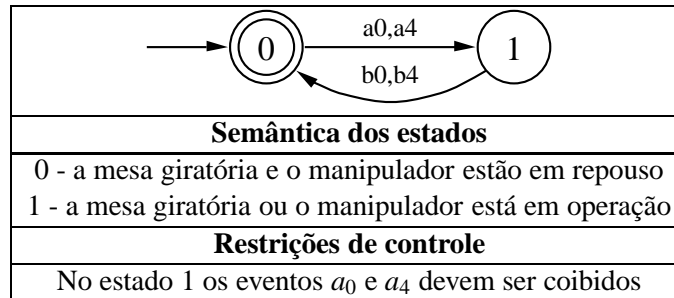
- Especificação Genérica b3

Realiza o intertravamento entre o módulo mesa giratória e o módulo tampador (Tabela 3.9).

Tabela 3.9: Gerador $Egen_{b3}$ 

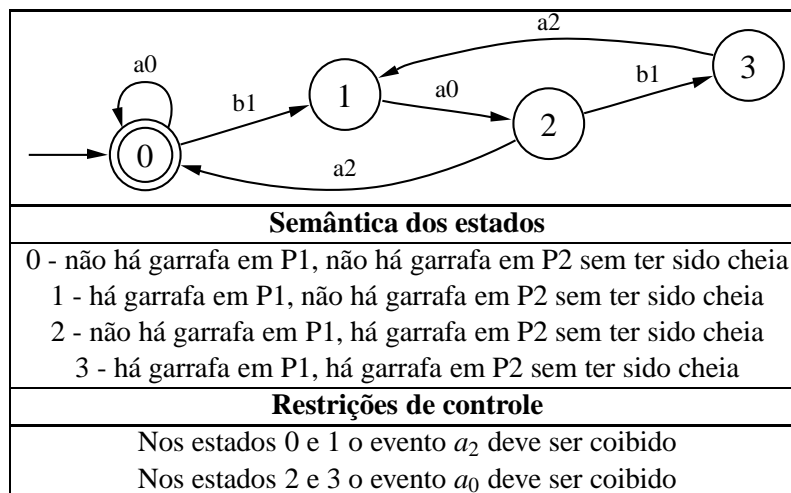
- Especificação Genérica b4

Realiza o intertravamento entre o módulo mesa giratória e o módulo manipulador (Tabela 3.10).

Tabela 3.10: Gerador $Egen_{b4}$ 

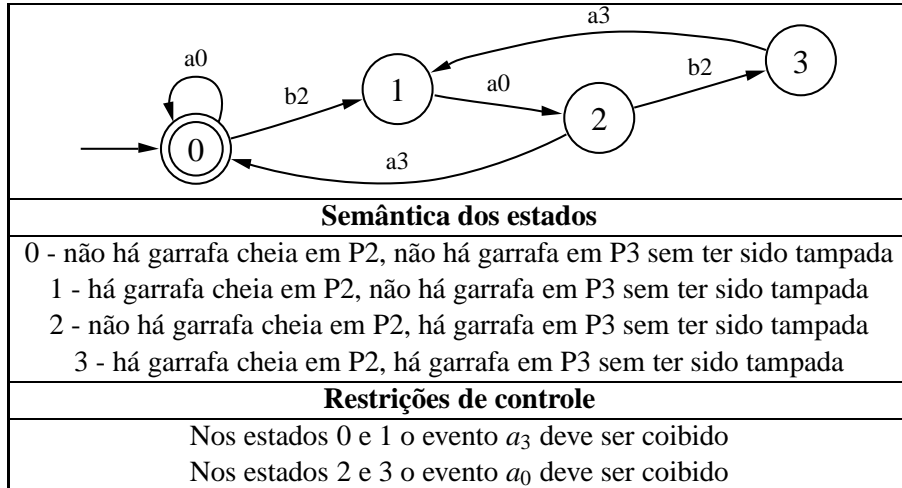
- Especificação Genérica c1

Evita sobrepor garrafas na posição P1, encher sem garrafa vazia na posição P2 e girar a mesa com garrafa vazia na posição P2 (Tabela 3.11).

Tabela 3.11: Gerador $Egen_{c1}$ 

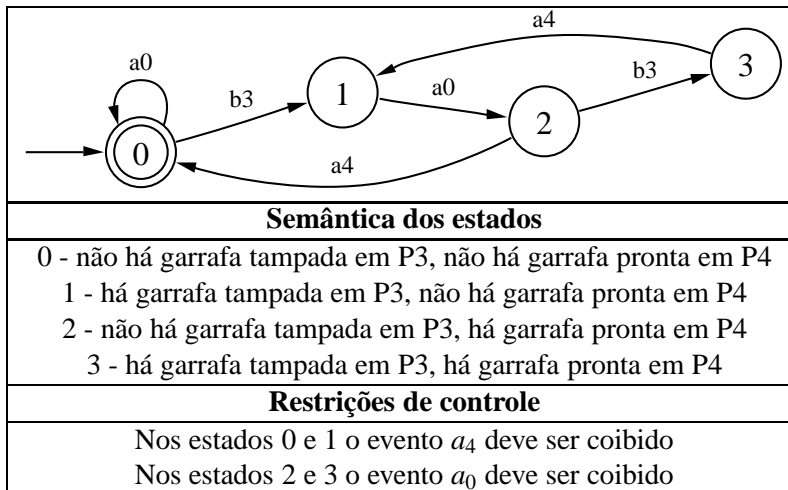
- Especificação Genérica c2

Evita encher garrafa já cheia na posição P2, tampar sem garrafa na posição P3 e girar a mesa com garrafa destampada na posição P3 (Tabela 3.12).

Tabela 3.12: Gerador $Egen_{c2}$ 

- Especificação Genérica c_3

Evita tampar garrafa já tampada na posição P3, retirada sem garrafa pronta na posição P4 e girar a mesa com garrafa pronta na posição P4 (Tabela 3.13).

Tabela 3.13: Gerador $Egen_{c3}$ 

3.2.3 Síntese de Supervisores Segundo Abordagem Modular Local

Todo o procedimento de síntese foi realizado com o auxílio das ferramentas computacionais GRAIL, disponibilizada pelo *Department of Computer Science - University of Western Ontario*, e CTCT, disponibilizada pelo *Systems Control Group - University of Toronto*. Ambos são softwares concebidos para pesquisa e aplicações que envolvem autômatos de estados finitos e linguagens regulares. A partir das especificações genéricas em consideração definem-se as plantas locais, que são formadas apenas pela composição dos subsistemas assíncronos que têm eventos em comum com a

especificação em questão. Para este projeto foram encontradas as plantas locais mostradas na Tabela 3.14.

Tabela 3.14: Plantas locais

Especificação Genérica	Planta Local
$Egen_a$	$Gloc_a = G_0 G_1 G_2 G_3$
$Egen_{b1}$	$Gloc_{b1} = G_0 G_1$
$Egen_{b2}$	$Gloc_{b2} = G_0 G_2$
$Egen_{b3}$	$Gloc_{b3} = G_0 G_3$
$Egen_{b4}$	$Gloc_{b4} = G_0 G_4$
$Egen_{c1}$	$Gloc_{c1} = G_0 G_1 G_2$
$Egen_{c2}$	$Gloc_{c2} = G_0 G_2 G_3$
$Egen_{c3}$	$Gloc_{c3} = G_0 G_3 G_4$

Calcula-se então as especificações locais ($Kloc_j$, $j = a, b1, b2, b3, b4, c1, c2, c3$), obtidas por meio da composição de cada especificação genérica com a sua respectiva planta local. Em seguida, encontra-se a componente trim destas especificações. No caso em estudo, todas as especificações são trim.

Finalmente são calculados os supervisores modulares locais, através do SupC (máxima linguagem controlável) da especificação local em relação à planta local. É realizado o teste de modularidade local, onde o resultado da composição entre todos os supervisores deve ser trim. Este teste indica a não existência de conflitos. Para este projeto esta condição foi satisfeita. Utilizando-se a ferramenta de redução de supervisores do software CTCT para controle supervisório foram encontrados os supervisores modulares locais mostrados na Figura 3.2.

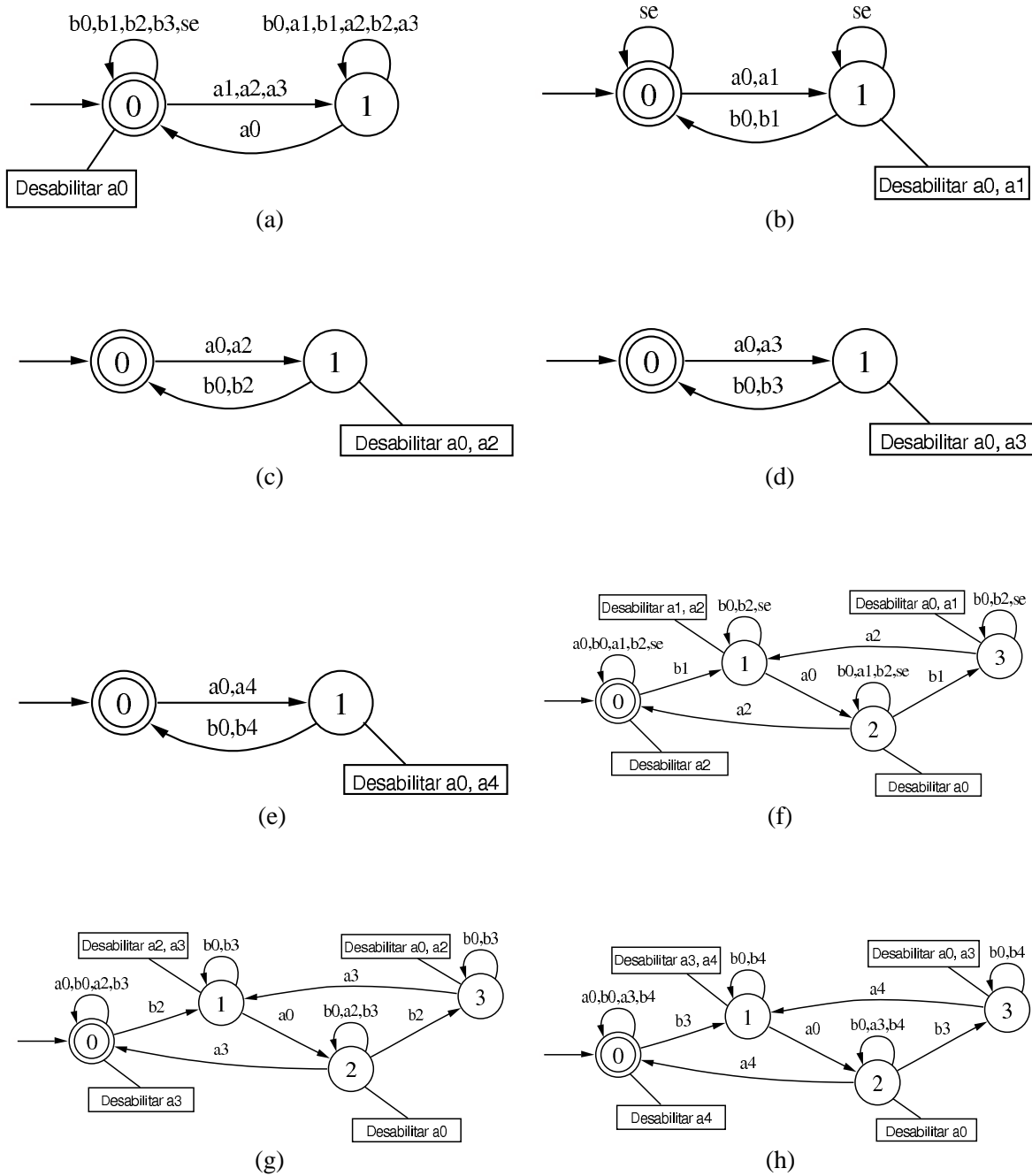


Figura 3.2: Supervisores modulares locais reduzidos: (a) $Sred_a$; (b) $Sred_{b_1}$; (c) $Sred_{b_2}$; (d) $Sred_{b_3}$; (e) $Sred_{b_4}$; (f) $Sred_{c_1}$; (g) $Sred_{c_2}$; (h) $Sred_{c_3}$

3.3 Conclusão do Capítulo

Este capítulo apresentou a modelagem do problema de controle tratado neste trabalho, que é a operação coordenada dos diversos subsistemas que compõem a CEBE. O sistema produto e os supervisores encontrados após o procedimento de síntese, assim como os componentes responsáveis pelo controle de baixo nível da planta, são as estruturas a serem distribuídas em um sistema de controle supervisório distribuído.

O capítulo que segue mostra como é possível implementar tais estruturas em um dispositivo de controle muito utilizado na indústria, o CLP. Isto é feito transformando-se estas estruturas encontradas na forma de autômatos em um programa escrito em uma linguagem que pode ser processada pelos CLPs.

Capítulo 4

Modelo para Implementação do Controle Supervisório

A partir dos procedimentos de síntese apresentados no Capítulo 2 é possível a obtenção de supervisores que atuam sobre o sistema para que seu funcionamento atenda a algumas especificações de projeto. Durante o processo de síntese dos supervisores muitas considerações e abstrações são feitas em relação ao SED real. Por essa razão, a implementação do controle supervisório em um dispositivo de controle como um Controlador Lógico Programável (CLP), por exemplo, não é trivial. Torna-se necessária a adoção de algum mecanismo de tradução da solução do problema do sistema modelado em uma solução que possa ser de fato implementada.

Algumas metodologias já foram desenvolvidas no intuito de resolver essa questão. A primeira seção deste capítulo apresenta uma revisão bibliográfica referente à problemática de implementação, destacando os principais pontos de cada um dos trabalhos estudados. Na segunda seção é apresentada em detalhes a estrutura genérica de implementação de supervisores modulares proposta por Queiroz [24] e formalizada por Vieira [35]. Este foi o modelo escolhido para a implementação distribuída do controle supervisório, objeto desta dissertação de mestrado. Na última seção mostra-se como essa estrutura é aplicada com sucesso na resolução de um problema de controle supervisório em uma célula flexível, a CEBE.

4.1 Algumas Abordagens da Problemática de Implementação

Segundo Fabian e Hellgren [10], o pequeno número de aplicações da TCS na indústria se deve à discrepância observada entre o supervisor abstrato e a sua implementação física, principalmente quando a implementação é baseada em CLPs. Fabian e Hellgren [10] abordam o problema da implementação do supervisor em CLPs, criando mecanismos de transformação do supervisor representado por uma máquina de estados finitos em Diagrama Escada [16]. Nesse processo, alguns problemas podem ser observados.

O primeiro deles seria sair do universo das máquinas de estados assíncronas baseadas em eventos para o universo dos CLPs síncronos baseados em sinais. Esta questão pode acarretar no efeito avalanche (um evento causa mais de uma transição, errôneamente) e no problema de simultaneidade de eventos, que devem ser evitados. O segundo problema destacado é a causalidade, uma vez que na prática nem todos os eventos são gerados pela planta, como é assumido pela TCS. A implementação no CLP deve incluir a geração de eventos e ser feita de modo que vários eventos não sejam gerados simultaneamente. A escolha é o terceiro problema levantado em [10]. Em algumas situações o supervisor pode ter que escolher entre algumas alternativas para a geração de um evento em um determinado estado. Para garantir que somente uma escolha é feita, a geração do evento deve ser realizada simultaneamente à transição. O último problema é a sincronização inexata, que possibilita que enquanto o programa executa alguma coisa possa acontecer que invalide as escolhas feitas pelo supervisor. Por isso, deve-se assegurar que a escolha do supervisor não seja invalidada por um evento da planta que ocorra durante o tempo de tomada da decisão.

Hellgren et al. [14] abordam o problema de implementação em CLP do controle supervisório modular. Além da sincronização da planta com o supervisor, um supervisor modular deve ser internamente sincronizado. Em [14] são utilizadas as redes de Petri devido ao seu inerente paralelismo e ao fato de que a linguagem SFC (*sequential function chart*) de programação de CLPs é baseada nas redes de Petri. Além disso, os autômatos de estados finitos podem ser considerados uma subclasse das redes de Petri. Hellgren et al. [14] discutem os quatro problemas levantados em [10] do ponto de vista da implementação do sistema de controle modular. Para que o problema de escolha [10] não ocorra, os supervisores modulares devem ser determinísticos. Para se obter esses supervisores determinísticos realiza-se a parametrização, na qual eventos e lugares em alguns supervisores modulares podem possuir alguns parâmetros livres que podem assumir valores especificados em outros supervisores modulares.

O efeito avalanche [10] não ocorre em sistemas implementados por meio de SFC, uma vez que SFC explicitamente evita as avalanches [14]. Quanto ao problema da causalidade [10], no caso da implementação do controle supervisório modular é necessário determinar qual dos supervisores modulares gerou um evento. Hellgren et al. [14] sugerem a utilização de outros agentes, chamados monitores de eventos [12], para a geração dos eventos de saída (gerados pelos supervisores). Para resolver o problema de sincronização inexata [10] deve-se utilizar o modelo de transição imediata/ação imediata [13], conseguido se os monitores de eventos forem executados antes dos supervisores modulares em cada ciclo de atualização do CLP.

Lauzon et al. [17] propõem uma metodologia de implementação baseada na utilização de leis de controle supervisório desenvolvidas de acordo com estruturas teóricas formais associadas ao estado da arte da tecnologia de controladores lógicos programáveis. Nesta metodologia a estratégia de controle é tratada como diversos subconjuntos de tamanho limitado compostos por múltiplos estados e transições que podem ser trocados em tempo real durante a operação da célula de manufatura de acordo com os eventos observados. É utilizado um sistema integrado onde um computador é empregado para a geração em tempo real das estratégias de controle e um CLP é empregado para executar estas estratégias. A utilização de CLPs é proposta para este fim porque estes dispositivos possuem funcionalidades para entrada/saída mais adequadas que os computadores pessoais (PCs) e além disso

são dispositivos de controle amplamente utilizados em aplicações industriais. O procedimento para a implementação do sistema de controle segundo esta metodologia é dado pelos seguintes passos [17]:

1. obtenção da estratégia de controle na forma de uma tabela de estado-transição padrão;
2. tradução da estratégia de controle por meio de um tradutor automático em código *ladder*;
3. realização do carregamento do código *ladder* no processador do CLP para o controle em tempo real da célula flexível de manufatura.

Nourelfath e Niel [21] implementam o controle supervisório modular em um sistema de manufatura automatizado experimental. Neste trabalho o formalismo de autômatos é utilizado para as etapas de modelagem e síntese enquanto que o Grafcet é utilizado para implementação nos CLPs. Os modelos em autômatos do comportamento básico da planta e dos supervisores são traduzidos em grafkets a serem implementados no CLP. Ressalta-se que o Grafcet foi utilizado por ser uma poderosa ferramenta que permite a especificação de grandes máquinas seqüenciais e que pode ser utilizada em CLPs. É também uma ferramenta de controle que possibilita que a ocorrência de eventos seja forçada, além de prover uma seqüência de saídas única e sem ambigüidades, dada uma seqüência de entradas. No entanto, os resultados observados no referido trabalho são independentes da ferramenta utilizada.

Os modelos em autômatos do comportamento básico da planta é traduzido em grafkets através da representação dos eventos controláveis por ações nos passos do grafcet e da associação dos eventos não controláveis com as transições dos passos. A tradução dos modelos em autômatos dos supervisores em grafkets é baseada no uso de condições booleanas que possibilitam que os eventos controláveis sejam desabilitados. Cada vez que um evento em um grafcet de uma planta tiver que ser desabilitado por um supervisor, ele é condicionado por uma variável booleana. Na proposta deste trabalho para a aplicação do controle supervisório, o supervisor conserva a sua função original de impedir a ocorrência de eventos como na teoria de Ramadge e Wonham. Isto porque, segundo Nourelfath e Niel [21], não se pode utilizar as estruturas nas quais os eventos são forçados quando se trata do controle supervisório modular. Assim sendo, os supervisores modulares e os modelos da planta são projetados e implementados para serem executados concorrentemente.

Charbonnier et al. [6] também utilizam o Grafcet para implementar o sistema de controle supervisório, justificando essa escolha pelas mesmas razões dadas no trabalho de Nourelfath e Niel [21]. Charbonnier et al. [6] mostram algumas das limitações das aproximações para implementação de sistemas de controle supervisório que consideram que a ocorrência de eventos seja forçada. Propõem desta forma o conceito de controle supervisionado, no qual controle e supervisão estão separados. Definem ainda o conceito de processo estendido, onde o processo a ser controlado acoplado ao controlador lógico podem ser vistos como um dispositivo único que evolui espontaneamente, sem que eventos necessitem ser forçados. A supervisão é então confinada à proibição de alguns dos eventos a serem gerados no processo estendido.

Queiroz [24] faz uma revisão bibliográfica de algumas outras metodologias e apresenta uma solução baseada na teoria de Controle Modular Local onde é preservada a modularidade tanto do

sistema físico quanto das especificações. Como o tema desta dissertação de mestrado diz respeito à implementação distribuída do controle supervisório, optou-se pela adoção desta metodologia, uma vez que a manutenção da característica intrinsecamente distribuída dos sistemas de manufatura reais deve ser preservada. Essa estrutura de implementação do sistema de controle supervisório é mostrada na seção seguinte.

Esta estrutura de implementação já foi utilizada em outros trabalhos [32] [20] com sucesso. Souto [32] utiliza esta estrutura para implementar o sistema de controle de uma unidade de potência hidráulica. Pode-se dizer que trata-se de um sistema de médio/grande porte, já que é composto por 33 módulos do sistema produto e 113 supervisores modulares locais. Neste trabalho os supervisores e os módulos do sistema produto foram implementados utilizando-se diagrama *ladder* enquanto que as seqüências operacionais foram implementadas por *Sequential Function Charts* (SFCs). A implementação bem sucedida da metodologia proposta por Queiroz [24] neste trabalho serviu como mais um incentivo para a sua adoção nesta dissertação de mestrado. Neto et al. [20] faz uma comparação entre as metodologias de projeto combinatório em Álgebra Booleana, Modelagem por Autômatos e a Teoria de Controle Supervisório Modular Local. São destacadas as vantagens e desvantagens quanto à facilidade de utilização dos métodos por parte dos projetistas, clareza do programa resultante, facilidade de criação do código, espaço físico de memória necessário para o programa, facilidade de manutenção, robustez e tempo de desenvolvimento.

4.2 Implementação do Controle Supervisório

A implementação de um supervisor ou conjunto de supervisores sintetizados a partir da Teoria de Controle Supervisório e suas extensões apresenta a seguinte dinâmica. Em um primeiro momento, tanto o(s) supervisor(es) quanto a planta encontram-se em seus estados iniciais. O procedimento de controle executado pelo(s) supervisor(es) consiste na observação da ocorrência de eventos no sistema físico, atualização do estado ativo do(s) supervisor(es) e desabilitação dos eventos controláveis associados a este estado ativo. Esse procedimento deve ser realizado toda vez que um novo evento ocorrer no sistema.

No entanto, essa abordagem da implementação do controle supervisório esbarra em algumas questões práticas que precisam ser levadas em conta em uma aplicação real. Em primeiro lugar deve-se considerar a possibilidade de eventos ocorrerem simultaneamente. Nos dispositivos de controle tradicionais, como os CLPs, a atualização dos dados é realizada em ciclos de duração finita. Quando dois ou mais eventos ocorrem durante o período de um ciclo, do ponto de vista do sistema de controle, esses eventos ocorreram em um mesmo instante, o que viola uma das assunções da TCS.

Deve-se levar em conta também que os eventos nem sempre ocorrem de maneira espontânea, sendo em muitos casos resultado de um comando proveniente do dispositivo de controle para a planta. Assim sendo, não é correto considerar a espontaneidade dos eventos como é feito na TCS. Por fim, outra questão importante diz respeito à abstração do sistema real realizada no procedimento de síntese. Muitas vezes, vários eventos e estados são desconsiderados, uma vez que não interferem diretamente no problema de controle e coordenação em questão. Porém, no momento da implementação,

esses eventos são relevantes, uma vez que fazem parte do sistema real. Um exemplo clássico desse problema diz respeito a situações onde é necessário levar em conta apenas os estados de operação e repouso de um equipamento. Nesses casos, normalmente, entre o início e o término da operação existem vários outros estados desconsiderados no processo de síntese que representam as atividades a serem desempenhadas pela máquina desde o momento em que é ligada até o término da operação. É essencial que a implementação do sistema de controle leve em conta a execução dessas atividades intermediárias.

A estrutura de implementação proposta por Queiroz [24] baseia-se em uma interface entre o sistema físico real e os supervisores modulares calculados pelos procedimentos de síntese (Figura 4.1).

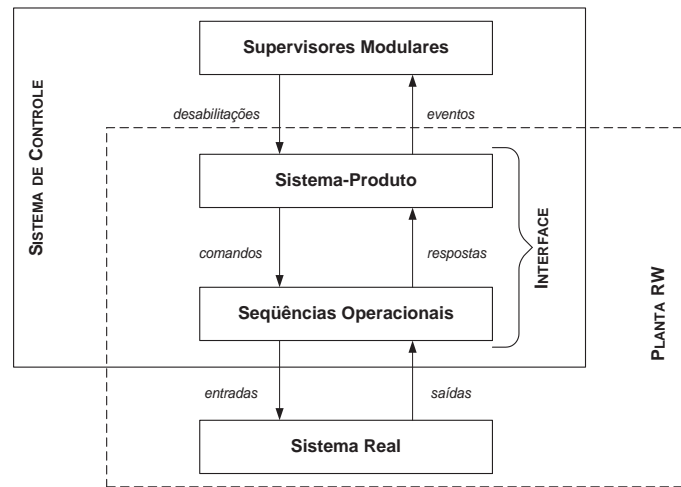


Figura 4.1: Esquema de implementação do sistema de controle

Nesse modelo, os eventos desabilitados pelos supervisores modulares determinam o funcionamento da planta abstrata definida pela representação por sistema-produto. O comportamento abstrato do sistema é traduzido em sinais de entrada e saída reais pelas seqüências operacionais, que definem as atividades intermediárias a serem desempenhadas por cada um dos subsistemas.

Nota-se que a modularidade tanto das especificações quanto do sistema físico é mantida nessa estrutura de implementação. A adição, retirada ou modificação de qualquer especificação ou subsistema não acarreta em mudanças dos módulos pré-existentes. A representação da planta por módulos do sistema produto significa, na maioria dos casos, uma redução extremamente significativa da quantidade de memória necessária para representar o sistema. Quando o sistema é representado pela composição síncrona de todos os subsistemas assíncronos, como na TCS de Ramadge e Wonham, a quantidade de memória necessária para a implementação em um CLP, por exemplo, pode muitas vezes tornar a implementação do sistema de controle proibitiva. Esse modelo de implementação permite ainda que as abstrações do sistema real utilizadas durante o processo de síntese sejam viáveis, já que por meio das seqüências operacionais possibilita a realização completa das tarefas anteriormente simplificadas [35].

4.2.1 Formalização do Modelo de Implementação

No intuito de facilitar a geração automática do código a ser implementado no dispositivo de controle, Vieira [35] [36] [37] formaliza o modelo de Queiroz [24]. A partir disso, são estabelecidas as variáveis a serem utilizadas no programa de controle e suas inter-relações.

Considere o conjunto de n subsistemas assíncronos entre si que constituem um sistema produto representados pelos geradores $G_i = (X_i, \Sigma_i, \delta_i, x_{0_i}, X_{m_i})$ com $i = \{1, \dots, n\}$. Cada subalfabeto Σ_i é particionado em eventos controláveis $\Sigma_{ci} \subseteq \Sigma_i$ e eventos não controláveis $\Sigma_{ui} \subseteq \Sigma_i$. O alfabeto do sistema físico completo é dado por $\Sigma = \cup_{i=1}^n \Sigma_i$. Em primeiro lugar, para cada subsistema, realiza-se um mapeamento que define quatro funções bijetoras que relacionam [35]:

1. a cada evento controlável de Σ_{ci} uma *desabilitação*, $fd_i : \Sigma_{ci} \rightarrow \Phi_i$;
2. a cada desabilitação uma *negação de desabilitação*, $fnd_i : \Phi_i \rightarrow \Pi_{ndi}$;
3. a cada ocorrência de evento controlável em G_i um *comando*, $fc_i : X_i \times \Sigma_{ci} \rightarrow \Delta_i$;
4. a cada ocorrência de evento não controlável em G_i uma *resposta*, $fr_i : X_i \times \Sigma_{ui} \rightarrow \Pi_{ri}$.

onde:

- Φ_i é o alfabeto de desabilitações de eventos controláveis;
- Π_{ndi} é o alfabeto de negações de desabilitações de eventos controláveis;
- Δ_i é o alfabeto de comandos;
- Π_{ri} é o alfabeto de respostas.

Estas funções são bijetoras já que cada elemento do domínio da função tem imagens distintas e cada elemento do contra-domínio da função é a imagem de algum elemento do domínio.

O nível Supervisores Modulares (SM) da estrutura mostrada na Figura 4.1 é constituído pelos supervisores modulares locais reduzidos, obtidos conforme o procedimento de síntese mostrado na seção 2.4.2. Esses supervisores são representados por um conjunto de autômatos ($S = \{S_{red_j} | j = \{1, \dots, m\}\}$) e suas respectivas funções de desabilitação de eventos ψ_{red_j} .

É possível construir a função $\Psi_j : \Phi \rightarrow 2^{Q_j}$, onde Q_j é o conjunto de estados do supervisor S_{red_j} , que relaciona os estados do gerador S_{red_j} às desabilitações que devem ser ativadas em cada um deles. Ela é dada por:

$$(\forall \alpha d \in \Phi)(\forall i \in I) \Psi_j(\alpha d) = \{q \in Q_j | \alpha d = fd_i(\alpha) \text{ e } \alpha \in \psi_{red_j}(q)\} \quad (4.1)$$

Isso significa que quando a função Ψ_{red_j} indica que um evento relacionado a uma determinada desabilitação de evento controlável deve ser desabilitado em um certo estado, este estado deve ser acrescentado à relação de estados em que esta desabilitação deve ser ativada.

A função $\Psi_{total} : \Phi \rightarrow 2^Q$, onde Q é o conjunto de estados de todos os supervisores, especifica em quais estados de Q cada uma das desabilitações deve ser ativada. Essa função, baseada no fato de que um evento controlável deve ser desabilitado sempre que algum dos supervisores modulares determinarem a desabilitação (seção 2.4.2), é dada por:

$$(\forall \alpha d \in \Phi) \Psi_{total}(\alpha d) = \bigcup_{j=1}^m \Psi_j(\alpha d) \quad (4.2)$$

No nível Sistema Produto (SP) a cada um dos autômatos G_i está relacionado um autômato g_i na forma de máquina de Mealy. Neste tipo de representação, que conta com algumas das características já presentes no autômato convencional, são associados ao autômato um alfabeto de saída e uma função de saída. Esta função é responsável por determinar quais eventos do alfabeto de saída são ativados quando ocorre uma transição de estados no autômato. A representação gráfica desta categoria de autômatos é semelhante à dos autômatos convencionais. Apenas as etiquetas dos arcos que ligam dois estados são alteradas, já que além do evento do alfabeto de entrada que causa a transição ela também contém o subconjunto de eventos do alfabeto de saída que são ativados durante essa transição. Os eventos de entrada e de saída são separados por uma barra, com os primeiros ficando à esquerda. No modelo de implementação em discussão, $g_i = (\Pi_i, 2^{\Gamma_i}, X_i, x_{0_i}, \xi_i, \omega_i)$ onde:

- Π_i é o alfabeto de entrada, sendo que $\Pi_i = \Pi_{ndi} \cup \Pi_{ri}$;
- 2^{Γ_i} é o alfabeto de saída, sendo que $\Gamma_i = \Sigma_i \cup \Delta_i$;
- ξ_i é a função de transição de estados, com $\xi_i : X_i \times \Pi_i \rightarrow X_i$;
- ω_i é a função de saída, com $\omega_i : X_i \times \Pi_i \rightarrow 2^{\Gamma_i}$;
- X_i e x_{0_i} correspondem respectivamente ao mesmo conjunto de estados e estado inicial de G_i .

A função de transição de estados ξ_i relacionada ao autômato g_i é dependente do tipo de evento ocorrido: controlável ou não controlável [36]. Esta função é dada por:

- $\xi_i(x, -\alpha d) = x'$ sempre que $\delta_i(x, \alpha) = x'$, onde α é um evento controlável e $-\alpha d = fnd_i(fd_i(\alpha))$;
- $\xi_i(x, rpl\beta) = x'$ sempre que $\delta_i(x, \beta) = x'$, onde β é um evento não controlável e $rpl\beta = fr_i(x, \beta)$.

Em suma, esta função de transição modificada substitui a ocorrência de cada evento controlável pela negação da desabilitação correspondente e a ocorrência de cada evento não controlável pela resposta correspondente.

A função de saída ω_i é definida por:

- $\omega_i(x, \neg\alpha d) = \{\alpha, cmd\alpha\}$, onde $\neg\alpha d = fnd_i(fd_i(\alpha))$ e $cmd\alpha = fc_i(q, \alpha)$;
- $\omega_i(x, rpl\beta) = \{\beta\}$, onde $rpl\beta = fr_i(x, \beta)$.

Em palavras, esta função de saída faz com que a transição de estado em g_i ocasionada por uma negação de desabilitação gere o evento e o comando correspondentes enquanto que a transição ocasionada por uma resposta gere apenas o evento correspondente, conforme ilustrado no Exemplo 4.1.

Exemplo 4.1 Considere a representação do comportamento livre do subsistema mesa giratória da CEBE, mostrada na Tabela 3.1 e reproduzida aqui na Figura 4.2. O evento $a0$ é controlável e o evento $b0$ é não controlável. Realiza-se o seguinte mapeamento:

$$fd_0(a0) = a0d;$$

$$fnd_0(a0d) = Na0d;$$

$$fc_0(0, a0) = cmda0;$$

$$fr_0(1, b0) = rplb0.$$

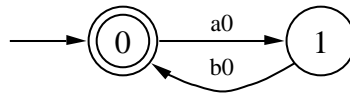


Figura 4.2: Autômato que representa o comportamento livre da mesa giratória

A função de transição de estados e a função de saída da representação por máquina de Mealy do módulo do sistema produto associado à mesa giratória são definidas por:

$$\xi_0(0, Na0d) = 1;$$

$$\xi_0(1, rplb0) = 0;$$

$$\omega_0(0, Na0d) = \{a0, cmda0\};$$

$$\omega_0(1, rplb0) = \{b0\}.$$

A Figura 4.3 mostra a representação gráfica, por máquina de Mealy, do módulo do sistema produto do subsistema mesa giratória.

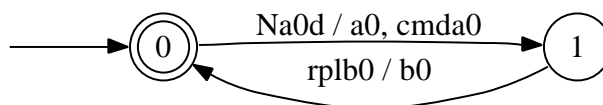


Figura 4.3: Módulo do sistema produto da mesa giratória representado por máquina de Mealy

Alguns aspectos ligados à inter-relação existente entre os subsistemas físicos devem ser considerados. Para cada módulo do sistema produto g_k pode-se relacionar todos os módulos do sistema produto g_i tal que g_k e g_i fazem parte simultaneamente de alguma planta local G_{loc_j} . A função $floc : G \rightarrow 2^G$ descreve essa relação:

$$(\forall g_k \in G)(\forall j \in J) floc(g_k) = \{g_i \in G_{loc_j} \mid g_k \in G_{loc_j}\} \quad (4.3)$$

São definidas ainda uma função que sinaliza a evolução de um módulo do sistema produto e outra que é responsável por desabilitar qualquer transição de estado de um módulo do sistema produto. Estas funções bijetoras são dadas por $fg : G \rightarrow \Xi$ e $fgd : G \rightarrow \Xi_d$ onde G é o conjunto de módulos do sistema produto, Ξ é o conjunto de sinalizações de evolução de módulos do sistema produto e Ξ_d é o conjunto de desabilitações de evolução de módulos do sistema produto.

O nível Seqüências Operacionais (SO) é uma interface necessária entre o sistema modelado implementado no nível SP e o sistema físico, uma vez que durante o processo de modelagem são realizadas diversas abstrações sobre o funcionamento real do sistema. Nesse nível, os comandos provenientes do nível SP conduzem à execução de diversas tarefas específicas do funcionamento do subsistema. Durante essas tarefas, sinais são trocados com o sistema real por meio dos canais de entrada e saída do dispositivo de controle. As SO são ainda responsáveis por enviarem respostas ao nível SP, refletindo a ocorrência dos eventos não-controláveis na planta.

Cada subsistema físico está relacionado a uma seqüência operacional que define o seu funcionamento. Este conjunto de seqüências operacionais é dado por $O = \{o_i \mid i = 1, \dots, n\}$.

Duas condições devem ser satisfeitas para o correto funcionamento do sistema de controle. Em primeiro lugar, o sistema produto só pode evoluir devido à ocorrência de um evento controlável se todos os supervisores que exercem ação de desabilitação sobre esse evento estiverem atualizados e se nenhum deles estiver desabilitando este evento. Além disso, cada supervisor só pode evoluir pela ocorrência de um, e somente um, evento por ciclo de atualização do dispositivo de controle. Isso faz com que, do ponto de vista do sistema de controle, não haja simultaneidade de eventos.

Na realidade é possível que eventos ocorram simultaneamente. Isso ocorre já que podem estar habilitados um ou mais eventos controláveis assim como uma ou mais respostas podem estar ativas. Para que as condições acima sejam obedecidas, faz-se necessário a criação de mecanismos que impossibilitem essa ocorrência simultânea de eventos. Assim sendo, sempre que um módulo do sistema produto g_k evoluir devido à ocorrência de um evento, todos os módulos do sistema produto $g_i \in floc(g_k)$ devem ter sua evolução desabilitada. Além disso, a evolução de g_k força a atualização de estado de todo supervisor S_j tal que $g_k \in G_{loc_j}$. A desabilitação de evolução de cada um dos módulos do sistema produto $g_i \in floc(g_k)$, com $g_i \in G_{loc_j}$ deve se manter ativa até a atualização do supervisor S_j correspondente. g_k também não pode evoluir até que todo supervisor S_j tal que $g_k \in G_{loc_j}$ seja atualizado.

4.2.2 Implementação no Dispositivo de Controle

Após representar o sistema de controle seguindo a formalização mostrada na seção 4.2.1 deve-se passar ao último passo do processo de implementação, que consiste na transformação do sistema de controle em um programa cuja forma seja adequada a um dispositivo de controle, no caso, um CLP. A forma de implementação escolhida é por meio de Diagrama Escada (do inglês, *ladder*) [16], dentre outras razões, por se tratar de uma linguagem de programação de CLPs muito utilizada e difundida na atualidade.

A etapa do processo de implementação mostrada na seção 4.2.1 se justifica por estabelecer as variáveis a serem empregadas no programa de controle e suas inter-relações [35]. Desta maneira, torna-se possível a construção de compiladores para geração automática de código, o que reduz consideravelmente o tempo despendido com programação no escopo de um projeto de automação. Um bit da memória interna do CLP deve ser alocado para cada uma das seguintes variáveis:

- estados dos geradores que representam os supervisores;
- estados dos autômatos que representam cada um dos módulos do sistema produto;
- eventos;
- desabilitações de evento controlável;
- comandos;
- respostas;
- sinalizações de evolução de módulo do sistema produto;
- desabilitações de evolução de módulo do sistema produto.

4.2.2.1 Nível dos Supervisores Modulares

Neste nível é necessário implementar estruturas que realizem as transições de estado dos geradores que representam os supervisores modulares assim como a ativação das desabilitações dos eventos controláveis.

Considere a função de transição de estados δ_j do supervisor modular S_j . Pode-se relacionar um estado de origem (x) e um evento (α) a um estado destino (x'), relação esta representada pela tripla ordenada (x, α, x') . Para a implementação do gerador que realiza o supervisor S_j , implementa-se cada uma das triplas ordenadas dadas pela função δ_j conforme mostrado na Figura 4.4.

Quando ocorrem nos supervisores os chamados autolaços (o estado de origem é o estado de destino) não é necessário implementar a estrutura da Figura 4.4, uma vez que o estado ativo do supervisor não se altera, mesmo com a ocorrência do evento do autolaço.

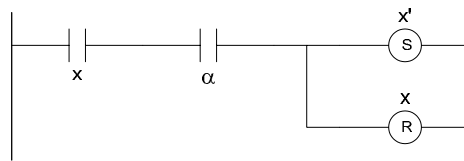


Figura 4.4: Estrutura de implementação de transição do supervisor

Considere agora a desabilitação de evento controlável αd e o conjunto de estados de todos os supervisores modulares $\{x_1, x_2, \dots, x_y\}$ nos quais esta desabilitação deve ser ativada. Para cada desabilitação e conjunto de estados correspondente deve ser implementada a estrutura mostrada na Figura 4.5. Essa estrutura faz com que caso um ou mais supervisores modulares esteja desabilitando algum evento controlável, o sistema de controle como um todo desabilite esse evento.

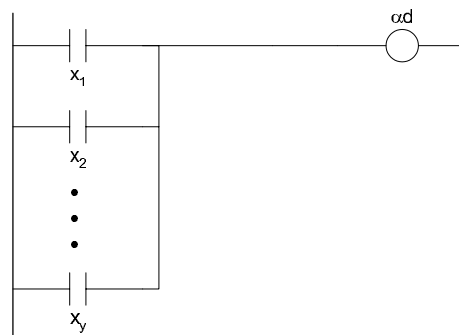


Figura 4.5: Estrutura de implementação de ativação de desabilitação de evento controlável

4.2.2.2 Nível do Sistema Produto

Considere a função $floc$ definida pela equação 4.3. Cada uma das n -uplas desta função pode ser representada por $(g_k, \{g_1, g_2, \dots, g_x\})$. Visto que o módulo do sistema produto relacionado com g_k deve ter sua evolução desabilitada caso um dos módulos em $\{g_1, g_2, \dots, g_x\}$ evolua, esta mesma n -upla pode ser dada por $(g_k d, \{g_1 evt, g_2 evt, \dots, g_x evt\})$, onde $g_i evt$ indica que o módulo g_i evoluiu e $g_i d$ estabelece a desabilitação de evolução deste módulo. Cada uma destas n -uplas deve ser implementada conforme mostrado na Figura 4.6.

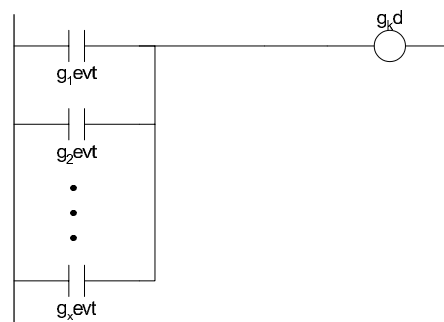


Figura 4.6: Estrutura de implementação de ativação de desabilitação de evolução de módulo do sistema produto

A implementação do código relacionado a uma resposta (rp/β) é diferente do relacionado a uma

negação de desabilitação de evento controlável ($\neg\alpha d$). Para o primeiro caso, considere a função de transição de estados ξ_i onde pode-se representar por $(x, rpl\beta, x')$ cada tripla ordenada da função que relaciona um estado de origem (x) e uma resposta ($rpl\beta$) a um estado de destino (x'). Durante essa transição o evento não-controlável β deve ser gerado, conforme definido pela função de saída ω_i . Para todo módulo do sistema produto, implementar a estrutura mostrada na Figura 4.7 para cada tripla ordenada da função de transição de estados ξ_i relacionada à uma transição ocasionada por uma resposta.

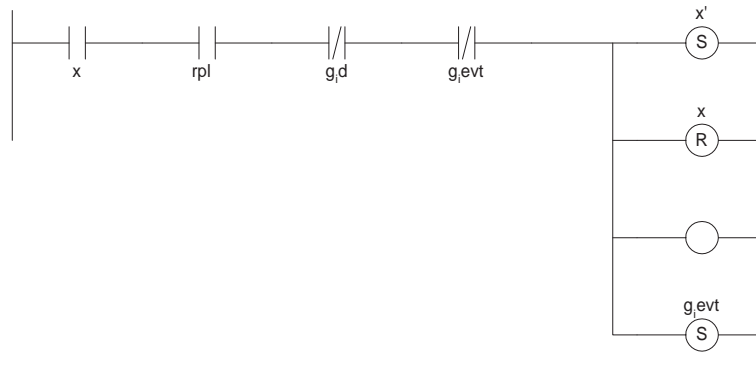


Figura 4.7: Estrutura de implementação de transição relacionada a uma resposta

Já para uma negação de desabilitação de evento controlável, a tripla ordenada associada à função de transição de estados ξ_i é dada por $(x, \neg\alpha d, x')$, que relaciona um estado de origem (x) e uma negação de desabilitação de evento controlável ($\neg\alpha d$) a um estado de destino (x'). Durante esta transição são gerados o evento controlável α e o comando $cmd\alpha$ conforme a função de saída ω_i . Para todo módulo do sistema produto, implementar a estrutura mostrada na Figura 4.8 para cada tripla ordenada da função de transição de estados ξ_i relacionada à uma transição ocasionada por uma negação de desabilitação de evento controlável.

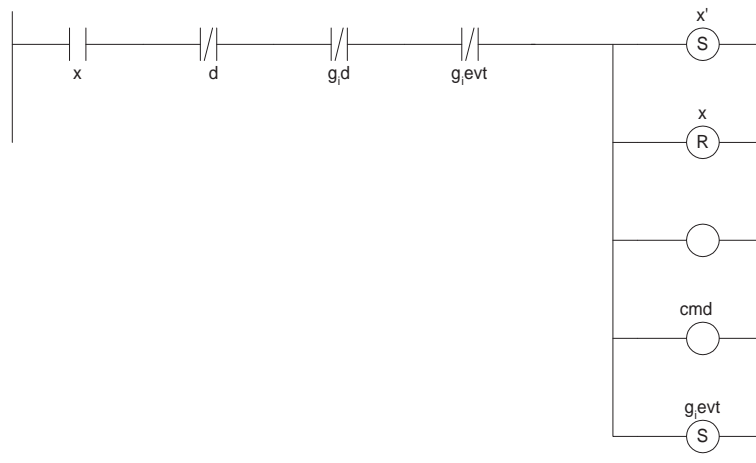


Figura 4.8: Estrutura de implementação de transição relacionada a uma negação de desabilitação de evento controlável

Cabe ressaltar que a estrutura que desabilita a evolução de um autômato g_i (Figura 4.6) deve preceder imediatamente todas as implementações das transições deste autômato (Figuras 4.7 e 4.8). A desabilitação de evolução de um autômato deve ser desativada depois que todas as suas transições de

estados tiverem sido processadas. Além disso, as variáveis que indicam a evolução dos autômatos devem ser desativadas depois que as transições de estados de todos os autômatos tiverem sido processadas.

Nas estruturas mostradas nas Figuras 4.7 e 4.8 percebe-se que quando ocorre uma transição de estado em um dos autômatos g_i , a sinalização de evolução do mesmo é ativada com retenção. Vieira [35] considera que para que uma transição ocorra em um autômato, é necessário também que apenas a desabilitação de evolução do autômato não esteja ativada. No entanto, caso o próprio autômato evolua em um determinado ciclo do CLP, a estrutura, tal como foi proposta, não impede a ocorrência de uma outra transição no mesmo autômato. Por isso, propõe-se uma outra condição para a ocorrência de uma transição: que a evolução do autômato em questão ($g_i\text{evt}$) não esteja ativada.

A ordem na qual as estruturas das Figuras 4.7 e 4.8 são colocadas no programa interfere na prioridade de uma transição sobre as demais. A transição que for processada primeiro terá maior prioridade sobre aquelas que a sucedem, visto que caso ela ocorra nenhuma outra transição poderá ocorrer no mesmo autômato no mesmo ciclo. Assim sendo, deve-se priorizar o tratamento da ocorrência das transições relacionadas às respostas, uma vez que elas estão diretamente relacionadas aos eventos não-controláveis, ou seja, aos eventos que ocorrem espontaneamente em algum instante de tempo desconhecido e não podem ser coibidos.

4.2.2.3 Nível das Sequências Operacionais

A implementação da lógica relativa às seqüências operacionais, assim como as demais já apresentadas, pode ser realizada por meio de diagrama escada. Devido à especificidade deste nível não existe um procedimento padrão para projeto. A implementação neste caso irá depender inteiramente das atividades a serem desempenhadas pelo subsistema a que diz respeito.

Neste nível é muito comum que outras formas de implementação sejam utilizadas visto que tais seqüências muitas vezes já são projetadas e otimizadas pelos próprios fabricantes do equipamento. Um exemplo tradicional é a implementação da lógica das seqüências operacionais em placas de circuito integrado embutidas no próprio equipamento. Neste casos utiliza-se muitas vezes programação em linguagens como C, C++ ou Java. Outras vezes pode-se programar as seqüências operacionais em alguma outra linguagem de programação de CLPs que seja mais conveniente ao programador e à situação.

Essa diversidade de possibilidades de programação confere uma flexibilidade ao nível Sequências Operacionais bastante interessante visto que possibilita a integração dessas seqüências aos demais níveis, independentemente da forma com que foram programadas.

4.2.2.4 Inicialização e Ordenação do Programa de Controle

Segundo Vieira [35], na inicialização do programa que implementa o sistema de controle é necessário:

- desativar todas as variáveis internas do CLP;
- desativar todas as saídas de sinal;
- ativar a variável relacionada ao estado inicial de todo gerador S'_j ;
- ativar as variáveis relacionadas ao estado inicial de todo autômato g_i ;
- ativar as variáveis relacionadas ao passo inicial de todo autômato g_i .

Recomenda ainda que a seqüência de implementação do programa seja a seguinte:

- inicialização do programa;
- estrutura dos geradores que representam os supervisores;
- estrutura de ativação de desabilitação de eventos controláveis;
- estrutura dos autômatos do sistema produto;
- desativação de desabilitações de evolução dos módulos do sistema produto;
- estrutura das seqüências operacionais.

4.3 Aplicação do modelo à CEBE

Esta seção mostra a implementação do sistema de controle projetado para resolver o problema de controle apresentado na seção 3.2. Mostra-se através desse exemplo como é possível automatizar a geração de código em linguagem de programação de CLPs, por meio da utilização de um compilador que transforma autômatos em lista de instruções. Acredita-se ser este um primeiro passo para o desenvolvimento de compiladores de ampla aplicabilidade, para produção de código para implementação de sistemas de controle supervisório para SEDs em dispositivos de controle, como os CLPs.

O software utilizado ainda apresenta algumas limitações, como a geração de código apenas para alguns CLPs específicos. Porém, para este trabalho, foi de grande valia, reduzindo bastante o tempo gasto com o desenvolvimento do programa. O objetivo desta seção é mostrar como cada um dos níveis de implementação (Supervisores Modulares, Sistema Produto e Seqüências Operacionais) foram implementados.

4.3.1 Supervisores Modulares e Módulos do Sistema Produto

Para a implementação do sistema de controle sintetizado na seção 3.2 (supervisores e sistema produto) foi utilizado o software Conversor de Grail para Lista de Instruções (CGLI), desenvolvido no Laboratório de Sistemas Hidráulicos e Pneumáticos (LASHIP) da Universidade Federal de Santa

Catarina [19], que converte a representação por sistema produto e os supervisores descritos por autômatos (no formato aceito pelo programa Grail) em uma lista de instruções que pode ser implementada em um CLP. Algumas funcionalidades da ferramenta computacional CGLI são mostradas a seguir.

Na primeira tela (Figura 4.9) o CLP a ser utilizado é selecionado, assim como as implementações desejadas.

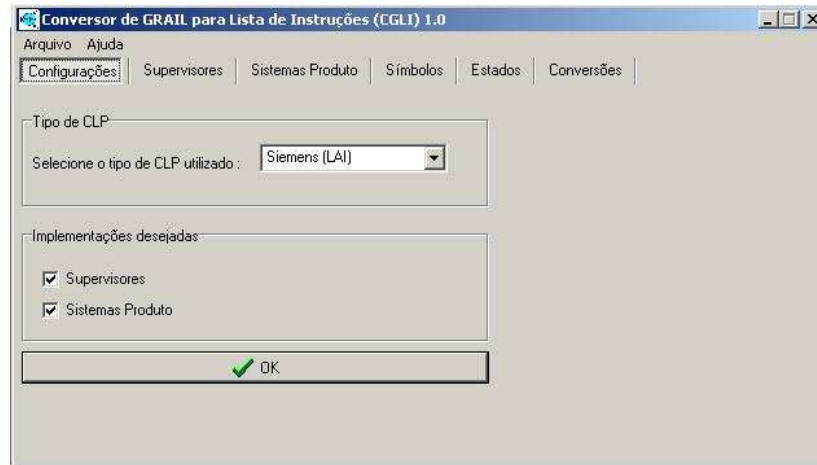


Figura 4.9: Tela *Configurações* do CGLI [19]

Nas telas *Supervisores* e *Sistemas Produto* são fornecidos os caminhos dos arquivos no formato .grail que contêm os autômatos a serem implementados, como mostrado na Figura 4.10.



Figura 4.10: Tela *Supervisores* do CGLI [19]

Na tela *Símbolos (Estados)* são listados todos os eventos (estados) e é possível determinar um nome para esses eventos (estados) no CLP (Figura 4.11). Além disso, quando os eventos são controláveis, deve-se fornecer um nome para a desabilitação deste evento e outro nome para o comando referente a este evento. Quando os eventos são não controláveis deve-se fornecer um nome para as respostas relativas a eles. Estes nomes serão os nomes das variáveis correspondentes no programa gerado.

Por fim, na tela *Conversões*, são gerados o código em lista de instruções do CLP e um arquivo de

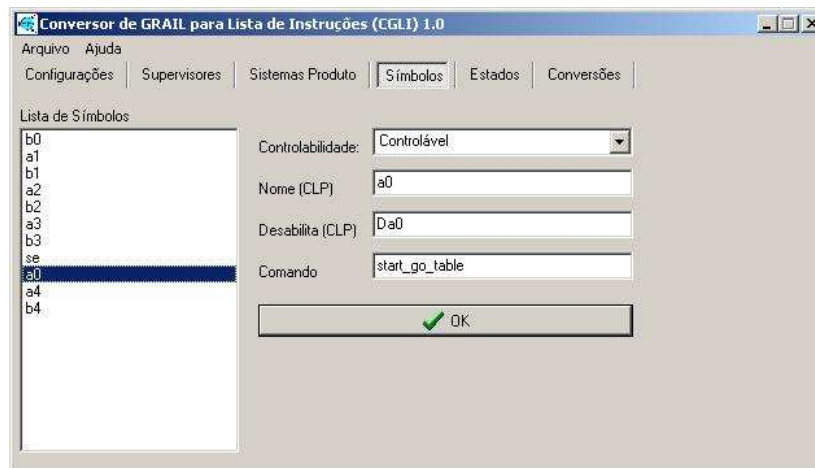


Figura 4.11: Tela *Símbolos* do CGLI [19]

texto com a declaração das variáveis necessárias. A título de ilustração, a Tabela 4.1 mostra a lista de instruções com os 6 primeiros grupos de instruções (redes) gerados pelo CGLI.

Tabela 4.1: Lista de instruções gerada pelo CGLI (apenas as 6 redes iniciais)

```

SUBROUTINE_BLOCK Super:SBR8
TITLE=SUBROUTINE COMMENTS
BEGIN
NETWORK 1
// Inicialização dos estados iniciais dos Supervisores
LD restart_automata
S Sup1E0, 1
S Sup2E0, 1
S Sup3E0, 1
S Sup4E0, 1
S Sup5E0, 1
S Sup6E0, 1
S Sup7E0, 1
S Sup8E0, 1
NETWORK 2
// Inicialização dos estados não iniciais dos Supervisores
LD restart_automata
R Sup1E1, 1
R Sup2E1, 1
R Sup3E1, 1
R Sup4E1, 1
R Sup5E1, 1
R Sup6E1, 1
R Sup6E2, 1
R Sup6E3, 1
R Sup7E1, 1
R Sup7E2, 1

```

Tabela 4.1: Lista de instruções gerada pelo CGLI (apenas as 6 redes iniciais)

```

R  Sup7E3, 1
R  Sup8E1, 1
R  Sup8E2, 1
R  Sup8E3, 1
NETWORK 3
// Inicialização dos estados iniciais dos Sistemas Produto
LD  restart_automata
S  SP1E0, 1
S  SP2E0, 1
S  SP3E0, 1
S  SP4E0, 1
S  SP5E0, 1
NETWORK 4
// Inicialização dos estados não iniciais dos Sistemas Produto
LD  restart_automata
R  SP1E1, 1
R  SP2E1, 1
R  SP2E2, 1
R  SP2E3, 1
R  SP3E1, 1
R  SP4E1, 1
R  SP5E1, 1
R  restart_automata, 1
NETWORK 5
// Máquina de estados Supervisor
LD  Sup1E1
A  a0
R  Sup1E1, 1
S  Sup1E0, 1
NETWORK 6
// Máquina de estados Supervisor
LD  Sup1E0
A  a1
LD  Sup1E0
A  a2
LD  Sup1E0
A  a3
OLD
OLD
R  Sup1E0, 1
S  Sup1E1, 1

```

Considere o supervisor $Sred_a$ calculado na seção 3.2.3, que é reproduzido na Figura 4.12.

Este supervisor apresenta a estrutura em diagrama ladder mostrada na Figura 4.13, que é equiva-

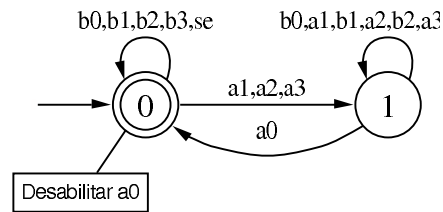


Figura 4.12: Supervisor modular local reduzido $Sred_a$

lente àquela implementada pelas redes 5 e 6 da Tabela 4.1. A estrutura dos demais supervisores e dos módulos do sistema produto é construída de maneira análoga, segundo a metodologia apresentada na seção 4.2.2.

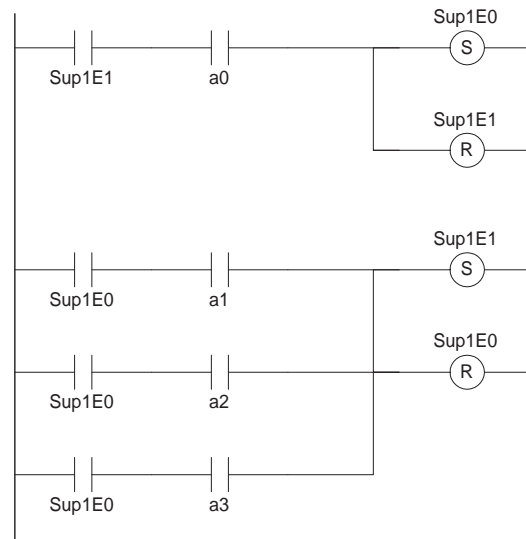


Figura 4.13: Diagrama ladder do supervisor $Sred_a$

A implementação da lógica de controle foi realizada em um CLP Siemens da família S7-200. O arquivo gerado pelo CGLI com a lista de instruções foi importado pelo software de desenvolvimento do CLP (STEP 7 - Micro/WIN 32) e inserido como uma subrotina da lógica padrão do CLP. Além disso, o arquivo com a declaração das variáveis gerado pelo CGLI foi copiado para a tabela de variáveis do programa base do CLP.

4.3.2 Seqüências Operacionais

Em muitos casos práticos, as seqüências operacionais relacionadas aos diversos módulos que compõem o sistema já encontram-se implementadas e otimizadas. Muitas vezes a implementação destas seqüências já é disponibilizada embutida no próprio equipamento pelo fabricante. Há situações onde simplesmente as seqüências operacionais já foram previamente implementadas, não havendo necessidade de reimplementação, como é o caso da CEBE.

4.4 Conclusão do Capítulo

Este capítulo mostrou como é possível implementar os supervisores, módulos do sistema produto e seqüências operacionais em um dispositivo de controle [24] [35]. Observa-se entretanto que em alguns casos surge naturalmente a necessidade de se implementar o sistema de controle distribuído, ou seja, utilizando-se mais de um dispositivo de controle. Um exemplo claro desta situação é o das seqüências operacionais da CEBE, que já estavam previamente implementadas e deveriam ser reaproveitadas. A tarefa de distribuição do sistema de controle não é trivial, uma vez que aspectos como a sincronização dos dispositivos de controle devem ser analisados para que a seqüência com que os diversos blocos que compõem o sistema de controle são processados seja adequada.

Apresenta-se no próximo capítulo a metodologia proposta por Vieira [35] a ser validada através da implementação distribuída do sistema de controle supervisório considerado neste trabalho. Esta metodologia garante o funcionamento correto do sistema de controle apesar da distribuição dos seus componentes em diversos dispositivos de controle.

Capítulo 5

Controle Supervisório Distribuído

Em muitas situações práticas não é possível implementar o sistema de controle supervisório em apenas um dispositivo de controle. O principal fator que restringe a implementação concentrada é a característica do dispositivo de controle utilizado, sendo a memória disponível o maior problema. Outros fatores limitantes são: número de canais de entrada e saída de sinal, disponibilidade de recursos internos como variáveis internas, contadores e temporizadores, dentre outros. Além destes, podemos destacar também a falta de legibilidade do programa e a dificuldade de se realizar a manutenção do sistema de controle. Ainda, quando há necessidade de aproveitar um sistema de controle já projetado para um determinado equipamento e implementado em um dispositivo de controle específico, a questão do controle supervisório distribuído torna-se essencial. Esta é uma situação muito comum já que muitas vezes os fabricantes de equipamentos fornecem o produto com um sistema de controle otimizado. Resta então ao projetista integrar estes equipamentos, realizando a coordenação da operação conjunta dos mesmos de forma a operarem segundo as especificações definidas.

Em outros contextos a implementação distribuída do controle supervisório também pode ser extremamente desejável, visto que por meio dela é possível preservar a modularidade do sistema físico e das especificações. Neste caso, uma implementação distribuída pode trazer diversos benefícios, inclusive financeiros, com a redução dos custos para interligação dos equipamentos ao CLP único. Com a distribuição pode-se alocar cada dispositivo de controle próximo aos subsistemas que ele controla, modularizando-se o sistema global. Além destas vantagens, acrescenta-se o fato de ser possível adicionar ou remover subsistemas físicos e especificações com maior facilidade, assim como a possibilidade de operação parcial do sistema em caso de falha ou até mesmo para manutenção. Necessita-se, contudo, realizar a troca de dados entre os dispositivos de controle envolvidos no sistema de controle distribuído. A interligação destes dispositivos pode ser realizada por redes industriais tradicionais (Modbus, Profibus, Ethernet, dentre outras) ou até mesmo através dos cartões de entrada e saída dos CLPs.

O maior inconveniente decorrente da descentralização do sistema de controle supervisório diz respeito à sincronização dos programas em execução em cada dispositivo de controle. Para sanar

esse problema Vieira [35] propõe a programação de interfaces responsáveis por manterem a operação conjunta dos sistemas de controle em sincronia.

Neste capítulo é revisada a solução do problema da distribuição de sistemas de controle supervisório, segundo a abordagem de Vieira [35]. No entanto, o capítulo não está restrito a uma revisão bibliográfica deste trabalho, uma vez que são propostas alterações quanto à forma e generalizações de alguns conceitos, que serão destacadas. Em um primeiro momento é abordada a distribuição do sistema no nível das Seqüências Operacionais, denominada *Distribuição Vertical* [36]. Parte-se então para o tratamento da distribuição dos níveis Supervisores Modulares e Sistema Produto chamada *Distribuição Horizontal*. Por fim, para possibilitar a distribuição do sistema de controle supervisório abrangendo todos os níveis é apresentada a *Distribuição Geral*, que nada mais é que a combinação da Distribuição Vertical com a Distribuição Horizontal [37].

5.1 Distribuição Vertical

Na distribuição vertical apenas as seqüências operacionais podem ser implementadas em diferentes dispositivos de controle. Já os supervisores modulares e módulos do sistema produto devem ser implementados no mesmo dispositivo.

Este primeiro grau de distribuição é representado por um grafo de distribuição vertical $D_v = (N_v, A_v)$, onde N_v é o conjunto não vazio de nós e A_v é um conjunto de arcos orientados dados por pares ordenados (x, y) , tal que x e $y \in N_v$. Cada dispositivo de controle utilizado na implementação do controle supervisório é representado por um nó deste grafo. A cada um destes nós (dispositivos de controle) poderão estar associadas uma ou mais seqüências operacionais. Esta associação se dá quando uma seqüência operacional é implementada no dispositivo de controle representado por um determinado nó.

O grafo formado desta maneira tem a estrutura de uma árvore, composta obrigatoriamente por um nó-raiz onde estão implementados os supervisores modulares e os módulos do sistema produto. Ligados a este nó-raiz por um arco orientado pode haver um ou mais nós denominados nós-folha. Nestes nós são implementadas as seqüências operacionais que não estão implementadas no nó-raiz. Ressalta-se que cada seqüência operacional deve ser implementada em apenas um dos dispositivos de controle. Assim sendo, a cada dispositivo de controle está associado um elemento da partição do conjunto de seqüências operacionais.

A Figura 5.1 mostra um grafo de distribuição vertical onde o nó a é o nó-raiz e os nós b e c são os nós-folha. Para um sistema nessa configuração existe um elemento da partição do conjunto de seqüências operacionais em cada um dos dispositivos de controle que o compõem. Em alguns casos, pode não haver nenhum elemento da partição do conjunto de seqüências operacionais definido para o nó-raiz quando não há nenhuma seqüência operacional implementada nele. Estes dispositivos de controle se comunicam através do envio de comandos do nó-raiz para os nós-folha e de respostas dos nós-folha para o nó-raiz. No caso de um sistema com implementação concentrada (em apenas um CLP) o grafo de distribuição vertical possui apenas o nó-raiz e o elemento da partição do conjunto

de seqüências operacionais contida neste nó possui todos os elementos do conjunto de seqüências operacionais.

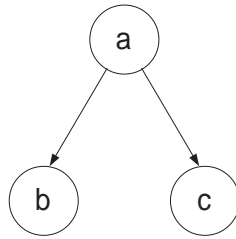


Figura 5.1: Sistema de controle supervisório com distribuição vertical

Alguns problemas podem surgir com a utilização da distribuição vertical. Para sistemas onde os dispositivos de controle são CLPs, problemas de falta de sincronismo podem aparecer resultando na violação da operação do sistema de controle. Esses problemas normalmente ocorrem quando um CLP não reconhece a ativação/desativação de um comando/resposta no mesmo instante que o CLP que gerou esse comando/reposta. Com isso, pode ser que a desativação de um comando ocorra antes que um CLP tenha reconhecido a sua ativação, ou ainda que ocorra mais de uma transição de estado devido a uma mesma ativação de uma resposta. Esta falta de sincronismo é ocasionada pela diferença dos tempos de ciclo que é observada em diferentes CLPs. Maiores detalhes sobre este problema e mecanismos para a sua resolução podem ser encontrados em [35]. Nos casos onde o sistema de controle é composto por outros dispositivos como microcontroladores e placas de circuito integrado os problemas são outros e devem ser resolvidos durante a implementação. Em todos os casos deve-se assegurar que [35]:

- cada comando gerado no nível Sistema Produto seja processado uma, e somente uma, vez no nível das Seqüências Operacionais;
- cada resposta gerada no nível Seqüências Operacionais seja processada uma, e somente uma, vez no nível Sistema Produto;
- a ordem e o instante nos quais tais comandos e respostas são processados não violem as regras que regem o funcionamento do sistema de controle supervisório. Por exemplo, duas respostas não podem ocorrer simultaneamente.

Os problemas decorrentes da distribuição vertical podem ser solucionados pela implementação do protocolo de comunicação do nó-raíz com os nós-folhas assim como por lógicas contidas no programa do nó-raíz e dos nós-folha. Estas soluções são bastante particulares e dependem basicamente do sistema a ser controlado e dos recursos (dispositivos de controle) disponíveis. Basicamente, o que se faz é criar estruturas abstratas denominadas interface raiz e interface folha que são implementadas no dispositivo de controle raiz e nos dispositivos de controle folhas, respectivamente. Estas interfaces têm por objetivo promover a comunicação entre os dispositivos através da geração de comandos externos e respostas externas [35].

Definem-se assim funções que relacionam um comando ao seu análogo externo e uma resposta à sua análoga externa. Para cada módulo do sistema produto representado pelo autômato g_i , com

$i \in I$, cuja seqüência operacional correspondente não pertence ao elemento da partição do conjunto de seqüências operacionais implementada no nó raiz ($\mathbf{o}_i \notin [\mathbf{o}]_{raiz}$) tem-se:

1. $fce_i : \Delta_i \rightarrow \Delta_{ei}$, função que relaciona cada comando ao seu análogo externo;
2. $fre_i : \Pi_{ri} \rightarrow \Pi_{rei}$, função que relaciona cada resposta à sua análoga externa.

onde:

Δ_i é o alfabeto de comandos;

Δ_{ei} é o alfabeto de comandos externos;

Π_{ri} é o alfabeto de respostas;

Π_{rei} é o alfabeto de respostas externas.

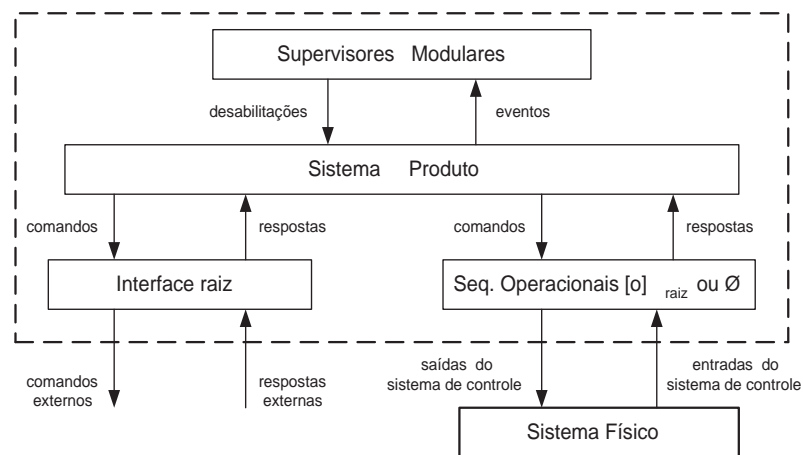
A Figura 5.2a mostra a estrutura do sistema de controle no dispositivo que corresponde ao nó raiz enquanto a Figura 5.2b mostra a estrutura do sistema de controle no dispositivo que corresponde ao nó folha.

É necessário o estabelecimento de canais de comunicação bidirecionais relacionados a cada um dos arcos que ligam o nó raiz a um nó folha no grafo de distribuição vertical. Considerando um nó folha f , o canal de comunicação entre este nó e o nó raiz transporta o pacote de informação $\{C_f, R_f\}$ onde C_f é o conjunto de comandos externos que o nó raiz envia para o nó folha f e R_f é o conjunto de respostas externas que o nó folha f envia para o nó raiz. Com a implementação destas interfaces e dos canais de comunicação bidirecionais, objetiva-se que o comportamento do sistema, provocado pelos comandos e respostas em cada um dos CLPs, seja o mesmo que aquele observado quando o sistema de controle está implementado em um dispositivo único. É de responsabilidade das interfaces o gerenciamento das ativações dos comandos e respostas nos respectivos dispositivos de controle mantendo a coerência do comportamento global.

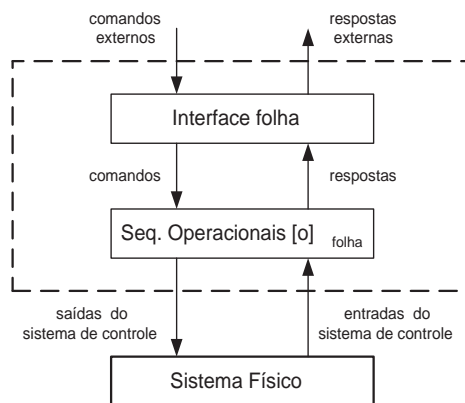
Exemplifica-se na seção 5.1.1 a aplicação destes conceitos através da implementação de controle supervisório com distribuição vertical na CEBE. Este exemplo ilustra bem o caso em que as seqüências operacionais responsáveis pela operação individual de cada subsistema já estavam previamente implementadas e otimizadas em dispositivos embutidos na própria planta.

5.1.1 Distribuição Vertical na CEBE

Nesta seção será feita uma pequena explanação sobre o processamento de baixo nível da CEBE, ou seja, a implementação das seqüências operacionais. Os supervisores modulares e o sistema produto encontrados na seção 3.2 são implementados em um CLP Siemens da família S7-200. Este é considerado o nó raiz do sistema sob distribuição vertical.



(a)



(b)

Figura 5.2: Estrutura do sistema de controle: (a) nó raiz; (b) nó folha [35]

O dispositivo de controle utilizado para implementar o nível das Sequências Operacionais é constituído por duas placas de controle microprocessadas denominadas *Handy Boards* (HBs). Estas placas são baseadas no microprocessador 68HC11 da Motorola e possuem, cada uma, 32k de memória estática, quatro saídas para motores DC, entradas para sensores e display LCD. A programação dos aplicativos para as HBs é realizada na linguagem *Interactive C*, semelhante à linguagem C, que possui suporte à programação multitarefa, além de funções específicas para acionamento de motores CC, entradas e saídas digitais das HBs. A CEBE utiliza duas HBs devido à necessidade de se acionar seis motores CC, uma vez que cada placa é capaz de acionar apenas quatro. Existe um canal de comunicação entre as duas placas realizado por meio da interface serial periférica de ambas. Não será detalhado o protocolo de comunicação neste caso. Considera-se neste exemplo que, do ponto de vista do CLP (nó a), as placas operando conjuntamente funcionam como um dispositivo de controle único (nó b). O grafo de distribuição vertical para este sistema é mostrado na Figura 5.3.

Seja $O = \{o_i | i \in \{0, \dots, 4\}\}$ o conjunto de seqüências operacionais. Como todas as seqüências estão implementadas na HB, tem-se as seguintes classes de equivalência:

- $[o]_a$ não é definida



Figura 5.3: Grafo de distribuição vertical da CEBE

$$- [o]_b = \{o_0, o_1, o_2, o_3, o_4\}$$

Define-se as funções fce_i e fre_i com $i \in \{0, \dots, 4\}$ como a seguir:

$$fce_0(cmd\alpha_0) = cmd\alpha_0ext; \quad fre_0(rpl\beta_0) = rpl\beta_0ext$$

$$fce_1(cmd\alpha_1) = cmd\alpha_1ext; \quad fre_1(rpl\beta_1) = rpl\beta_1ext$$

$$fce_2(cmd\alpha_2) = cmd\alpha_2ext; \quad fre_2(rpl\beta_2) = rpl\beta_2ext$$

$$fce_3(cmd\alpha_3) = cmd\alpha_3ext; \quad fre_3(rpl\beta_3) = rpl\beta_3ext$$

$$fce_4(cmd\alpha_4) = cmd\alpha_4ext; \quad fre_4(rpl\beta_4) = rpl\beta_4ext$$

O canal de comunicação bidirecional entre o CLP e a HB transporta o pacote de informação $\{C_b, R_b\}$ onde:

$$- C_b = \{cmd\alpha_0ext, cmd\alpha_1ext, cmd\alpha_2ext, cmd\alpha_3ext, cmd\alpha_4ext\}$$

$$- R_b = \{rpl\beta_0ext, rpl\beta_1ext, rpl\beta_2ext, rpl\beta_3ext, rpl\beta_4ext\}$$

Cada um dos cinco módulos que constituem a CEBE tem sua seqüência operacional associada a um processo que é executado em uma das HBs. Além destes processos existem alguns outros responsáveis por atividades de monitoramento e comunicação. A Figura 5.4 mostra um diagrama de blocos que representa o funcionamento do programa das HBs onde cada bloco está associado a um processo executado pelo microcontrolador sobre um kernel multitarefas [2].

Como pode ser visto na Figura 5.4 os processos são agrupados em três níveis, sendo eles:

- *1º Nível*: constituído pelos processos responsáveis por monitorar os botões de inicialização e parada da planta. São lançados durante a inicialização da CEBE.
- *2º Nível*: uma vez apertado o botão de início, o processo *MonitoraInício* do 1º Nível instancia os processos deste nível. Estes processos são responsáveis pela interface de comunicação com o CLP (*Comunicação*) e pela criação dos processos que realizam as seqüências operacionais

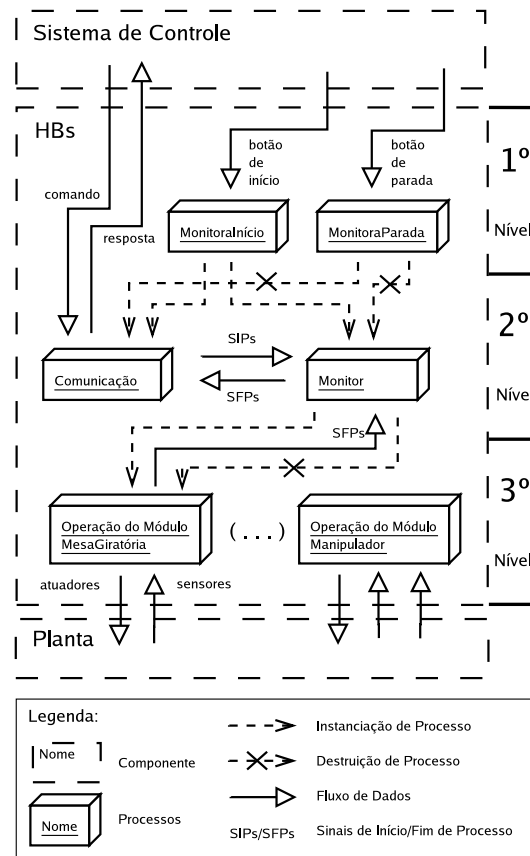


Figura 5.4: Diagrama de blocos do programa das HBs [2]

propriamente ditas (*Monitor*). Estes processos trocam informações, sendo um comando do CLP recebido por *Comunicação* enviado para *Monitora* como um sinal de início de processo (SIP) e um sinal de fim de processo (SFP) recebido por *Comunicação* de *Monitora* enviado para o CLP como uma resposta. Uma vez apertado o botão de parada, o processo *MonitoraParada* do 1º nível é responsável pela finalização dos processos deste nível.

- **3º Nível:** o processo *Monitor* do 2º nível instancia um dos processos *Operação* que realiza a seqüência operacional relativa ao módulo que o CLP comandou o início de operação. Ao terminar a execução da seqüência operacional o processo *Operação* envia um (SFP) para *Monitor* no 2º nível que destrói o processo *Operação* correspondente.

5.1.1.1 Soluções Adotadas nas Interfaces

Nesta seção procura-se explicar as soluções que foram adotadas por Bouzon et al. [2] nas interfaces para possibilitar a distribuição vertical do sistema de controle supervisório. A comunicação entre o CLP Siemens da família S7-200 e a HB principal se dá através da porta serial. O modo de comunicação escolhido foi o *freepoint*, no qual o protocolo de comunicação é definido pelo usuário. A interface utilizada foi a RS485, que não permite a comunicação bidirecional simultânea. Por esta razão, o protocolo implementado é do tipo mestre-escravo, adotando-se o CLP como o mestre da rede, ou seja, a HB só envia uma mensagem se solicitada pelo CLP. Mecanismos para detecção de er-

ros (CRC), falhas no seqüenciamento (números de seqüência) e reenvio de mensagens também foram implementados.

Os comandos externos são representados no CLP por posições de memória pertencentes ao buffer de saída enquanto que as respostas externas são representadas por posições de memória pertencentes ao buffer de entrada. Na HB os comandos externos são posições de memória no buffer de entrada enquanto as respostas externas são posições de memória no buffer de saída.

A comunicação entre o CLP e a HB é simples e ocorre como descrito a seguir. O CLP envia a mensagem contida no seu buffer de saída periodicamente para a HB. Esta mensagem pode conter um único comando externo ou não conter informação alguma. Quando um comando é setado pelo sistema produto presente no nó raiz, ele é traduzido em um inteiro representando um comando externo que deve ser prontamente enviado ao nó folha correspondente. A partir deste momento o CLP fica aguardando a ocorrência de uma interrupção significando a chegada de uma mensagem de resposta vinda da HB.

Quando a HB recebe uma mensagem do CLP contendo um comando externo, o processo *Comunicação* identifica o comando externo que foi enviado pelo CLP e envia para o processo *Monitor* o respectivo comando (SIP). *Monitor* lança então o processo de operação do subsistema relativo ao comando recebido. A cada nova mensagem recebida com um comando externo este procedimento é realizado.

Terminado o processamento de uma seqüência operacional o processo de operação gera uma resposta (SFP) que passa por *Monitor* até chegar em *Comunicação*. Neste processo, a posição do buffer de saída correspondente à resposta externa desta resposta é setada. Este processo pode ocorrer várias vezes até que a mensagem contida no buffer de saída seja de fato enviada, ou seja, uma mensagem da HB para o CLP pode conter mais de uma resposta externa.

A chegada da mensagem da HB no CLP causa uma interrupção que é tratada por ele. Caso seja observada a integridade da mensagem recebida, as respostas externas contidas na mensagem têm as suas respostas análogas setadas. Estas respostas são processadas uma a uma pelo sistema produto e pelos supervisores, produzindo os comandos que serão enviados para a HB como comandos externos. Nesta solução considera-se que não é permitida a chegada de uma nova mensagem da HB antes de todas as respostas externas da mensagem anterior terem sido processadas. Caso isto ocorra as respostas ainda não processadas são sobrescritas pelas novas respostas externas. Na Figura 5.5 é apresentado o fluxograma de funcionamento do programa implementado no CLP.

5.2 Distribuição Horizontal

Como foi visto na seção anterior, a distribuição vertical trata da implementação das seqüências operacionais em dispositivos de controle que não sejam os mesmos onde estão implementados os supervisores e o sistema produto. Já na abordagem de distribuição horizontal, objetiva-se distribuir os supervisores e os módulos do sistema produto. Impõe-se, no entanto, a restrição de que cada

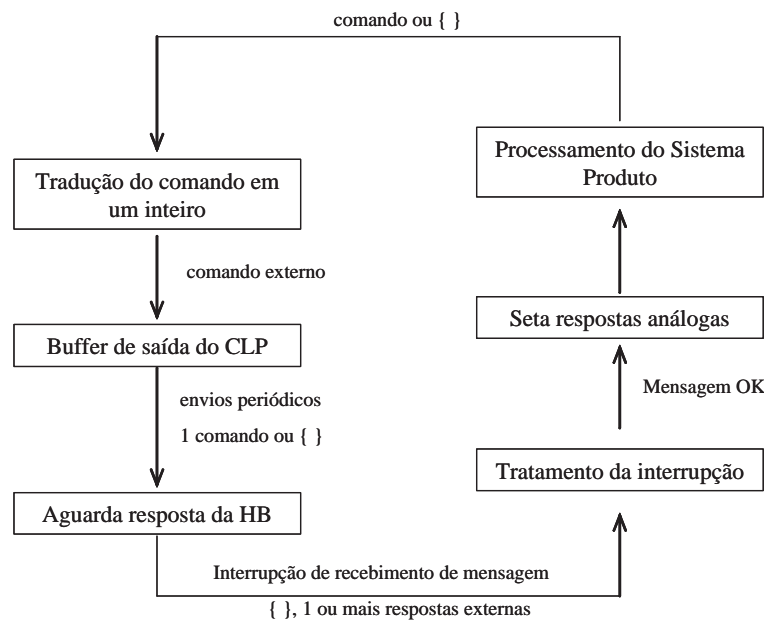


Figura 5.5: Fluxograma de funcionamento do programa implementado no CLP

seqüência operacional é implementada no mesmo dispositivo de controle que o seu correspondente módulo do sistema produto.

A escolha dos supervisores modulares e módulos do sistema produto com suas respectivas seqüências operacionais a serem implementados em cada dispositivo de controle vai depender de basicamente dois fatores: distribuição espacial dos subsistemas e recursos do dispositivo de controle. Com relação a este último pode-se citar a memória de programa disponível, quantidade de portas de comunicação, número de pontos de entrada e saída, dentre outros. Quanto mais complexo o sistema a ser controlado maior a necessidade e as vantagens de se utilizar a implementação distribuída do sistema de controle, uma vez que sistemas maiores são normalmente mais distribuídos espacialmente além de apresentarem uma lógica de controle mais complexa e que dificilmente pode ser implementada em um dispositivo de controle único por falta de recursos deste. Quando é necessário um grande detalhamento na modelagem do comportamento livre dos subsistemas, o programa de controle a ser implementado tende a se tornar maior também. Este é um outro fator que induz a implementação distribuída do sistema de controle supervisório.

Um sistema de controle supervisório com distribuição horizontal é obtido com o particionamento dos conjuntos de supervisores, módulos do sistema produto e seqüências operacionais respeitadas as restrições já mencionadas, com a realização da sincronização de processamento destas estruturas e o estabelecimento da comunicação entre os dispositivos de controle envolvidos. A sincronização de processamento das estruturas deve ser realizada uma vez que pode haver componentes (supervisores e módulos do sistema produto) associados a dispositivos de controle distintos e que façam parte de uma única *célula de controle*. Uma célula de controle é formada por um supervisor local S_j e por todos os módulos do sistema produto que compõem a sua planta local G_{loc_j} , e é representada por $CC_j = \{S_j, G_{loc_j}\}$.

5.2.1 Seqüência de Processamento Sincronizado

Para o correto funcionamento do sistema de controle, deve-se garantir que um módulo do sistema produto só evolua devido à ocorrência de um evento controlável se todos os supervisores que exercem ação de desabilitação sobre esse evento estiverem atualizados e se nenhum deles estiver desabilitando este evento. Além disso, cada supervisor só pode evoluir pela ocorrência de um, e somente um, evento por ciclo de atualização do dispositivo de controle. Por esta razão, o processamento das estruturas implementadas nos diversos dispositivos que compõem o sistema de controle deve ocorrer de forma seqüencial e cíclica em relação às células de controle estabelecidas. Para uma célula de controle $CC_j = \{S_j, G_{loc_j}\}$ um ciclo de atualização é definido pela execução das seguintes tarefas [35]:

1. atualização do estado ativo do gerador que realiza o supervisor S_j seguida da ativação das desabilitações de eventos controláveis definidas pelo novo estado após o processamento;
2. processamento de todos os módulos do sistema produto, cujo autômato seja dado por algum $g_i \in G_{loc_j}$. Este processamento inclui a atualização do estado ativo do autômato além da geração do respectivo evento e deve ocorrer somente após todos os supervisores das células de controle a qual o módulo do sistema produto em questão pertence tiverem sido atualizados.

Admite-se que na distribuição horizontal a priorização das transições referentes a respostas em relação às referentes à negação de desabilitação de eventos controláveis não seja mais necessária. Apesar de distanciar o sistema observado pelo sistema de controle do sistema real, esta simplificação facilita e flexibiliza a distribuição do sistema de controle, o que justifica esta assunção.

Considerando que os dispositivos de controle sejam CLPs, a partir do já exposto, para que os ciclos de atualização ocorram sem infringir alguma das restrições não se pode processar todas as partes do programa de um dado CLP em todos os seus ciclos. É necessário um mecanismo que permita ou não a execução das várias partes do programa de acordo com os critérios estabelecidos de sincronização do sistema distribuído. Estabelece-se, desta forma, uma seqüência de processamento sincronizado sempre que componentes de uma determinada célula de controle estiverem associadas a CLPs diferentes. Por meio das habilitações e desabilitações dos componentes realizadas por essa seqüência de processamento sincronizado os ciclos de atualização são processados na seqüência correta. Cada uma destas seqüências é representada pela listagem da identificação de cada um dos CLPs na ordem em que devem ser processados, como mostrado no exemplo 5.1.

Exemplo 5.1 *Considere um sistema de controle cujas estruturas estão implementadas em 3 CLPs distintos CLP-A, CLP-B e CLP-C. Considere que para o correto processamento das células de controle o CLP-C deve ser processado em primeiro lugar, seguido pelo CLP-A e pelo CLP-B. Esta seqüência de processamento sincronizado é representada por CAB.*

◇

Para o estabelecimento de uma seqüência de processamento sincronizado é necessário definir quais componentes do sistema de controle que a constituirão. Assume-se que cada supervisor e

cada módulo do sistema produto só pode estar presente em uma única seqüência de processamento sincronizado. Assim sendo, considere um módulo do sistema produto $g_i \in G$ que pertence a cada uma das plantas locais do conjunto $\{G_{loc_j}, G_{loc_k}, \dots\}$. A seqüência de processamento sincronizado que inclui g_i deve incluir também todos os módulos do sistema produto que pertençam a alguma planta local pertencente ao conjunto $\{G_{loc_j}, G_{loc_k}, \dots\}$, assim como todos os supervisores que definem células de controle com cada uma destas plantas locais.

Exemplo 5.2 *Considere um sistema cujo conjunto de módulos do sistema produto é dado por $G = \{g_0, g_1, g_2, g_3, g_4\}$. São definidas três especificações genéricas S_a, S_b e S_c que determinam os conjuntos de módulos do sistema produto $G_a = \{g_0, g_1\}$, $G_b = \{g_1, g_2\}$ e $G_c = \{g_3, g_4\}$, respectivamente, que definem as plantas locais. Tem-se, portanto, três células de controle dadas por $CC_a = \{S_a, G_a\}$, $CC_b = \{S_b, G_b\}$ e $CC_c = \{S_c, G_c\}$.*

Observa-se que a célula de controle CC_c não apresenta nenhum componente em comum com as demais, enquanto que CC_a e CC_b possuem o elemento g_1 em comum. Devido a esta inter-relação dos sistemas físicos podem ser criadas duas seqüências de processamento sincronizado que irão ser processadas independentemente. Em uma seqüência serão incluídos os componentes S_a, S_b, g_0, g_1 e g_2 e na outra os componentes S_c, g_3 e g_4 .

◇

Em alguns casos é possível habilitar o processamento das estruturas associadas a diferentes dispositivos de controle relacionados em uma mesma seqüência de processamento sincronizado simultaneamente [35]. Considere dois CLPs (CLP-A e CLP-B) envolvidos em uma mesma seqüência de processamento sincronizado. Considere ainda que S_A e S_B são os conjuntos de supervisores implementados no CLP-A e no CLP-B respectivamente e que G_A e G_B são os conjuntos de módulos do sistema produto implementados no CLP-A e no CLP-B respectivamente. Caso nenhum dos componentes que pertencem a S_A e G_A façam parte de uma mesma célula de controle que os componentes pertencentes a S_B e G_B , o processamento dos componentes de ambos CLPs pode ser executado simultaneamente. Representa-se esta situação na seqüência de processamento sincronizado listando-se os dois CLPs a serem processados simultaneamente entre colchetes ($[AB]$)

Exemplo 5.3 *Considere um sistema cujo conjunto de módulos do sistema produto é dado por $G = \{g_0, g_1, g_2, g_3\}$. São definidas três especificações genéricas E_a, E_b e E_c que determinam os conjuntos de módulos do sistema produto $G_a = \{g_0, g_1\}$, $G_b = \{g_1, g_2\}$ e $G_c = \{g_2, g_3\}$, respectivamente, que definem as plantas locais. Tem-se, portanto, três células de controle dadas por $CC_a = \{S_a, G_a\}$, $CC_b = \{S_b, G_b\}$ e $CC_c = \{S_c, G_c\}$ e todos os componentes devem ser processados em uma mesma seqüência de processamento sincronizado.*

Considere ainda que este sistema de controle terá sua implementação distribuída em três CLPs, CLP-A, CLP-B e CLP-C, segundo as seguintes partições do conjunto de supervisores e do conjunto de módulos do sistema produto:

$$[S]_a = \{S_a\}; \quad [S]_b = \{S_b\}; \quad [S]_c = \{S_c\}$$

$$[g]_a = \{g_0\}; \quad [g]_b = \{g_1, g_2\}; \quad [g]_c = \{g_3\}$$

Observa-se que os componentes implementados no CLP-A não definem células de controle com nenhum dos componentes implementados no CLP-C. Assim sendo, é possível processar os componentes do CLP-A e do CLP-C simultaneamente, o que resulta na seqüência de processamento sincronizado dada por [AC]B.

◇

Para o estabelecimento de uma seqüência de processamento sincronizado deve ser levada em conta a seguinte condição:

Condição Fundamental de Seqüenciamento: *Em um ciclo de atualização de uma célula de controle cujos componentes estão associados a mais de um CLP, só é permitida a atualização de estado de qualquer um dos módulos do sistema produto se já tiver sido realizada a atualização de estado do supervisor que define esta célula de controle.*

Exemplo 5.4 *Considere um sistema cujo conjunto de módulos do sistema produto é dado por $G = \{g_0, g_1, g_2, g_3, g_4\}$. São definidas quatro especificações genéricas E_a, E_b, E_c e E_d que determinam os conjuntos de módulos do sistema produto $G_a = \{g_0, g_2\}$, $G_b = \{g_0, g_4\}$, $G_c = \{g_1, g_2, g_3\}$ e $G_d = \{g_3, g_4\}$, respectivamente, que definem as plantas locais. Tem-se, portanto, três células de controle dadas por $CC_a = \{S_a, G_a\}$, $CC_b = \{S_b, G_b\}$, $CC_c = \{S_c, G_c\}$ e $CC_d = \{S_d, G_d\}$ e todos os componentes devem ser processados em uma mesma seqüência de processamento sincronizado.*

Considere ainda que este sistema de controle terá sua implementação distribuída em três CLPs, CLP-A, CLP-B e CLP-C e que a seqüência de processamento sincronizado é ABC. Uma possível partição do conjunto de supervisores e do conjunto de módulos do sistema produto que atenda à condição fundamental de seqüenciamento é:

$$[S]_a = \{S_a, S_b\}; \quad [S]_b = \{S_c\}; \quad [S]_c = \{S_d\}$$

$$[g]_a = \{g_0\}; \quad [g]_b = \{g_1, g_2\}; \quad [g]_c = \{g_3, g_4\}$$

Observa-se que um módulo do sistema produto só é atualizado se todos os supervisores que definem células de controle com ele já tiverem o sido. Imagine que o supervisor S_b não seja mais processado no CLP-A, e sim no CLP-B. Neste caso, a condição fundamental de seqüenciamento é violada, uma vez que g_0 é processado antes de S_b e ambos fazem parte da célula de controle CC_b . Em uma outra situação hipotética, imagine que g_3 fosse implementado no CLP-B. A condição fundamental de seqüenciamento também seria quebrada uma vez que g_3 seria processado antes de S_d .

◇

Cada seqüência de processamento sincronizado de um sistema de controle supervisório com distribuição horizontal é representada por um grafo de distribuição horizontal $D_h = (N_h, A_h)$, onde N_h é o conjunto não vazio de nós e A_h é um conjunto de arcos orientados dados por pares ordenados (x, y) , tal que x e $y \in N_h$. Cada dispositivo de controle utilizado na implementação do controle supervisório é representado por um nó deste grafo. Deverá haver um arco interligando os nós x e y , com origem no primeiro, sempre que o CLP-Y suceda imediatamente o CLP-X na seqüência de processamento sincronizado. Neste grafo é representada a seqüência na qual os CLPs terão suas estruturas processadas e indica ainda a possibilidade ou não de simultaneidade de processamento, como mostra o exemplo 5.5.

Exemplo 5.5 *Um sistema de controle supervisório com distribuição horizontal é implementado em cinco CLPs: CLP-A, CLP-B, CLP-C, CLP-D e CLP-E. Suponha que a seqüência de processamento sincronizado seja dada por $A[BC]DE$. O grafo de distribuição horizontal que representa este sistema é mostrado na Figura 5.6.*

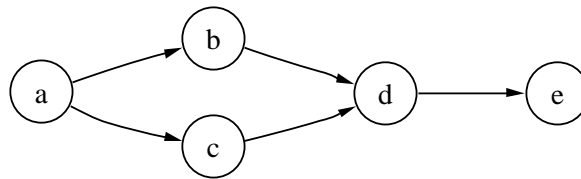


Figura 5.6: Grafo de distribuição horizontal que representa a seqüência de processamento sincronizado $A[BC]DE$

◇

5.2.2 Autômatos Sincronizadores

Para cada uma das seqüências de processamento sincronizado estabelecidas são definidas estruturas denominadas autômatos sincronizadores [35]. O autômato sincronizador referente a uma determinada seqüência de processamento sincronizado é implementado em cada um dos CLPs que fazem parte desta seqüência de processamento sincronizado. Caso um CLP esteja incluso em mais de uma seqüência de processamento sincronizado, deve-se implementar nele mais de um autômato sincronizador. Estes autômatos sincronizadores evoluem independentemente. A habilitação ou não do processamento das estruturas implementadas em um CLP é baseada no estado ativo dos autômatos sincronizadores.

O autômato sincronizador implementado no CLP correspondente ao primeiro elemento da seqüência de processamento sincronizado difere dos demais quanto ao estado inicial. Propõe-se que o autômato referente ao primeiro elemento da seqüência de processamento sincronizado seja como o mostrado na Figura 5.7a e que os demais sejam como o mostrado na Figura 5.7b.

Nestes autômatos o estado W_k (*waiting state*) representa o estado no qual o CLP está aguardando o recebimento de informações atualizadas. O estado S_k (*settling state*) indica que o CLP está recebendo

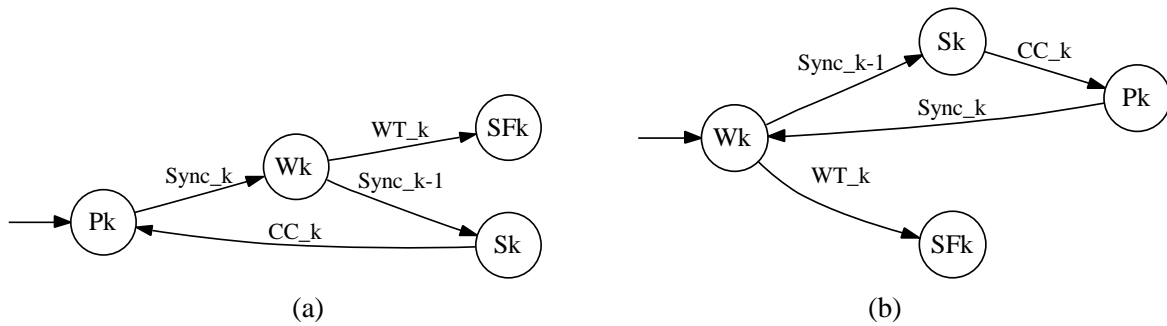


Figura 5.7: Autômatos Sincronizadores: (a) primeiro elemento da seqüência de processamento sincronizado; (b) demais elementos

informações atualizadas. No estado P_k (*processing state*) o CLP está processando as estruturas de controle. O estado SF_k (*synchronization failure state*) representa um estado de falha na sincronização.

Para viabilizar a sincronização entre os vários CLPs que constituem a seqüência de processamento sincronizado são empregadas as variáveis de sincronização $sync_k$. Após o processamento das estruturas de controle no CLP- k é gerado o evento de sincronização $sync_k$, enviado ao próximo CLP da seqüência de processamento sincronizado, e o estado ativo do autômato torna-se W_k . O CLP fica neste estado de espera pelo início de recebimento de informações atualizadas até que recebe do CLP que o antecede na seqüência de processamento sincronizado o evento de sincronização $sync_{k-1}$. A sinalização deste evento leva ao estado S_k . Caso este evento de sincronização não chegue por alguma razão, é gerado o evento WT_k (*waiting time exceeded*) que leva o autômato para o estado SF_k , de onde ele não sai. Uma vez no estado S_k , quando é concluído com sucesso o recebimento dos dados, o evento CC_k (*communication concluded*) é gerado no próprio CLP, levando-o ao estado de processamento das estruturas de controle novamente, fechando o ciclo de atualização.

Nota-se que para um sistema implementado com distribuição horizontal, um ciclo de atualização das células de controle só ocorre após diversos ciclos de atualização do CLP. Isto porque a necessidade de sincronização faz com que o CLP fique por vários ciclos em estado de espera, sem de fato processar nenhuma estrutura do sistema de controle. Esta é uma desvantagem com relação ao sistema implementado em um dispositivo de controle único, uma vez que neste caso um ciclo de atualização das células de controle ocorre a cada ciclo de atualização do CLP.

Vieira [35] implementa os autômatos sincronizadores com cada um dos estados mostrados nas Figuras 5.7a e 5.7b duplicados. Além disso, considera dois tipos de variáveis de sincronização ($sync_{k1}$ e $sync_{k2}$) como mostra a Figura 5.8. Este tipo de representação abrange os casos onde a comunicação realizada entre os dispositivos de controle é cíclica, e por isso há a necessidade de dois tipos de variável de sincronização para que o sistema identifique que a mensagem que está chegando é nova. Essa identificação ocorre sempre quando há alternância entre as variáveis de sincronização que são recebidas ciclicamente. Já no caso da comunicação acíclica, não há a necessidade de se utilizar variáveis de sincronização distintas para diferenciação, uma vez que o sistema acíclico identifica a chegada da nova mensagem sempre que a mensagem de sincronização única é recebida.

Sugere-se que a representação dos autômatos sincronizadores seja sempre feita conforme a Figura

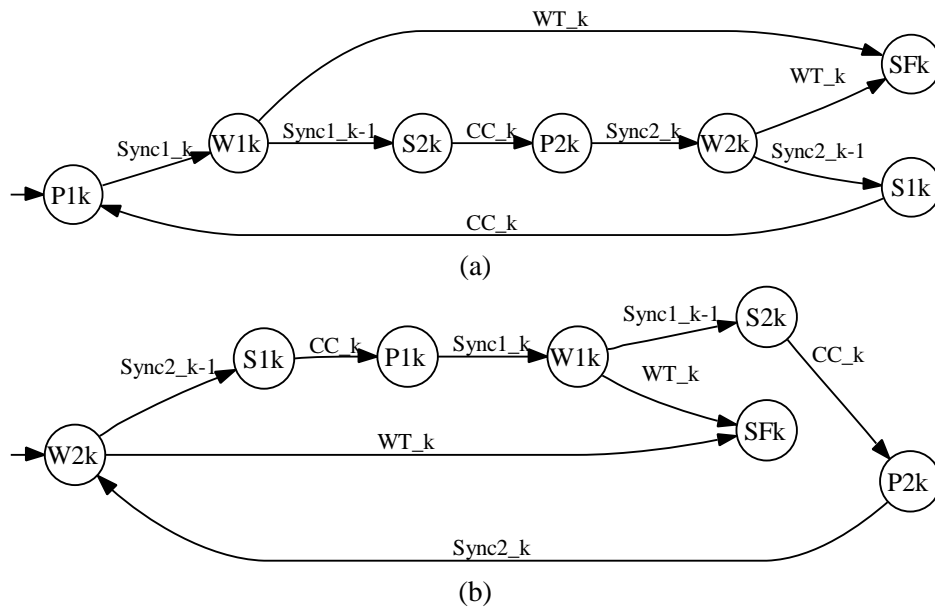


Figura 5.8: Autômatos Sincronizadores segundo [35] : (a) primeiro elemento da seqüência de processamento sincronizado; (b) demais elementos

5.7 e que a adequação ao tipo de comunicação utilizado seja realizada no programa implementado no dispositivo de controle de acordo com as necessidades, como será mostrado na Seção 6.4. Esta proposta generaliza a representação dos autômatos sincronizadores, conferindo maior simplicidade e tornando-a independente do tipo de tecnologia de comunicação utilizada.

Vieira [35] ainda adapta a estrutura destes autômatos para os casos onde é possível a habilitação simultânea do processamento de mais de um CLP. Uma vez que o processamento simultâneo dos componentes do sistema de controle de múltiplos CLPs não é possível no problema abordado neste trabalho, o estudo aprofundado destas estruturas não será feito aqui.

5.2.3 Canais de Comunicação

Além da sincronização das estruturas do sistema de controle distribuído, é necessário promover a troca de informações entre os vários CLPs que compõem o sistema, já que estruturas que fazem parte de uma mesma célula de controle podem estar implementadas em diferentes dispositivos de controle. O maior desafio é determinar entre quais dispositivos de controle é necessário estabelecer canais de comunicação e ainda definir quais são os dados que realmente precisam ser transmitidos de modo a evitar ausência de dados necessários e/ou sobrecarga da rede com o envio de dados irrelevantes. Assim como na distribuição vertical, esta comunicação é realizada por meio dos dados externos.

Define-se assim funções que relacionam um evento ao seu análogo externo, uma desabilitação de eventos controláveis à sua análoga externa e uma sinalização de evolução de um módulo do sistema produto à sua análoga externa. Para cada módulo do sistema produto representado pelo autômato g_i , com $i \in I$, pertencente a alguma célula de controle cujos componentes estão associados a dois ou mais CLPs, devem ser estabelecidas as seguintes funções bijetoras [35]:

1. $f_{ee_i} : \Sigma_i \rightarrow \Sigma_{ei}$, função que relaciona cada evento ao seu análogo externo;
2. $f_{de_i} : \Theta_i \rightarrow \Theta_{ei}$, função que relaciona cada desabilitação de evento controlável à sua análoga externa;
3. $f_{ge_i} : \Xi \rightarrow \Xi_e$, função que relaciona cada sinalização de evolução de um módulo do sistema produto à sua análoga externa.

onde:

Σ_i é o alfabeto de eventos;

Σ_{ei} é o alfabeto de eventos externos;

Θ_i é o alfabeto de desabilitações de eventos controláveis;

Θ_{ei} é o alfabeto de desabilitações externas;

Ξ é o conjunto de sinalizações de evolução dos módulos do sistema produto;

Ξ_e é o conjunto de evoluções externas;

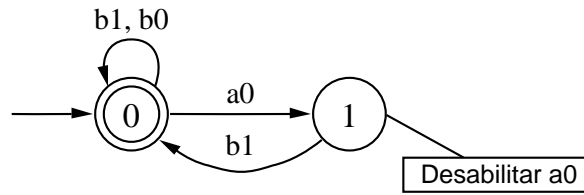
5.2.3.1 Eventos Relevantes e Desabilitações Efetivas

Como já foi dito é de extrema importância reconhecer os eventos externos e as desabilitações externas que realmente precisam ser transmitidas. Com este intuito, Vieira [35] estabelece um subconjunto do conjunto de eventos e um subconjunto do conjunto de desabilitações que formam respectivamente o conjunto dos eventos relevantes ($\Sigma_{relev. i-j} \subseteq \Sigma_i$) e o conjunto das desabilitações efetivas ($\Theta_{efet. i-j} \subseteq \Theta_i$).

Um evento σ em Σ_i pertence ao conjunto de eventos relevantes para o par (S_j, g_i) , ou seja, $\sigma \in \Sigma_{relev. i-j}$ se σ causa pelo menos uma mudança de estado no gerador que realiza o supervisor S_j . Pode-se generalizar o conceito de eventos externos para o de eventos externos relevantes. O conjunto de eventos externos relevantes $\Sigma_{ext. relev. i-j}$ para o par (S_j, g_i) é obtido pela função $f_{ee_i} : \Sigma_{relev. i-j} \rightarrow \Sigma_{ext. relev. i-j}$.

Uma desabilitação de evento controlável αd em Θ_i pertence ao conjunto de desabilitações efetivas para o par (S_j, g_i) , ou seja, $\alpha d \in \Theta_{efet. i-j}$ se αd , responsável pela desabilitação de α , é ativada pela função de desabilitação de eventos em pelo menos um estado do gerador que realiza o supervisor S_j . Pode-se generalizar o conceito de desabilitações externas para o de desabilitações externas efetivas. O conjunto de desabilitações externas efetivas $\Theta_{ext. efet. i-j}$ para o par (S_j, g_i) é obtido pela função $f_{de_i} : \Theta_{efet. i-j} \rightarrow \Theta_{ext. efet. i-j}$.

Exemplo 5.6 Considere uma célula de controle composta pelo supervisor S_a mostrado na Figura 5.9 e pelos módulos do sistema produto g_0 e g_1 mostrados nas Figuras 5.10a e 5.10b, respectivamente. Eventos do tipo a são controláveis enquanto que os do tipo b são não controláveis.

Figura 5.9: Supervisor S_a Figura 5.10: Módulos do sistema produto: (a) g_0 ; (b) g_1

O alfabeto de eventos relevantes e de desabilitações efetivas para os pares (S_a, g_0) e (S_a, g_1) são:

$$\begin{aligned} \Sigma_{relev. 0-a} &= \{a0\}; & \Sigma_{relev. 1-a} &= \{b1\} \\ \Theta_{efet. 0-a} &= \{a0d\}; & \Theta_{efet. 1-a} &= \{\} \end{aligned}$$

Observe que $b0$ aparece no supervisor apenas como um auto-laço. Desta forma, $b0 \notin \Sigma_{relev. 0-a}$.

◇

A partir da definição destas funções deve-se definir quais são, efetivamente, os dados a serem trocados entre os CLPs envolvidos no sistema de controle. Vieira [35] define três ítems que devem ser levados em consideração na definição desta troca de dados:

1. Os eventos externos relevantes relacionados a um módulo do sistema produto $g_k \in G$ e a um supervisor $S_y \in S$ ($\sigma_{ext} \in \Sigma_{ext. relev. k-y}$) devem ser transmitidos do CLP no qual g_k está implementado para o CLP em que está implementado o supervisor S_y . É necessário transmitir os eventos externos relevantes de g_k para todo CLP onde estejam implementados supervisores que constituem uma célula de controle com g_k .
2. Considere o conjunto de módulos do sistema produto G_k que forma uma célula de controle com o supervisor S_k . As desabilitações externas efetivas pertencentes ao alfabeto de desabilitações externas $\Theta_{ext. efet. i-k}$ de algum dos módulos $g_i \in G_k$, devem ser transmitidas do CLP onde está implementado o supervisor S_k para o CLP onde está implementado o autômato g_i correspondente. É necessário transmitir as desabilitações externas efetivas do CLP onde está implementado S_k para todo CLP onde estejam implementados módulos do sistema produto que constituem uma célula de controle com S_k .
3. A evolução externa correspondente a um módulo do sistema produto $g_k \in G$ deve ser transmitida para todos os CLPs onde estejam implementados módulos do sistema produto que constituem uma célula de controle com g_k , desde que estes CLPs sucedam o CLP onde está implementado g_k na seqüência de processamento sincronizado.

Exemplo 5.7 Considere o supervisor S_a e os módulos do sistema produto g_0 e g_1 mostrados no exemplo 5.6. Suponha que S_a está implementado em um CLP-A, enquanto g_0 e g_1 estão implementados no CLP-B e no CLP-C, respectivamente. Suponha ainda que a seqüência de processamento sincronizado é dada por ABC.

De acordo com os itens definidos anteriormente e com os eventos relevantes e as desabilitações efetivas encontradas no exemplo 5.6, necessita-se realizar a seguinte troca de dados:

1. do CLP-B para o CLP-A transmitir $\{a_{0e}\}$ e do CLP-C para o CLP-A transmitir $\{b_{1e}\}$;
2. do CLP-A para o CLP-B transmitir $\{a_{0d_e}\}$;
3. do CLP-B para o CLP-C transmitir $\{g_0evt_e\}$;

◇

5.2.3.2 Grafo de Comunicação

A partir da definição dos dados a serem trocados entre os dispositivos de controle, estabelece-se os canais de comunicação necessários para a troca de dados entre os CLPs. Cada um destes canais transporta um pacote de informações $\{E_{ab}, D_{ab}, V_{ab}\}$ onde:

- E_{ab} é o conjunto de eventos externos transmitidos do CLP-A para o CLP-B
- D_{ab} é o conjunto de desabilitações externas transmitidas do CLP-A para o CLP-B
- V_{ab} é o conjunto de evoluções externas transmitidas do CLP-A para o CLP-B

Definidos os canais de comunicação necessários para a troca de informações entre os CLPs que constituem o sistema de controle, pode-se elaborar um grafo de comunicação que representa a interligação entre os CLPs. Cada nó do grafo de comunicação representa um CLP envolvido no sistema de controle. Cada arco direcionado deste grafo representa um canal de comunicação unidirecional que interliga o CLP representado pelo nó de onde parte o arco ao CLP representado pelo nó destino deste arco.

Exemplo 5.8 A partir da definição dos dados a serem trocados no sistema do problema do Exemplo 5.7, pode-se definir o grafo de comunicação que representa como deve ser feita a interligação entre os CLPs. Os pacotes de informação a serem transmitidos são constituídos pelos seguintes elementos:

- $E_{ba} = \{a_{0e}\}$;
- $E_{ca} = \{b_{1e}\}$;
- $V_{bc} = \{g_0evt_e\}$;

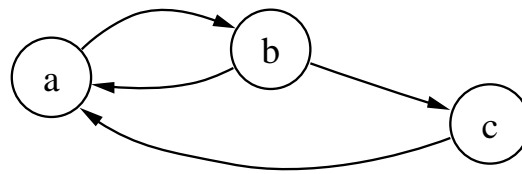


Figura 5.11: Grafo de comunicação

$$- D_{ab} = \{a_0 d_e\}.$$

Assim sendo, é necessário estabelecer a interconexão dos CLPs dada pelo grafo de comunicação mostrado na Figura 5.11.

◇

Vieira [35] representa os canais de comunicação entre os dispositivos de controle no grafo de distribuição horizontal através da adição de rótulos a cada um dos arcos. A Figura 5.12 ilustra como seria a representação do sistema do problema do Exemplo 5.7 pelo grafo de distribuição horizontal proposto em [35]. Assim como na representação porposta nesta dissertação de mestrado, cada um dos nós representa um dispositivo de controle e eles são interligados pelos arcos conforme a seqüência de processamento sincronizado. Os rótulos dos arcos contém duas informações: a representação da seqüência de processamento sincronizado, que fica à esquerda, e os canais bidirecionais de comunicação a partir do nó do qual o arco se origina. Observa-se que a informação relativa à comunicação nada tem a ver com a estrutura do grafo, uma vez que na Figura 5.12 os dados trocados entre o nó A e o nó C não passam pelo nó B, o que é sugerido pela representação. Além disso, nesta representação todos os canais de comunicação são bidirecionais. Como pode ser observado pela Figura 5.11 os canais de comunicação entre os nós B e C e entre os nós A e C não precisam ser bidirecionais e esta representação não possibilita que esta informação seja representada.

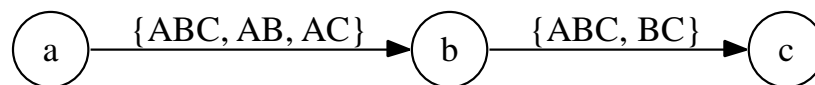


Figura 5.12: Grafo de distribuição horizontal do sistema do problema do Exemplo 5.7 segundo Vieira [35]

Por esta razão foi proposta uma representação distinta para os canais de comunicação, o grafo de comunicação. Além de simplificar a representação do grafo de distribuição horizontal, possibilita que os aspectos relativos à comunicação sejam melhores ilustrados, já que através do grafo de comunicação pode-se diferenciar um canal de comunicação unidirecional de um canal de comunicação bidirecional, além de representar fielmente a interligação física entre os dispositivos de controle. Esta proposta confere maior generalidade e simplicidade à representação da distribuição horizontal de sistemas de controle supervisório.

5.2.4 Implementação do Sistema de Controle com Distribuição Horizontal

Analogamente ao que foi feito na distribuição vertical, são criadas estruturas abstratas denominadas interface de entrada e interface de saída que são implementadas em cada um dos dispositivos de controle que correspondem a um nó do grafo de distribuição horizontal. Estas interfaces têm por objetivo promover a comunicação entre os dispositivos através da ativação e desativação adequada de eventos externos, desabilitações externas e evoluções externas. É de responsabilidade da interface de saída a ativação com retenção dos sinais a serem transmitidos e a desativação dos mesmos somente quando a transmissão tiver sido concluída. Já a interface de entrada recebe os sinais vindos dos outros CLPs e ativa internamente os análogos aos eventos externos, desabilitações externas e evoluções externas recebidos. Estas interfaces atuam de acordo com o estado ativo dos autômatos sincronizadores.

A Figura 5.13 mostra a estrutura para implementação do i -ésimo dispositivo de controle envolvido no sistema de controle com distribuição horizontal com a inclusão das interfaces de entrada e saída.

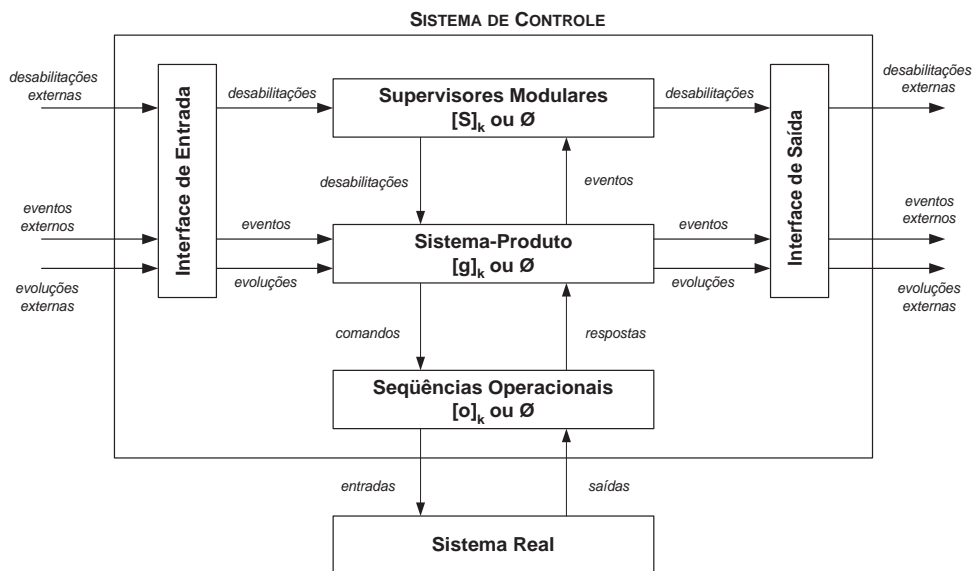


Figura 5.13: Implementação do sistema de controle para um nó do grafo de distribuição horizontal [35]

5.3 Distribuição Geral

Há situações em que apenas a distribuição vertical ou apenas a distribuição horizontal do sistema de controle não é suficiente. Em alguns casos é desejável, ou até mesmo fundamental, a distribuição dos componentes de todos os níveis do modelo de implementação apresentado no Capítulo 4. E isto não é possível nos outros modelos já apresentados devido às restrições de distribuição impostas em cada um deles.

No caso da distribuição vertical, os supervisores e os módulos do sistema produto devem sempre estar implementados em um mesmo dispositivo de controle. O grau de liberdade de distribuição

fica por conta somente das seqüências operacionais, que podem ser implementadas em diferentes dispositivos de controle. Já no caso da distribuição horizontal, pode-se implementar os supervisores e os módulos do sistema produto em diferentes dispositivos de controle. No entanto, é imposta a restrição de que as seqüências operacionais e seus respectivos módulos do sistema produto devem sempre ser implementados em um mesmo dispositivo de controle. A distribuição geral tem como objetivo combinar a distribuição vertical à distribuição horizontal, viabilizando a distribuição de todos os níveis do modelo de distribuição. A distribuição geral consiste da distribuição horizontal do sistema de controle seguida da distribuição vertical de cada CLP utilizado na distribuição horizontal [35].

Após a realização da distribuição horizontal do sistema de controle, todas as seqüências operacionais estão implementadas no mesmo dispositivo de controle que seus respectivos módulos do sistema produto. No entanto, em vários casos práticos, pode ser necessário implementar estas seqüências operacionais em dispositivos de controle diferentes. Um exemplo típico é quando esta seqüência já está otimizada e implementada em um dispositivo de controle interno ao subsistema físico. Para situações como essas, aplica-se a distribuição vertical aos dispositivos de controle envolvidos na distribuição horizontal de modo a possibilitar a implementação das seqüências operacionais em dispositivos de controle diferentes daqueles onde estão implementados os seus respectivos módulos do sistema produto.

A representação de um sistema com distribuição geral é realizada em dois níveis. No primeiro nível tem-se o grafo de distribuição horizontal, representando o primeiro grau de distribuição do sistema. Em um segundo nível obtém-se os grafos de distribuição vertical onde cada nó do grafo de distribuição horizontal representa o nó raiz de um destes grafos, aos quais podem ser associados nós folhas, que correspondem aos dispositivos de controle nos quais estão implementadas as seqüências operacionais associadas aos módulos do sistemas produto implementados no nó raiz correspondente. Cada nó k do grafo de distribuição horizontal está associado a um nó raiz do grafo de distribuição vertical denominado k -raiz e os nós folha a ele associados são denominados k -folha1, k -folha2, ..., sendo que é possível que um ou mais destes nós estejam associados a dois ou mais nós raiz. Este procedimento possibilita que as seqüências operacionais que constituem a classe de equivalência $[o]_k$ sejam particionadas em novas classes de equivalência definidas por $[o]_{k-raiz}$, $[o]_{k-folha1}$, $[o]_{k-folha2}$, ... sendo que $[o]_{k-raiz}$ pode não ser definida.

Exemplo 5.9 *Considerando-se o Exemplo 5.4, suponha que é necessário implementar algumas das seqüências operacionais em dispositivos de controle (CLP-D e CLP-E) diferentes dos que estão implementados os seus respectivos módulos do sistema produto. Considere a seguinte partição do conjunto de seqüências operacionais:*

$$[o]_a = \text{não definida}; \quad [o]_b = \{o_2\}; \quad [o]_c = \text{não definida};$$

$$[o]_d = \{o_0, o_1\}; \quad [o]_e = \{o_3, o_4\}$$

Pode-se representar este sistema de controle com distribuição geral por meio da representação em dois níveis mostrada na Figura 5.14.

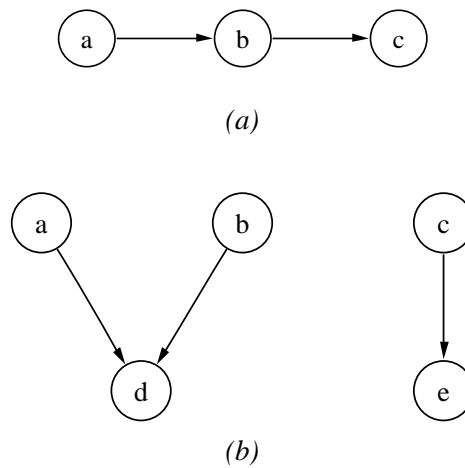


Figura 5.14: Representação da distribuição geral: (a) 1º nível; (b) 2º nível

◇

A Figura 5.15 mostra a estrutura para implementação em um CLP correspondente a um nó do grafo de distribuição horizontal em que foi realizada a distribuição vertical das seqüências operacionais. Caso não haja necessidade de se realizar a distribuição vertical dos componentes de algum nó do grafo de distribuição horizontal, a sua estrutura permanece sendo a mostrada na Figura 5.13. A estrutura dos nós correspondentes aos nós folha é a mesma mostrada na Figura 5.2b.

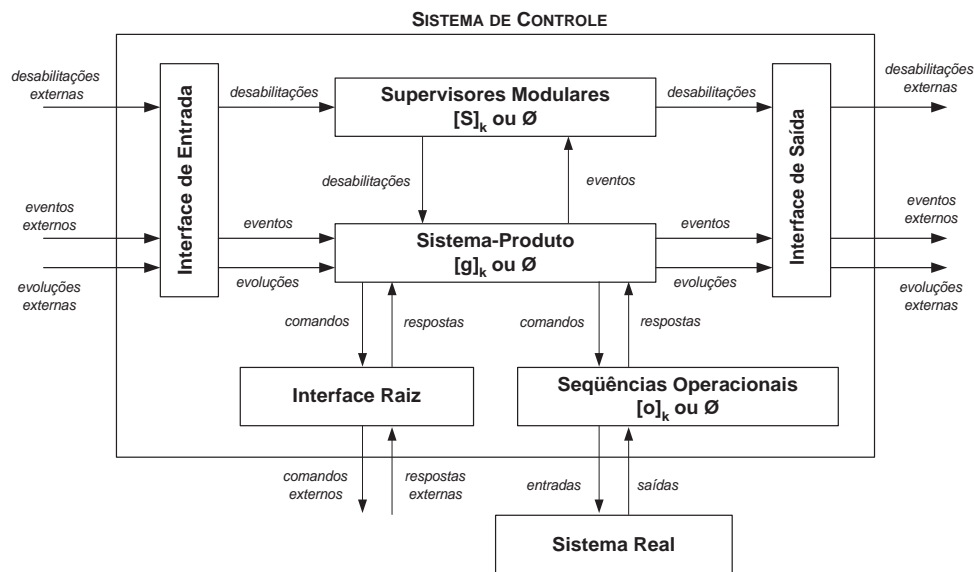


Figura 5.15: Implementação do sistema de controle para um nó do grafo de distribuição horizontal com distribuição vertical das seqüências operacionais [35]

5.4 Conclusão do Capítulo

Neste capítulo foi mostrada a metodologia utilizada para a realização da distribuição do sistema de controle supervisório. Essa metodologia abrange tanto aspectos de sincronização quanto de comu-

nicação de dados que são essenciais para o correto funcionamento do sistema. O desenvolvimento desta metodologia em sua forma final (distribuição geral) se deu por meio da união de outras duas formas de distribuição menos complexas, porém mais restritas: a distribuição vertical e a distribuição horizontal. Além da revisão bibliográfica do trabalho de Vieira [35] foram propostas algumas generalizações e alterações quanto à forma de algumas representações, como no caso dos autômatos sincronizadores, do grafo de distribuição horizontal e do grafo de comunicação.

No capítulo seguinte apresenta-se a aplicação desta metodologia a um sistema real, a CEBE. Serão abordadas três configurações distintas de distribuição, sendo que em todos os casos será utilizado o conceito de distribuição geral. O objetivo principal deste trabalho é implementar em um sistema real um sistema de controle supervisório distribuído em mais de um dispositivo de controle. Com isso é possível validar e mostrar a aplicabilidade da teoria apresentada neste capítulo. Além disso, serão propostos critérios de avaliação para a escolha da configuração do sistema de controle distribuído que mais se adeque a um determinado problema.

Capítulo 6

Implementação Distribuída do Controle Supervisório na CEBE

Muitos dos sistemas de manufatura encontrados na indústria atualmente são constituídos por células flexíveis. O controle deste tipo de sistema visa a coordenação da operação conjunta dos diversos subsistemas que o compõem, de modo que atividades pré-estabelecidas sejam desempenhadas sem que, no entanto, sejam desrespeitadas certas condições de funcionamento, ou especificações.

Conforme já foi exaustivamente mencionado no decorrer deste trabalho, em muitos casos práticos não é possível implementar um sistema de controle supervisório em um dispositivo de controle único. Isso se deve ao grande fator restritivo que impede que a TCS seja utilizada largamente na solução de problemas de controle de sistemas a eventos discretos. Trata-se do procedimento de síntese do supervisor monolítico, cuja complexidade computacional é polinomial no tamanho dos modelos da planta e das especificações. Para modelos com muitos estados, a síntese do supervisor pode levar a uma explosão no número de estados.

No entanto, implementando-se o sistema de controle em mais de um dispositivo de controle, pode-se viabilizar a utilização desta teoria na resolução de problemas práticos de pequeno/médio porte reais. E é exatamente neste ponto que entra a principal contribuição deste trabalho. Os conceitos abordados até aqui serão aplicados na implementação distribuída do controle supervisório na CEBE. Este sistema, apesar de ser uma planta piloto que simula uma situação hipotética, representa uma classe de sistemas recorrentemente encontrados em aplicações práticas reais. Neste processo surgem alguns problemas de implementação específicos para a célula flexível escolhida e seus dispositivos de controle, que são também tratados.

Neste capítulo são mostrados diversos aspectos e os procedimentos que levaram ao principal resultado dessa dissertação de mestrado: a implementação distribuída do controle supervisório para um SED. A seção inicial aborda aspectos relacionados ao hardware disponível para esta implementação. Em um segundo momento são formulados três problemas de distribuição e apresentadas as suas soluções. Em seguida são propostos critérios para a avaliação da configuração do sistema de controle supervisório distribuído mais adequada a cada aplicação, com suas características específicas. Por

fim, a estrutura básica do programa implementado nos CLPs é mostrada, assim como as soluções de algumas questões julgadas relevantes do problema abordado.

6.1 Configurações de Hardware

Para a elaboração de um sistema de controle supervisório distribuído é de extrema importância o conhecimento do hardware disponível. É necessário saber a capacidade de cada um dos dispositivos de controle, assim como as formas com que eles podem se comunicar entre si. A partir destes dados, é possível especificar quais e quantos são os dispositivos de controle que devem ser de fato utilizados na implementação.

Nesta seção são apresentadas a configuração de hardware existente antes do início das atividades deste trabalho e a configuração atual, após a aquisição de um novo dispositivo de controle. Com a configuração pré-existente foi possível implementar o sistema de controle concentrado em um único dispositivo de controle como mostrado no capítulo 3. Com a compra do novo dispositivo foi possível implementar o controle distribuído nas várias configurações propostas e realizar a troca de dados por meio de uma rede Profibus DP, cujos conceitos básicos são também apresentados.

6.1.1 Configuração Pré-Existente

No início das atividades deste trabalho, o sistema de controle existente no Laboratório de Automação Industrial (LAI) da Universidade Federal de Santa Catarina era constituído por duas Handy-Boards (vide seção 5.1.1) HBmain e HBaux, dois CLPs Siemens da família S7-200 e um PC com a plataforma de programação e monitoramento Step 7-MicroWin 32 para os CLPs S7-200. Os CLPs S7-200 combinam um microprocessador, fonte integrada e circuitos de entrada e saída em um compacto encapsulamento, caracterizando-os como micro CLPs. A memória de programa tem capacidade para armazenar 2048 palavras (4096 bytes), enquanto que a memória de dados é de 1024 palavras.

As duas Handy-Boards são responsáveis pelo controle de baixo nível da planta. Do ponto de vista do CLP, funcionavam como um dispositivo de controle único. Para possibilitar isso, comunicavam-se através da interface serial periférica, um padrão da Motorola para comunicações síncronas.

Nesta configuração apenas um dos CLPs era utilizado para a implementação. A comunicação do CLP com as HBs se dava por meio de troca serial de informações entre a porta serial padrão RS485 (serial multiponto) do CLP e a porta serial padrão RS232 (serial ponto-a-ponto) da HBmain. Como os padrões eram diferentes utilizava-se um elemento de conversão para compatibilização dos padrões. O modo de comunicação escolhido foi o *freepoint*, no qual o protocolo de comunicação é definido pelo usuário. O protocolo desenvolvido é do tipo mestre-escravo, e tem como mestre o CLP.

A rede profibus existente era utilizada nesta época somente para comunicação do PC, que tem a plataforma de programação e monitoramento Step 7-MicroWin 32, com os CLPs. Dois módulos de expansão EM-277 possibilitavam a comunicação dos CLPs na rede profibus. Desta forma, podia-se fazer *download* e *upload* dos programas para os CLPs, além do monitoramento de variáveis em

tempo real. No entanto, estes módulos só são capazes de atuarem como escravos na rede. Por isso, percebia-se que a rede profibus não estava tendo todo seu potencial utilizado, uma vez que não existia um dispositivo capaz de assumir o papel de mestre da mesma. O diagrama do sistema, conforme descrito acima, é mostrado na Figura 6.1.

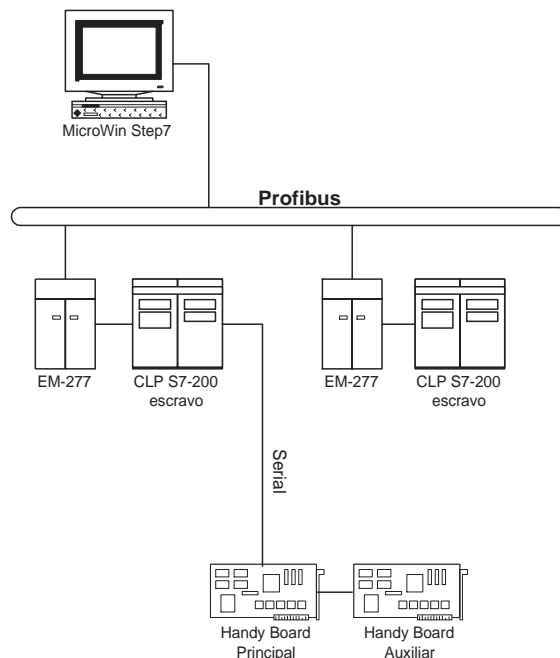


Figura 6.1: Configuração de *hardware* pré-existente

6.1.2 Configuração Atual

Como já foi mencionado na seção anterior, percebeu-se que a rede profibus existente não vinha tendo todas as suas potencialidades utilizadas. Desejava-se que as trocas de dados entre os dispositivos de controle quando fosse realizada a distribuição do sistema de controle fossem realizadas através da rede profibus, uma rede padrão e amplamente utilizada na indústria. Para que isto fosse possível era necessária a existência de um dispositivo que atuasse como mestre da rede.

Especificou-se então para compra um CLP da família S7-300 da Siemens com interface PROFIBUS DP integrada. A CPU comprada é do modelo 313C-2 DP e conta com uma memória principal com capacidade para armazenar 32 KBytes. Possui ainda 16 canais de entradas digitais e 16 canais de saídas digitais. A interface PROFIBUS DP integrada suporta 32 escravos por estação e velocidade de transmissão de 12 MBit/s. O novo CLP é capaz de se comunicar com os CLPs S7-200 (escravos) pré-existentes. Foi instalada no PC a plataforma de programação e monitoramento Step 7 do novo CLP, que se comunica com ele para *download*, *upload* e monitoramento através da rede profibus.

O *firmware* das HBs também foi alterado, de modo que cada HB pode funcionar independentemente. Assim sendo, pode-se implementar parte das seqüências operacionais em HBmain e parte em HBaux. A comunicação das HB só se dá com os CLPs S7-200 porque o CLP S7-300 não suporta

comunicação serial no modo *freepoint*, como protocolo definido pelo usuário. Com estas alterações, a configuração atual de hardware é a mostrada no diagrama da Figura 6.2.

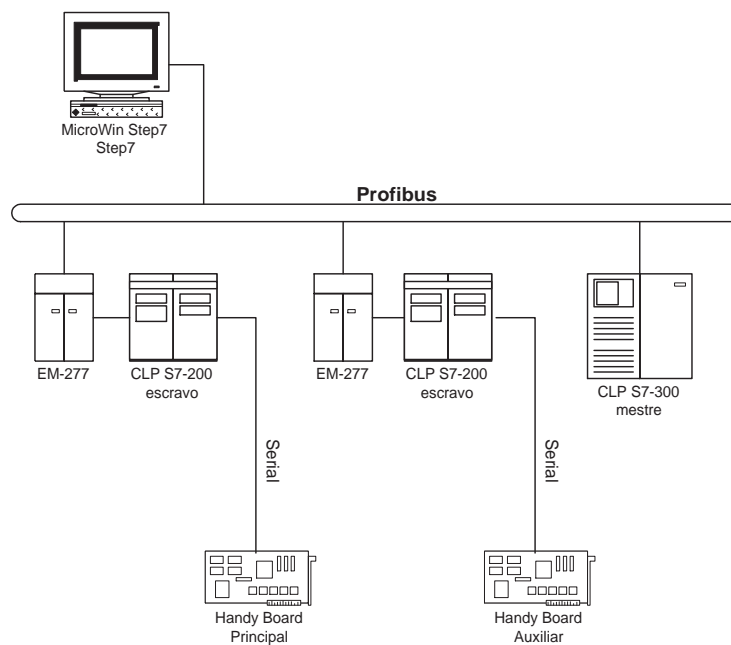


Figura 6.2: Configuração de *hardware* atual

6.1.3 PROFIBUS

Como já foi dito, buscou-se realizar a comunicação entre os CLPs na implementação do sistema supervisório distribuído através da rede PROFIBUS DP. Esta seção se propõe a abordar alguns aspectos básicos desta tecnologia, bastante utilizada em aplicações na indústria.

O padrão PROFIBUS é um sistema de comunicação digital aberto, com uma ampla gama de aplicações, principalmente nos campos industriais e de automação de processos. O PROFIBUS é adequado tanto para aplicações de tempo crítico quanto para tarefas de comunicação complexas.

O PROFIBUS especifica as características técnicas e funcionais de um sistema de comunicação industrial, através das quais dispositivos digitais podem se interconectar, desde o nível de campo até o nível de células. O PROFIBUS é um sistema multi-mestre e permite a operação conjunta de diversos sistemas de automação, engenharia ou visualização, com seus respectivos dispositivos periféricos (por ex. E/S).

O PROFIBUS diferencia seus dispositivos entre mestres e escravos. Dispositivos mestres determinam a comunicação de dados no barramento. Um mestre pode enviar mensagens, sem uma requisição externa, sempre que possuir o direito de acesso ao barramento. Os mestres também são chamados de estações ativas no protocolo PROFIBUS. Os dispositivos escravos são dispositivos remotos, tais como módulos de entrada e saída (E/S), válvulas, acionamentos de velocidade variável e transdutores. Eles não têm direito de acesso ao barramento e só podem enviar mensagens ao mestre ou reconhecer mensagens recebidas quando solicitados. Os escravos também são chamados estações passivas.

O perfil de comunicação PROFIBUS define como os dados serão transmitidos serialmente através do meio de comunicação. O perfil PROFIBUS DP (*Decentralized Periphery*) é o perfil mais frequentemente utilizado. Otimizado para alta velocidade e conexão de baixo custo, foi projetado especialmente para a comunicação entre sistemas de controle e automação e suas respectivas E/S distribuídas no nível de dispositivos [15].

O propósito de uma conexão PROFIBUS DP é a troca de dados. Esta troca de dados é sempre descrita em relação ao mestre. Os dados que são transferidos do mestre para o escravo são sempre considerados dados de saída. Os dados que são transferidos do escravo para o mestre são sempre considerados dados de entrada. No escravo, dados que são procedentes do mestre ainda assim são considerados dados de saída, ainda que eles sejam entradas para o escravo. Da mesma forma, dados que são retornados do escravo para o mestre são considerados dados de entrada, ainda que eles sejam saídas para o escravo.

Controladores PROFIBUS (CLPs / PCs) comunicam-se com seus dispositivos de campo distribuídos (E/S, drives, válvulas) através de um canal serial de alta velocidade. Grande parte desta comunicação com estes dispositivos distribuídos é cíclica. Isto significa que cada dispositivo é acessado em seqüência e a seqüência é repetida continuamente. Algumas funções como as de configuração, parametrização, diagnósticos e tratamento de interrupções ocorrem fora do ciclo normal de entrada e saída (E/S). Estas são as funções acíclicas.

A transmissão de dados entre o mestre e os escravos que estão associados a ele é realizada automaticamente pelo mestre em uma ordem recorrente e definida. Essa transferência de dados recorrente é conhecida como transferência cíclica de dados. Quando o sistema é configurado especifica-se quais escravos estão associados ao mestre e quais escravos devem ser incluídos ou excluídos da transferência cíclica de dados.

A transmissão de dados entre o mestre e o escravo é dividida em três fases: parametrização, configuração e troca de dados. As primeiras duas fases normalmente ocorrem apenas uma vez quando o sistema é inicializado ou quando um escravo é adicionado. A fase de troca de dados é o modo de operação normal do sistema. Neste modo, o mestre escreve saídas para o escravo e o escravo retorna entradas. Se o escravo não possui nenhuma saída, o mestre assim mesmo envia uma mensagem, sem nenhum dado de saída, e o escravo responde com as suas entradas. Se um escravo tem saídas mas nenhuma entrada, o escravo responde à mensagem de saída com uma mensagem que não contém dados.

O PROFIBUS DP permite sistemas com mestre único ou com múltiplos mestres. Até 126 dispositivos (mestre ou escravos) podem ser conectados para formar um sistema. A descrição da configuração do sistema consiste do número de estações, associação entre os endereços das estações e os endereços de E/S, formato dos dados de E/S, formato das mensagens de diagnóstico e parâmetros utilizados.

A quantidade de informações de entrada e saída depende do tipo de dispositivo. O PROFIBUS DP permite um máximo de 246 bytes de informação de entrada e 246 bytes de informação de saída

a serem transmitidas dentro de uma mensagem. A quantidade de dados transmitidos é normalmente específica a cada dispositivo. Muitos dispositivos suportam apenas um ou dois bytes de dados.

Do ponto de vista tecnológico, o nível mais baixo da estrutura do sistema da PROFIBUS é baseada no modelo de referência OSI/ISO [33]. Neste modelo, a camada 1 (nível físico) define as características físicas de transmissão, a camada 2 (data link layer) define o protocolo de acesso ao meio e a camada 7 (application layer) define as funções de aplicação. O PROFIBUS-DP usa somente as camadas 1 e 2, bem como a interface do usuário. As camadas 3 a 7 não são utilizadas. Esta arquitetura simplificada assegura uma transmissão de dados eficiente e rápida.

6.2 Problemas de Distribuição

São propostos e implementados três problemas distintos de distribuição. Procurou-se abordar situações e escolher configurações que representassem uma classe de problemas de distribuição. A cobertura destas configurações com relação à gama possível de problemas de distribuição ficou restrita ao *hardware* disponível, mas acredita-se que para o porte do sistema tratado grande parte das configurações de distribuição factíveis é abordada.

O problema de controle supervisório tratado é o mesmo apresentado na Seção 3.2, o que significa que o objetivo é realizar a distribuição das estruturas encontradas na Seção 4.3 ($S_a, S_{b1}, S_{b2}, S_{b3}, S_{b4}, S_{c1}, S_{c2}, S_{c3}, g_0, g_1, g_2, g_3, g_4, o_0, o_1, o_2, o_3, o_4$) nos dispositivos de controle disponíveis mostrados na Seção 6.1.2. As células de controle para a CEBE são dadas, segundo a Tabela 3.14, por:

$$CC_a = \{S_a, g_0, g_1, g_2, g_3\}$$

$$CC_{b1} = \{S_{b1}, g_0, g_1\}$$

$$CC_{b2} = \{S_{b2}, g_0, g_2\}$$

$$CC_{b3} = \{S_{b3}, g_0, g_3\}$$

$$CC_{b4} = \{S_{b4}, g_0, g_4\}$$

$$CC_{c1} = \{S_{c1}, g_0, g_1, g_2\}$$

$$CC_{c2} = \{S_{c2}, g_0, g_2, g_3\}$$

$$CC_{c3} = \{S_{c3}, g_0, g_3, g_4\}$$

Como pode ser observado, g_0 está presente em todas as células de controle. A seqüência de processamento sincronizada que inclui g_0 deve incluir também g_1, g_2, g_3 e g_4 . Desta forma, todos os supervisores também devem ser incluídos nesta seqüência de processamento sincronizado.

A existência de uma única seqüência de processamento sincronizado na CEBE indica um elevado grau de inter-relação entre os sistemas físicos. Observa-se que a mesa giratória (g_0), responsável pela

movimentação das garrafas entre as diversas estações (subsistemas), é a maior responsável por esta inter-relação entre os subsistemas.

Outro fator importante na realização da distribuição do sistema de controle é a análise das informações que devem ser transmitidas entre cada par de CLPs. Para a definição destas informações é necessário estabelecer quais são os eventos relevantes e quais são as desabilitações efetivas para cada par (S,G), onde S é um supervisor e G é um dos módulos do sistema produto que faz parte da célula de controle deste supervisor.

A Tabela 6.1 mostra os alfabetos de eventos relevantes e desabilitações efetivas para cada um dos pares (S_j, G_i) existentes na CEBE, onde o índice i define cada módulo do sistema produto e o índice j define cada um dos supervisores modulares locais reduzidos. A primeira coluna desta tabela traz os índices i e j que definem os pares (S_j, G_i) para os quais serão encontrados os eventos relevantes e as desabilitações efetivas nas outras duas colunas. Estes alfabetos foram encontrados segundo o método mostrado no exemplo 5.6.

Tabela 6.1: Alfabetos de eventos relevantes e desabilitações efetivas da CEBE

$i - j$	$\Sigma_{relev.i-j}$	$\Theta_{efet.i-j}$
0-a	$\{a_0\}$	$\{a_0d\}$
1-a	$\{a_1\}$	$\{\}$
2-a	$\{a_2\}$	$\{\}$
3-a	$\{a_3\}$	$\{\}$
0-b1	$\{a_0, b_0\}$	$\{a_0d\}$
1-b1	$\{a_1, b_1\}$	$\{a_1d\}$
0-b2	$\{a_0, b_0\}$	$\{a_0d\}$
2-b2	$\{a_2, b_2\}$	$\{a_2d\}$
0-b3	$\{a_0, b_0\}$	$\{a_0d\}$
3-b3	$\{a_3, b_3\}$	$\{a_3d\}$
0-b4	$\{a_0, b_0\}$	$\{a_0d\}$
4-b4	$\{a_4, b_4\}$	$\{a_4d\}$
0-c1	$\{a_0\}$	$\{a_0d\}$
1-c1	$\{b_1\}$	$\{a_1d\}$
2-c1	$\{a_2\}$	$\{a_2d\}$
0-c2	$\{a_0\}$	$\{a_0d\}$
2-c2	$\{b_2\}$	$\{a_2d\}$
3-c2	$\{a_3\}$	$\{a_3d\}$
0-c3	$\{a_0\}$	$\{a_0d\}$
3-c3	$\{b_3\}$	$\{a_3d\}$
4-c3	$\{a_4\}$	$\{a_4d\}$

Os eventos externos relevantes e as desabilitações externas relevantes são obtidos por meio da aplicação das funções fee_i e fde_i , respectivamente, aos elementos contidos nos alfabetos mostrados na Tabela 6.1.

A escolha das configurações implementadas levam em conta tanto a necessidade de distribuição devido à distribuição espacial do sistema físico quanto a necessidade ou opção em se modularizar o sistema de controle. Estas três implementações são apresentadas nas seções que se seguem em ordem crescente de complexidade. Cada seção traz a apresentação da configuração do sistema de controle a ser implementado assim como a solução do problema de distribuição. Para todas as configurações considera-se que o CLP S7-300 é o Dispositivo A, um dos CLPs S7-200 é o Dispositivo B, o outro CLP S7-200 é o Dispositivo C, a HB_{main} é o Dispositivo B1 e a HB_{aux} é o Dispositivo C1. Apenas na Implementação 1, na qual as HBs operadas conjuntamente são tratadas como um dispositivo único, ambas as HBs são consideradas um dispositivo único, Dispositivo B1. As três configurações foram implementadas na CEBE e testadas, funcionando corretamente.

6.2.1 Implementação 1

A primeira configuração de distribuição implementada, denominada aqui Implementação 1, foi realizada de forma que o sistema de controle fosse implementado em 3 níveis. Todos os supervisores foram implementados no CLP S7-300 (Dispositivo A), todos os módulos do sistema produto foram implementados em um CLP S7-200 (Dispositivo B) e todas as seqüências operacionais foram implementadas nas HBs. As duas HBs nessa implementação operam conjuntamente já que se comunicam internamente e apenas a HB_{main} se comunica com o CLP. Portanto, podem ser consideradas como um dispositivo de controle único (Dispositivo B1), do ponto de vista dos CLPs.

Para a implementação 1 são propostas as seguintes partições dos conjuntos S , G e O :

$$[S]_A = \{S_a, S_{b1}, S_{b2}, S_{b3}, S_{b4}, S_{c1}, S_{c2}, S_{c3}\}; \quad [S]_B \text{ e } [S]_{B1} \text{ não definidas}$$

$$[g]_B = \{g_0, g_1, g_2, g_3, g_4\}; \quad [g]_A \text{ e } [g]_{B1} \text{ não definidas}$$

$$[o]_{B1} = \{o_0, o_1, o_2, o_3, o_4\}; \quad [o]_A \text{ e } [o]_B \text{ não definidas}$$

Para as partições dos conjuntos S e G apresentadas acima tem-se que $S_A = [S]_A$, $G_A = [g]_A$, $S_B = [S]_B$ e $G_B = [g]_B$ são os conjuntos de supervisores e o conjunto de módulos do sistema produto associados, respectivamente, ao Dispositivo A e ao Dispositivo B e que estão na seqüência de processamento sincronizado que deve ser estabelecida.

Observa-se que os componentes de S_A definem células de controle com os componentes de G_B . Assim sendo, não pode haver simultaneidade de processamento dos componentes dos dois CLPs. Pela condição fundamental de seqüenciamento a única seqüência de processamento sincronizado possível é dada por AB, cujo grafo de distribuição horizontal é mostrado na Figura 6.3.

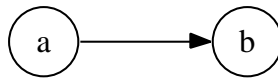


Figura 6.3: Grafo de distribuição horizontal da Implementação 1

Para esta implementação, observa-se, segundo os três itens que devem ser levados em consideração na definição da troca de dados entre os CLPs apresentados na Seção 5.2.3.1, que do Dispositivo A, no qual estão implementados todos os supervisores, para o Dispositivo B, onde estão todos os módulos do sistema produto, devem ser enviados todos os dados referentes às desabilitações efetivas externas, segundo a Tabela 6.1. Do Dispositivo B para o Dispositivo A devem ser enviados todos os dados referentes aos eventos relevantes externos mostrados na Tabela 6.1. Nenhuma sinalização de evolução externa necessita ser trocada entre os CLPs, uma vez que todos os módulos do sistema produto estão implementados em um mesmo dispositivo.

É necessária a criação de dois canais de comunicação ligando o Dispositivo A ao Dispositivo B e vice-versa, ou um canal de comunicação bi-direcional entre estes dispositivos. Os pacotes de informação $\{E_{AB}, D_{AB}, V_{AB}\}$ e $\{E_{BA}, D_{BA}, V_{BA}\}$ devem ser transmitidos do Dispositivo A para o Dispositivo B e do Dispositivo B para o Dispositivo A, respectivamente, onde:

- $E_{AB} = \{ \}$; $D_{AB} = \{a_0d_e, a_1d_e, a_2d_e, a_3d_e, a_4d_e\}$; $V_{AB} = \{ \}$
- $E_{BA} = \{a_0e, a_1e, a_2e, a_3e, a_4e, b_0e, b_1e, b_2e, b_3e, b_4e\}$; $D_{BA} = \{ \}$; $V_{BA} = \{ \}$

Com base nestes pacotes de informação observa-se que 5 dados são enviados do Dispositivo A para o Dispositivo B e 10 dados são enviados do Dispositivo B para o Dispositivo A, totalizando 15 informações a serem trocadas. Percebe ainda que, embora nenhuma sinalização de evolução de módulo do sistema produto seja trocada, todas as desabilitações e todos os eventos devem ser transmitidos de um CLP a outro.

O grafo de comunicação para esta implementação é mostrado na Figura 6.4.



Figura 6.4: Grafo de comunicação para Implementação 1

Para a conclusão da distribuição geral do sistema de controle conforme especificado para esta implementação, definida a etapa de distribuição horizontal, realiza-se a distribuição vertical do sistema de controle. Como pode ser observado por meio da partição do conjunto de seqüências operacionais para a Implementação 1, todas as seqüências operacionais estão implementadas no Dispositivo B1.

Realiza-se a distribuição vertical entre os Dispositivos B e B1. Nesta implementação, como já foi dito, as HBs operando conjuntamente são consideradas um dispositivo de controle único. Assim sendo, a distribuição vertical entre os Dispositivos B e B1 pode ser realizada conforme mostrado na

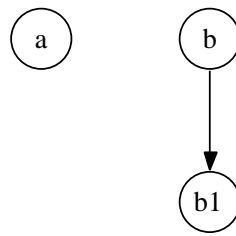


Figura 6.5: Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 1

Seção 5.1.1. O grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 1 é mostrado na Figura 6.5.

A Figura 6.6 mostra um diagrama com a divisão dos componentes do sistema de controle em cada um dos dispositivos, assim como os canais de comunicação entre eles.

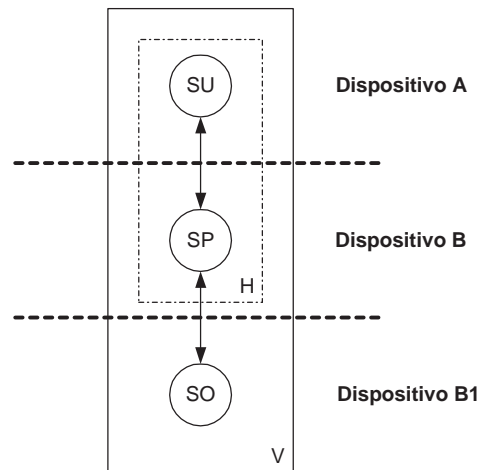


Figura 6.6: Diagrama Implementação 1

A Implementação 1 representa a solução do problema de controle supervisório para a classe de problemas em que não há distribuição espacial dos subsistemas físicos e/ou não é necessário preservar a modularidade natural do sistema físico e das especificações de comportamento. É um modelo de implementação modular do sistema de controle em 3 níveis, com as estruturas de cada nível do modelo de implementação alocadas em um dispositivo de controle diferente. Mostra ser uma forma prática e simples de resolver o problema de distribuição para os casos onde há escassez de recursos, como memória de programa do CLP.

6.2.2 Implementação 2

A segunda configuração de distribuição implementada, denominada aqui Implementação 2, foi realizada de forma que o sistema de controle possibilite a preservação da modularidade natural do sistema físico e das especificações. Alguns supervisores e alguns módulos do sistema produto foram implementados em um CLP S7-200 (Dispositivo B), enquanto que os demais supervisores e módulos do sistema produto foram implementados no outro CLP S7-200 (Dispositivo C). Parte das seqüências

operacionais foi implementada na HB_{main} (Dispositivo B1) e as demais na HB_{aux} (Dispositivo C1). Cada HB nessa implementação opera separadamente, já que se comunicam individualmente com o CLP no qual estão implementados os módulos do sistema produto correspondentes às seqüências operacionais nelas implementadas.

Para a Implementação 2 são propostas as seguintes partições dos conjuntos S , G e O :

$$\begin{aligned} [S]_B &= \{S_a, S_{b2}, S_{b3}, S_{c1}, S_{c2}, S_{c3}\}; & [S]_C &= \{S_{b1}, S_{b4}\}; & [S]_{B1} &\text{ e } [S]_{C1} &\text{ não definidas} \\ [g]_B &= \{g_2, g_3\}; & [g]_C &= \{g_0, g_1, g_4\}; & [g]_{B1} &\text{ e } [g]_{C1} &\text{ não definidas} \\ [o]_{B1} &= \{o_2, o_3\}; & [o]_{C1} &= \{o_0, o_1, o_4\}; & [o]_B &\text{ e } [o]_C &\text{ não definidas} \end{aligned}$$

Para as partições dos conjuntos S e G apresentadas acima tem-se que $S_B = [S]_B$, $G_B = [g]_B$, $S_C = [S]_C$ e $G_C = [g]_C$ são os conjuntos de supervisores e o conjunto de módulos do sistema produto associados, respectivamente, ao Dispositivo A e ao Dispositivo B e que estão na seqüência de processamento sincronizado que deve ser estabelecida.

Observa-se que os componentes de S_B definem células de controle com os componentes de G_C . Assim sendo, não pode haver simultaneidade de processamento dos componentes dos dois CLPs. Pela condição fundamental de seqüenciamento a única seqüência de processamento sincronizado possível é dada por BC, cujo grafo de distribuição horizontal é mostrado na Figura 6.7.

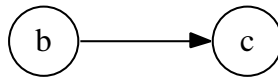


Figura 6.7: Grafo de distribuição horizontal da Implementação 2

Para esta implementação, observa-se, segundo os três itens que devem ser levados em consideração na definição da troca de dados entre os CLPs apresentados na Seção 5.2.3.1, que do Dispositivo B para o Dispositivo C devem ser enviados todos os dados referentes às desabilitações efetivas externas relacionadas a eventos controláveis dos módulos do sistema produto implementados no Dispositivo C, conforme a Tabela 6.1 e as partições dos conjuntos de supervisores e módulos do sistema produto. Além disso, algumas sinalizações de evolução externa necessitam ser enviadas do Dispositivo B para o Dispositivo C, uma vez que módulos do sistema produto que formam células de controle com os módulos do sistema produto que estão implementados no Dispositivo B estão implementados no Dispositivo C, que sucede o Dispositivo B na seqüência de processamento sincronizado. Do Dispositivo C para o Dispositivo B devem ser enviados alguns dos dados referentes aos eventos relevantes externos mostrados na Tabela 6.1, de acordo com as células de controle e com as partições dos conjuntos de supervisores e de módulos do sistema produto.

É necessária a criação de dois canais de comunicação ligando o Dispositivo B ao Dispositivo C e vice-versa, ou um canal de comunicação bi-direcional entre estes dispositivos. Os pacotes de informação $\{E_{BC}, D_{BC}, V_{BC}\}$ e $\{E_{CB}, D_{CB}, V_{CB}\}$ devem ser transmitidos do Dispositivo B para o Dispositivo C e do Dispositivo C para o Dispositivo B, respectivamente, onde:

- $E_{BC} = \{ \}; \quad D_{BC} = \{a_0d_e, a_1d_e, a_4d_e\}; \quad V_{BC} = \{g_2evt_e, g_3evt_e\}$
- $E_{CB} = \{a_{0e}, a_{1e}, a_{4e}, b_{0e}, b_{1e}\}; \quad D_{CB} = \{ \}; \quad V_{CB} = \{ \}$

Com base nestes pacotes de informação observa-se que 5 dados são enviados do Dispositivo B para o Dispositivo C e outros 5 dados são enviados do Dispositivo C para o Dispositivo B, totalizando 10 informações a serem trocadas. Percebe-se ainda que, embora algumas sinalizações de evolução de módulo do sistema produto sejam trocadas (ao contrário do que ocorre na Implementação 1), menos desabilitações e menos eventos devem ser transmitidos de um CLP a outro comparando-se a Implementação 2 com a Implementação 1.

Apesar de não ser utilizado para a implementação das estruturas do sistema de controle supervisório distribuído, o CLP S7-300 é necessário nesta Implementação 2. Os dois CLPs S7-200 devem trocar dados pela rede Profibus. No entanto, ambos são CLPs escravos. Por esta razão houve a necessidade de inserção do CLP S7-300 (mestre) para que a troca de dados se viabilizasse. O CLP S7-300 não foi utilizado para implementar as estruturas do sistema de controle já que não é possível realizar a comunicação deste dispositivo com as HBs, conforme mencionado na Seção 6.1.2, e para esta implementação esta capacidade é necessária para ambos os CLPs. Visto isto, do ponto de vista do sistema de controle, a existência deste CLP pode ser desconsiderada.

O grafo de comunicação para esta implementação é mostrado na Figura 6.8.



Figura 6.8: Grafo de comunicação para Implementação 2

Para a conclusão da distribuição geral do sistema de controle conforme especificado para esta implementação, definida a etapa de distribuição horizontal, realiza-se a distribuição vertical do sistema de controle. Como pode ser observado por meio da partição do conjunto de seqüências operacionais para a Implementação 2, uma parte das seqüências operacionais está implementada no Dispositivo B1 enquanto que a outra parte está implementada no Dispositivo C1.

Realiza-se a distribuição vertical entre os Dispositivos B e B1 e entre os Dispositivos C e C1. Nesta implementação, como já foi dito, as HBs operam independentemente. Assim sendo, a distribuição vertical entre os Dispositivos B e B1 e entre os Dispositivos C e C1 é realizada nos moldes daquela mostrada na Seção 5.1.1, com apenas as seqüências operacionais dadas pela partição do conjunto de seqüências operacionais implementadas nas respectivas HBs. O funcionamento das HBs no que diz respeito à comunicação com os CLPs e à execução das seqüências operacionais permanece o mesmo. O grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 2 é mostrado na Figura 6.9.

A Figura 6.10 mostra um diagrama com a divisão dos componentes do sistema de controle em cada um dos dispositivos, assim como os canais de comunicação entre eles.

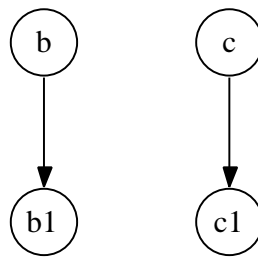


Figura 6.9: Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 2

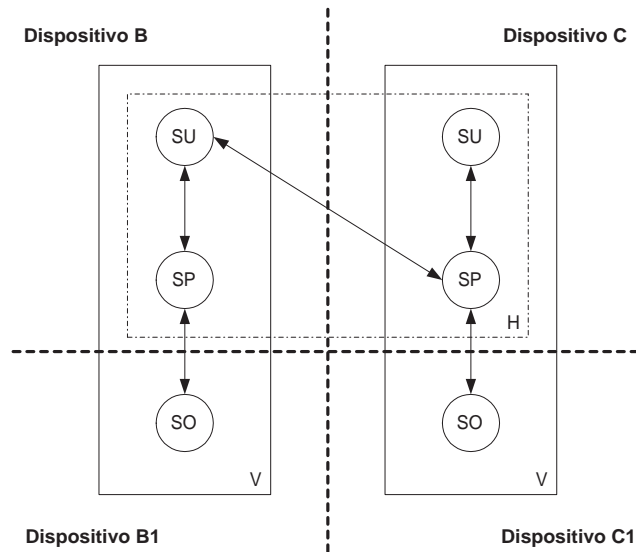


Figura 6.10: Diagrama Implementação 2

A Implementação 2 representa a solução do problema de controle supervisório para a classe de problemas em que há distribuição espacial dos subsistemas físicos e/ou é necessário preservar a modularidade natural do sistemas físico e das especificações de comportamento. É um modelo de implementação modular do sistema de controle em 2 níveis, com as estruturas dos supervisores e dos módulos do sistema produto alocadas em dispositivos de controle de um nível e as estruturas das seqüências operacionais alocadas em dispositivos de um outro nível. Portanto, não é uma arquitetura que tem como foco principal a modularização completa do sistema de controle. Mostra ser uma forma simples e econômica, do ponto de vista da utilização de dispositivos de controle, de resolver o problema de distribuição para os casos onde as seqüências operacionais de vários subsistemas já tenham sido previamente implementadas e otimizadas, como é o caso da CEBE.

6.2.3 Implementação 3

A terceira configuração de distribuição implementada, denominada aqui Implementação 3, foi realizada de forma que o sistema de controle possibilite a preservação da modularidade natural do sistema físico e das especificações. A distribuição do sistema de controle também foi realizada de forma modular, em 3 níveis. Todos os supervisores foram implementados no CLP S7-300 (Disposi-

tivo A), alguns módulos do sistema produto foram implementados em um CLP S7-200 (Dispositivo B), enquanto que os demais módulos do sistema produto foram implementados no outro CLP S7-200 (Dispositivo C). Parte das seqüências operacionais foi implementada na HB_{main} (Dispositivo B1) e as demais na HB_{aux} (Dispositivo C1). Cada HB nessa implementação opera independentemente, assim como na Implementação 2.

Para a Implementação 3 são propostas as seguintes partições dos conjuntos S , G e O :

$$\begin{aligned}
 [S]_A &= \{S_a, S_{b1}, S_{b2}, S_{b3}, S_{b4}, S_{c1}, S_{c2}, S_{c3}\}; & [S]_B, [S]_C, [S]_{B1} \text{ e } [S]_{C1} &\text{ não definidas} \\
 [g]_B &= \{g_2, g_3\}; & [g]_C &= \{g_0, g_1, g_4\}; & [g]_A, [g]_{B1} \text{ e } [g]_{C1} &\text{ não definidas} \\
 [o]_{B1} &= \{o_2, o_3\}; & [o]_{C1} &= \{o_0, o_1, o_4\}; & [o]_A, [o]_B \text{ e } [o]_C &\text{ não definidas}
 \end{aligned}$$

Para as partições dos conjuntos S e G apresentadas acima tem-se que $S_A = [S]_A$, $G_A = [g]_A$, $S_B = [S]_B$, $G_B = [g]_B$, $S_C = [S]_C$ e $G_C = [g]_C$ são os conjuntos de supervisores e o conjunto de módulos do sistema produto associados, respectivamente, ao Dispositivo A, ao Dispositivo B e ao Dispositivo C e que estão na seqüência de processamento sincronizado que deve ser estabelecida.

Observa-se que os componentes de S_A definem células de controle com os componentes de G_B e G_C , assim como componentes de G_B formam células de controle com componentes de G_C . Assim sendo, não pode haver simultaneidade de processamento dos componentes dos três CLPs. Pela condição fundamental de seqüenciamento, duas seqüências de processamento sincronizado são possíveis: ABC e ACB. Escolheu-se a seqüência dada por ABC, cujo grafo de distribuição horizontal é mostrado na Figura 6.11.

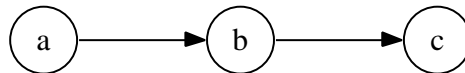


Figura 6.11: Grafo de distribuição horizontal da Implementação 3

Para esta implementação, observa-se, segundo os três itens que devem ser levados em consideração na definição da troca de dados entre os CLPs apresentados na Seção 5.2.3.1, que do Dispositivo A para o Dispositivo B (Dispositivo C) devem ser enviados todos os dados referentes às desabilitações efetivas externas relacionadas a eventos controláveis dos módulos do sistema produto implementados no Dispositivo B (Dispositivo C), conforme a Tabela 6.1 e as partições dos conjuntos de supervisores e módulos do sistema produto. Além disso, algumas sinalizações de evolução externa necessitam ser enviadas do Dispositivo B para o Dispositivo C, uma vez que módulos do sistema produto que formam células de controle com os módulos do sistema produto que estão implementados no Dispositivo B estão implementados no Dispositivo C, que sucede o Dispositivo B na seqüência de processamento sincronizado. Do Dispositivo B para o Dispositivo A e do Dispositivo C para o Dispositivo A devem ser enviados alguns dos dados referentes aos eventos relevantes externos mostrados na Tabela 6.1, de acordo com as células de controle e com as partições dos conjuntos de supervisores e de módulos do sistema produto.

É necessária a criação de dois canais de comunicação bi-direcionais sendo um ligando o Dispositivo A ao Dispositivo B e outro ligando o Dispositivo A ao Dispositivo C. Além disso, deve ser criado um outro canal uni-direcional ligando o Dispositivo B ao Dispositivo C. As seguintes informações deverão ser transmitidas nos pacotes de informação que serão trocados entre os CLPs:

- $E_{AB} = \{ \}; D_{AB} = \{a_2d_e, a_3d_e\}; V_{AB} = \{ \}$
- $E_{BA} = \{a_{2e}, a_{3e}, b_{2e}, b_{3e}\}; D_{BA} = \{ \}; V_{BA} = \{ \}$
- $E_{AC} = \{ \}; D_{AC} = \{a_0d_e, a_1d_e, a_4d_e\}; V_{AC} = \{ \}$
- $E_{CA} = \{a_{0e}, a_{1e}, a_{4e}, b_{0e}, b_{1e}, b_{4e}\}; D_{CA} = \{ \}; V_{CA} = \{ \}$
- $E_{BC} = \{ \}; D_{BC} = \{ \}; V_{BC} = \{g_2evt_e, g_3evt_e\}$

Com base nestes pacotes de informação observa-se que um total de 17 informações devem ser trocadas entre os CLPs. Percebe-se que a adição de mais um dispositivo ao sistema de controle aumenta consideravelmente a necessidade de se realizar esta troca de informações, o que em sistemas de grande porte pode inclusive resultar em um aumento considerável do fluxo de dados na rede. No entanto, para problemas de pequeno e médio porte como a CEBE, este tipo de problema não foi observado.

Ao contrário da Implementação 2, na Implementação 3 o CLP S7-300 é utilizado para a implementação das estruturas do sistema de controle supervisório distribuído. Além de viabilizar a troca de informações na rede Profibus atuando como mestre este CLP implementa todas as estruturas de supervisores modulares. O CLP S7-300 não foi utilizado para implementar nenhum módulo do sistema produto já que, como já foi dito, não é possível realizar a comunicação deste dispositivo com as HBs.

O grafo de comunicação para esta implementação é mostrado na Figura 6.12.

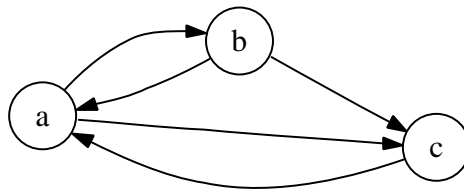


Figura 6.12: Grafo de comunicação para Implementação 3

Para a conclusão da distribuição geral do sistema de controle conforme especificado para esta implementação, definida a etapa de distribuição horizontal, realiza-se a distribuição vertical do sistema de controle. Como pode ser observado por meio da partição do conjunto de seqüências operacionais para a Implementação 3, uma parte das seqüências operacionais está implementada no Dispositivo B1 enquanto que a outra parte está implementada no Dispositivo C1.

Realiza-se a distribuição vertical entre os Dispositivos B e B1 e entre os Dispositivos C e C1. Nesta implementação as HBs operam independentemente e a distribuição vertical entre os Dispositivos B e B1 e entre os Dispositivos C e C1 é realizada da mesma forma que foi feita na Implemen-

tação 2. O grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 3 é mostrado na Figura 6.13.

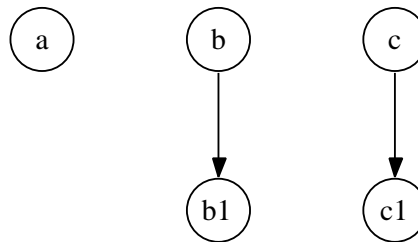


Figura 6.13: Grafo do segundo nível de representação da distribuição geral do sistema de controle da Implementação 3

A Figura 6.14 mostra um diagrama com a divisão dos componentes do sistema de controle em cada um dos dispositivos, assim como os canais de comunicação entre eles.

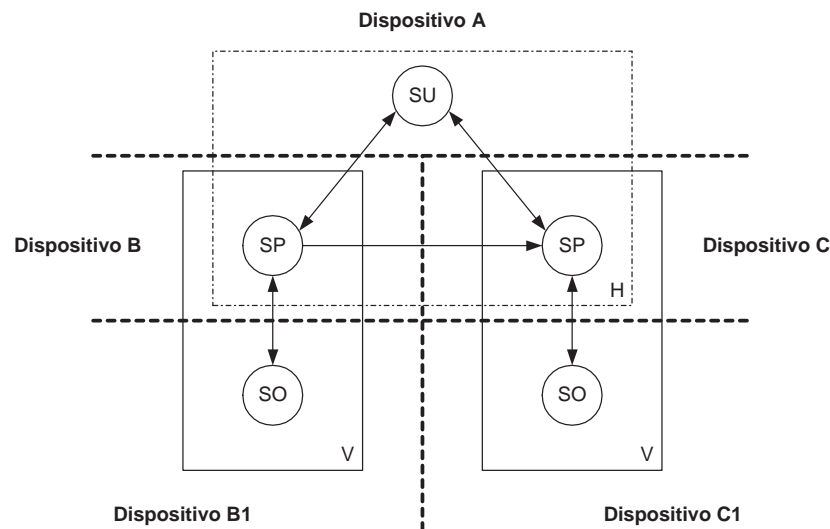


Figura 6.14: Diagrama Implementação 3

A Implementação 3 representa a solução do problema de controle supervisório para a classe de problemas em que há distribuição espacial dos subsistemas físicos e/ou é necessário preservar a modularidade natural do sistemas físico e das especificações de comportamento. Além disso, é um modelo de implementação modular do sistema de controle em 3 níveis, com as estruturas de cada nível do modelo de implementação alocadas em dispositivos de controle que estão em níveis diferentes, como pode ser visto na Figura 6.14. Portanto, é uma arquitetura que tanto preserve a modularidade natural do sistema físico e das especificações quanto realiza a modularização completa do sistema de controle. Por esta razão, dentre as três implementações realizadas neste trabalho, pode ser considerada a mais completa.

6.3 Análise para Projeto de Sistemas Distribuídos

Nesta seção é proposta uma análise a ser realizada para o projeto de sistemas de controle supervisório distribuídos. Esta análise leva em consideração alguns critérios que devem ser avaliados para posterior escolha da configuração do sistema distribuído a ser implantada. É realizada ainda a análise, segundo estes critérios propostos, das três configurações de distribuição apresentadas na Seção 6.2 para algumas situações hipotéticas. Por fim, mostra-se através de um exemplo como escolher uma configuração de distribuição que mais se adequa a um problema.

6.3.1 Critérios de Avaliação

Para a escolha da configuração adequada a um determinado problema de distribuição é importante que seja realizada uma avaliação das possíveis configurações segundo alguns critérios. São trazidos aqui os critérios que, após a realização da implementação distribuída do sistema de controle supervisório, tema desta dissertação, foram avaliados como sendo os de maior relevância para esta escolha. Cabe ressaltar, no entanto, que os critérios aqui levantando não são os únicos, e as possibilidades de avaliação segundo outros parâmetros não devem ser descartadas, ficando a adoção destes novos parâmetros de análise a critério do projetista. A importância de cada um desses critérios para a escolha de uma configuração em detrimento das demais é completamente dependente do problema tratado, ou seja, a relevância dos critérios está sujeita aos requisitos do projeto, recursos disponíveis, viabilidade técnica e financeira, disponibilidade de meios de comunicação, condições físicas e espaciais, dentre outras. Os critérios propostos para avaliação das configurações de distribuição são dados a seguir.

Simplicidade: a forma como se distribui o sistema de controle pode simplificar ou tornar mais complexo o seu desenvolvimento. A simplicidade do projeto é dependente tanto do número de dispositivos de controle envolvidos quanto da distribuição dos módulos do sistema de controle em cada um deles. A concentração das estruturas de todos os supervisores no primeiro dispositivo de controle da seqüência de processamento sincronizada, por exemplo, é uma decisão de projeto que implica em uma análise positiva do sistema em questão no quesito simplicidade. Pode-se afirmar também que quanto mais dispositivos de controle são utilizados no sistema de controle distribuído, menor a simplicidade de projeto do mesmo.

Custos com Equipamentos: este critério de avaliação diz respeito ao capital gasto com equipamentos que o sistema de controle distribuído demanda se projetado de uma determinada forma. Normalmente os maiores gastos ficam por conta dos dispositivos de controle, normalmente CLPs, cujo custo médio gira em torno dos R\$10.000,00, variando para mais ou para menos de acordo com o fabricante e as funcionalidades oferecidas por cada modelo. Portanto, deve-se sempre avaliar a viabilidade técnica e financeira da utilização de um CLP com mais recursos ao invés de dois ou mais CLPs com menos recursos. Esta análise deve sempre ser feita levando em consideração a questão dos impactos nos outros parâmetros de análise. Outros equipamentos,

como os dispositivos de rede, também devem ser analisados, não devendo ser relevados a segundo plano. Em alguns casos, a parcela gasta com esses dispositivos pode impactar de maneira significativa os custos do projeto. Questões como aproveitamento de redes pré-existentes, substituição ou criação de novas redes devem sempre estar em pauta nesta análise.

Custos com Cabos: este é um critério que tem maior ou menor relevância dependendo do processo no qual o sistema de controle distribuído será implementado. Em sistemas que ocupam uma região geográfica ampla este critério apresenta grande relevância, o que não ocorre em sistemas cuja área seja restrita. Verifica-se a tendência do aumento dos gastos com cabo quanto mais concentrado for o sistema de controle. Isto porque é necessário interligar os instrumentos de campo (sensores e atuadores) com seus respectivos dispositivos de controle, o que será tanto menos custoso quanto mais próximo estiver localizado este dispositivo de controle destes instrumentos. Outras vezes, o distanciamento do dispositivo de controle dos instrumentos de campo pode impor a utilização de meios de interligação alternativos, como por exemplo comunicação sem fio, aumentando os custos. No caso deste critério, a adição de dispositivos de controle no sistema de controle distribuído normalmente induz uma redução dos gastos com cabos.

Funcionamento Parcial: em muitos casos, mesmo com a falha de um dispositivo de controle o sistema de controle distribuído pode continuar funcionando parcialmente, sem infringir as condições impostas pelas especificações de funcionamento. Em outras situações pode ser desejável, ou necessário, a manutenção do funcionamento parcial do sistema mesmo que parte dele esteja inativada devido a algum procedimento de manutenção. Desta forma, é importante avaliar o quanto uma determinada configuração de distribuição possibilita este funcionamento parcial do sistema de controle. Neste caso, quanto mais distribuídos estiverem os módulos do sistema de controle maiores as possibilidades de funcionamento parcial do sistema. Considere um CLP no qual esteja implementado apenas um módulo do sistema produto. No caso de uma manutenção, pode-se desabilitar este dispositivo de controle mantendo o estado deste módulo do sistema produto fixo, de modo que a parte do sistema que independe deste módulo do sistema produto poderá continuar evoluindo normalmente.

Fluxo de Dados: a implementação de um sistema de controle supervisório distribuído implica, obrigatoriamente, na troca de informações entre os dispositivos de controle envolvidos. No entanto, o volume de dados a serem trocados é inteiramente dependente da configuração de distribuição escolhida. Para sistemas de médio e até grande porte, estes dados isoladamente não apresentam risco de congestionamento da rede, uma vez que a tecnologia atual permite a transmissão de até alguns gigabits por segundo. Porém, no caso desses dados terem que ser transmitidos por uma rede já saturada, tal critério de avaliação passa a ter importância crucial. Pode inclusive apontar para mudanças das soluções de rede inicialmente consideradas, por exemplo, induzindo a adoção de uma nova rede.

6.3.2 Análise das Configurações de Implementação

Nesta seção é realizada uma análise das três configurações de implementação distribuída propostas na Seção 6.2 por meio da avaliação dos critérios apresentados na Seção 6.3.1 na aplicação dessas configurações a algumas situações levantadas. Não objetiva-se aqui a escolha de uma configuração específica para cada situação, uma vez que essa escolha depende das restrições de projeto e da importância de cada um dos critérios para a aplicação em questão. A avaliação dos critérios é apenas uma maneira de se identificar as vantagens e desvantagens de cada uma das configurações de implementação para posterior escolha da que mais se adequa ao problema de distribuição a ser resolvido.

SITUAÇÃO 1:

Considere que, finalizado o projeto do sistema de controle supervisório para uma determinada aplicação, o programa que implementa tal sistema possua um tamanho X ($X = 30\text{KB}$, por exemplo) e que a memória do dispositivo de controle disponível para a implementação deste sistema seja de $Y = X + \Delta$ ($Y = 16\text{KB}$ e $\Delta = 14\text{KB}$, por exemplo). Assim sendo, não existe a possibilidade de se implementar este sistema de controle apenas neste dispositivo, ou seja, é necessário realizar a distribuição. Considere ainda que os subsistemas que compõem o sistema a ser controlado encontram-se concentrados espacialmente. Apresenta-se na Tabela 6.2 a avaliação das três configurações de distribuição propostas na Seção 6.2 segundo os critérios de avaliação. Nesta tabela, as setas para cima (\uparrow) indicam avaliação positiva com relação ao critério e as setas para baixo (\downarrow) indicam avaliação negativa com relação ao critério. Setas duplas indicam avaliação mais positiva (ou negativa) da configuração em relação às demais.

Tabela 6.2: Avaliação das configurações para Situação 1

	Implementação 1	Implementação 2	Implementação 3
Simplicidade	$\uparrow\uparrow$	\downarrow	\uparrow
Custos com Equipamentos	$\downarrow\downarrow$	\downarrow	\uparrow
Custos com Cabos	-	-	-
Funcionamento Parcial	\downarrow	\uparrow	$\uparrow\uparrow$
Fluxo de Dados	\uparrow	\downarrow	$\uparrow\uparrow$

Para a Situação 1 a Implementação 1 seria a mais simples, por apresentar o sistema de controle distribuído em 3 níveis, com cada um dos níveis da estrutura de implementação implementados em um dispositivo de controle. A Implementação 2 também é simples, mas um pouco menos, já que o sistema de controle é distribuído também em 3 níveis, mas os níveis estão estratificados em mais de um dispositivo de controle. A Implementação 3 seria a mais complexa, por possuir estruturas de mais de um nível do modelo de implementação em um mesmo dispositivo de controle. Quanto aos custos com equipamentos pode-se dizer que a Implementação 1 é a mais vantajosa, uma vez que utiliza menos dispositivos de controle (3). As Implementações 2 e 3 com 4 e 5 dispositivos de controle, respectivamente, são avaliadas também conforme esta quantidade.

O critério de custos com cabos não foi avaliado, uma vez que os subsistemas que formam o sistema a ser controlado estão espacialmente concentrados e, portanto, todos os dispositivos de controle envolvidos no sistema de controle distribuído estariam localizados próximos aos instrumentos de campo com os quais se comunicam. Quanto ao critério de funcionamento parcial percebe-se que Implementação 3 é a mais eficiente, por possuir o sistema de controle mais estratificado, o que possibilita, em caso de falha de um dispositivo de controle (desde que o estado após a falha seja o mesmo que a estrutura se encontrava antes da falha) ou mesmo manutenção que o sistema continue operando sem perder a sincronização, até que o problema/manutenção seja finalizado. A avaliação das outras implementações é realizada também de acordo com a estratificação do sistema de controle. O critério de fluxo de dados é melhor avaliado para a Implementação 2, uma vez que o sistema de controle é implementado em 2 níveis, o que reduz a necessidade de troca de dados entre as estruturas de implementação em cada dispositivo de controle. A Implementação 3, por ter o sistema de controle implementado em 3 níveis, além de uma estratificação entre mais de um dispositivo de controle em cada nível, é a que apresenta o pior desempenho neste quesito.

SITUAÇÃO 2:

Suponha agora uma situação na qual os subsistemas que formam o sistema a ser controlado estejam espacialmente separados, como ilustra a Figura 6.15

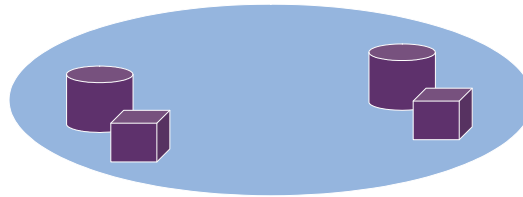


Figura 6.15: Representação do sistema da Situação 2

Para esta situação, além dos critérios já avaliados para a Situação 1, é importante considerar o critério de custos com cabos (Tabela 6.3). A Implementação 2 é a que apresenta os menores custos com cabos, uma vez que para esta configuração é possível alocar os dispositivos de controle próximos aos instrumentos com os quais trocam dados. Além disso, supervisores e módulos do sistema produto estão em um mesmo dispositivo, razão pela qual a Implementação 2 foi melhor avaliada que a Implementação 3, que precisa de cabos para interligar os dispositivos onde estão os módulos do sistema produto e os dispositivos onde estão os supervisores. A Implementação 1 é a que apresenta os maiores custos, uma vez que neste caso alguns dos instrumentos de campo ficariam distantes dos dispositivos de controle com os quais trocam dados, já que as seqüências operacionais estão concentradas em um único dispositivo de controle.

Ainda considerando esta mesma situação, pode ser que não seja possível a transmissão de dados entre as duas regiões nas quais estão localizados os subsistemas do sistema a ser controlado (Figura 6.16). Isso pode ocorrer tanto devido a barreiras físicas (que impedem a transmissão via cabos) quanto devido a interferências eletromagnéticas (que impedem a transmissão via ondas eletromagnéticas). Neste caso a Implementação 1 deve ser descartada como opção de configuração de distribuição, uma vez que nessas condições ela é inviável.

Tabela 6.3: Avaliação das configurações para Situação 2

	Implementação 1	Implementação 2	Implementação 3
Simplicidade	↑↑	↓	↑
Custos com Equipamentos	↓↓	↓	↑
Custos com Cabos	↑	↓↓	↓
Funcionamento Parcial	↓	↑	↑↑
Fluxo de Dados	↑	↓	↑↑

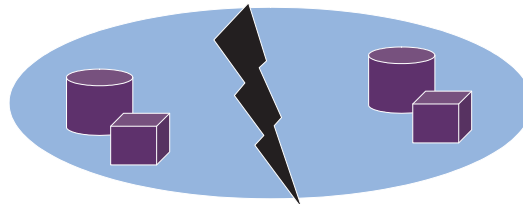


Figura 6.16: Representação do sistema da Situação 2 com barreira de comunicação

SITUAÇÃO 3:

Na Situação 3 os subsistemas que compõem o sistema a ser controlado estão distribuídos espacialmente, não existe a possibilidade de comunicação entre as duas áreas onde se concentram os subsistemas e, além disso, alguns subsistemas são críticos e devem ter o seu processamento priorizado em relação aos demais. A Figura 6.17 ilustra este panorama.

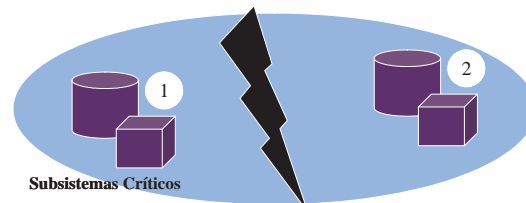


Figura 6.17: Representação do sistema da Situação 3

Neste caso, será feita somente a análise das Implementações 2 e 3, uma vez que devido à barreira de comunicação entre as áreas, a Implementação 1 fica inviabilizada. A Tabela 6.4 mostra a análise segundo os critérios de avaliação.

Observa-se que a Implementação 3 tem o critério simplicidade bem melhor avaliado que a Implementação 2, uma vez que neste caso os módulos do sistema produto estão alocados em vários dispositivos de controle isoladamente, e os supervisores estão alocados em outro dispositivo de controle, em outro nível do sistema de controle. Portanto, a questão da priorização de processamento dos módulos do sistema produto relacionados a algum subsistema é facilmente resolvida com o correto posicionamento do dispositivo de controle na seqüência de processamento sincronizado. Já no caso da Implementação 2 isto não ocorre, já que supervisores e módulos do sistema produto estão alocados em um mesmo dispositivo de controle. Neste caso, o posicionamento do dispositivo de controle na seqüência de processamento sincronizado torna-se muito mais complexo, chegando a ser algumas vezes inviável. Os custos com equipamentos da Implementação 3 continuam sendo maiores, uma vez

Tabela 6.4: Avaliação das configurações para Situação 3

	Implementação 2	Implementação 3
Simplicidade	↓	↑↑
Custos com Equipamentos	↓	↑
Custos com Cabos	-	-
Funcionamento Parcial	↑	↑↑
Fluxo de Dados	↓	↑

que mais dispositivos de controle são utilizados. Não é necessária uma análise com relação aos custos com cabos, uma vez que nas duas configurações em questão é possível posicionar os dispositivos de controle próximos aos instrumentos de campo com os quais se relacionam. Ambas as implementações possibilitam o funcionamento parcial do sistema, com vantagem para a Implementação 3, que apresenta o sistema de controle implementado de maneira mais estratificada. Pela mesma razão apresentada para as outras situações, na Implementação 3 o fluxo de dados é maior, o que faz com que a Implementação 2 seja mais vantajosa considerando-se este critério.

6.3.3 Exemplo de Aplicação da Distribuição

Será apresentado nesta seção um exemplo de aplicação da distribuição do sistema de controle a um sistema hipotético. Considere que para a realização desta distribuição sejam disponibilizados 5 CLPs, cada um com 16 Kb de memória. O sistema a ser controlado possui três grupos de subsistemas separados espacialmente, onde as prioridades de processamento dos subsistemas de cada um dos grupos é diferente. Suponha que o projeto do sistema de controle resultou em supervisores cujas estruturas de implementação ocupam 25kb de memória no total. As estruturas de implementação dos módulos do sistema produto relativas aos subsistemas que compõem cada região é menor que 16kB em cada uma dessas regiões. Suponha ainda que as seqüências operacionais são implementadas nos próprios equipamentos. para este panorama propõe-se a utilização da configuração de distribuição mostrada na Figura 6.18. A seqüência de processamento sincronizado seria dada por ABCDE, satisfazendo assim a necessidade de se priorizar os subsistemas mais críticos.

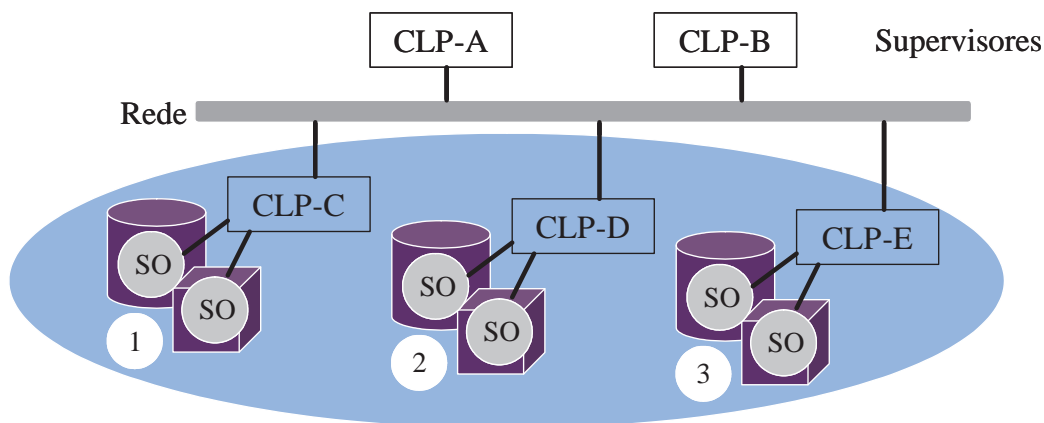


Figura 6.18: Representação do sistema de exemplo da aplicação da distribuição

Observa-se que o sistema de controle distribuído foi implementado em três níveis. No primeiro, as seqüências operacionais, implementadas nos próprios equipamentos. No segundo, os módulos do sistema produto, implementados em dispositivos de controle próximos aos subsistemas cujas seqüências operacionais eles se relacionam. No terceiro nível, os supervisores, implementados em dois dispositivos de controle devido a restrições de memória. Percebe-se que neste exemplo foram priorizados os critérios de simplicidade, custos com cabos e funcionamento parcial. Essa configuração é simples já que facilita a priorização de processamento dos subsistemas críticos, tornando trivial a obtenção da seqüência de processamento sincronizado. Os custos com cabos são também otimizados, uma vez que os CLPs que implementam os módulos do sistema produto localizam-se na mesma região dos equipamentos com os quais se relacionam. O funcionamento parcial também é possibilitado por esta configuração, e seria ainda mais eficiente caso houvesse mais de um CLP implementando módulos do sistema produto em cada região, o que no entanto aumentaria os gastos com equipamentos. A opção por se utilizar um CLP em cada região, apesar dos ganhos já salientados, trouxe um aumento nos gastos com equipamentos. Além disso, o sistema de controle implementado em três níveis também contribuiu para uma pior avaliação no critério fluxo de dados.

6.4 Estrutura do Programa

Os programas implementados têm características diferentes de acordo com o CLP em que estão hospedados e com os tipos de distribuição que realizam. Nos CLPs da família S7-200, em todas as implementações realizadas, é feita em um primeiro momento a distribuição vertical, que inclui a comunicação entre o CLP e a Handy-Board. Dentro deste procedimento é realizada a distribuição horizontal do sistema, que trata da comunicação entre os CLPs. No caso dos CLPs da família S7-300, como não é possível a comunicação com as Handy-Boards, não é viável a realização da distribuição vertical. Realiza-se, portanto, a distribuição horizontal nos moldes da implementada para os CLPs S7-200.

Descreve-se na seqüência, em linhas gerais, o funcionamento do programa residente nos CLPs S7-200. O detalhamento de cada uma das subrotinas não é aqui realizado já que o objetivo é dar uma visão geral de como funciona o programa que implementa um sistema de controle supervisório distribuído. A Figura 6.19 mostra o fluxograma do algoritmo implementado.

A subrotina *Initialize* é responsável por inicializar as variáveis globais utilizadas no programa e por preparar a primeira mensagem a ser enviada para a HB. Em seguida é executada a subrotina *Send*, que envia para a HB a mensagem previamente preparada através da interface serial. Após o envio da mensagem é habilitada a subrotina *Receive*, responsável pelo recebimento de mensagens provenientes da HB.

Se o recebimento da mensagem ocorrer dentro de um tempo pré-determinado, o programa segue o seu fluxo normal. Caso contrário, é chamada novamente a subrotina *Send* uma vez que nessa situação ou a HB não recebeu a última mensagem enviada pelo CLP ou a mensagem de resposta não foi enviada pela HB. Com esse procedimento o CLP estará, respectivamente, enviando a mensagem

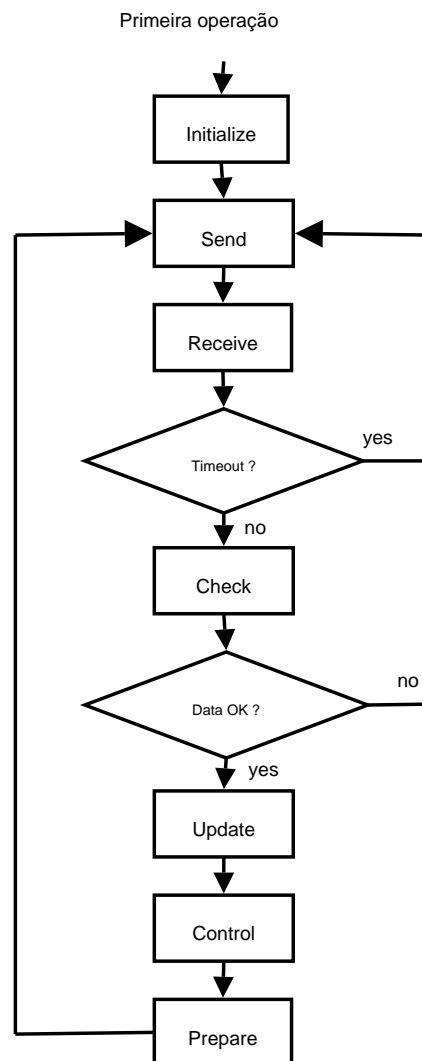


Figura 6.19: Fluxograma do algoritmo que realiza a Distribuição Vertical nos CLPs S7-200

não recebida pela HB ou comunicando à HB o não recebimento da mensagem de resposta, já que reenviará a última mensagem.

Dando continuidade ao fluxo normal do programa, é realizada uma verificação da integridade dos dados através do cálculo de CRC pela subrotina *Check*. Caso os dados sejam considerados íntegros e passarem nos testes o programa segue o fluxo normal. Caso contrário é chamada a subrotina *Send* com o propósito de solicitar que os dados corretos sejam reenviados pela HB.

Seguindo o fluxo natural do programa, após a certificação da integridade dos dados, realiza-se a interpretação da mensagem recebida na subrotina *Update*. Nesta subrotina identifica-se qual dos sinais de final de operação (respostas) foi enviado.

É chamada então a subrotina *Control*, onde o controle de alto nível da planta é efetivamente realizado. A implementação da distribuição horizontal do sistema de controle é realizada nesta subrotina e será detalhada posteriormente. Ainda dentro desta subrotina existe uma outra subrotina *Super*, na qual são implementadas as estruturas do sistema de controle (supervisores e módulos do sistema pro-

duto) que estão alocadas no respectivo CLP. Nota-se que alterações nestas estruturas irão ocasionar modificações apenas nesta parte do código.

Por último, finalizando um ciclo de operação do CLP, é chamada a subrotina *Prepare*, responsável pela preparação da próxima mensagem a ser enviada para a HB com os comandos a serem executados segundo o sistema de controle. O programa retorna então à subrotina *Send*, fechando o ciclo.

Conforme já foi previamente mencionado, dentro da subrotina *Control* é realizada a distribuição horizontal do sistema de controle. Optou-se por considerar aquilo que é tratado como um estado no autômato sincronizador como uma subrotina. As transições do autômato sincronizador são representadas por bits de permissão, que vão permitir ou não a execução de cada uma destas subrotinas. Em linhas gerais, a implementação horizontal é realizada como mostra a Figura 6.20.

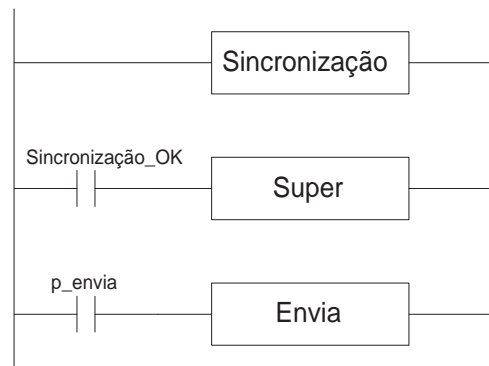


Figura 6.20: Estrutura básica do programa que realiza a Distribuição Horizontal nos CLPs S7-200

A subrotina *Sincronização* lê os dados dos buffers que contêm mensagens provenientes dos outros CLPs. Esta subrotina é executada seguidas vezes até que sejam recebidas informações atualizadas do CLP que antecede o CLP em questão na seqüência de processamento sincronizado. Enquanto isso não ocorre nenhuma outra subrotina é executada, inclusive aquelas que implementam a distribuição vertical. Esta parte da subrotina equivale ao estado W_k do autômato sincronizador.

Conforme mencionado na seção 6.1.3 a comunicação em uma rede Profibus é cíclica e ocorre de maneira recorrente e definida. Para que haja o reconhecimento e identificação de uma mensagem com conteúdo atualizado é necessário que o emissor e o receptor estabeleçam algum tipo de sincronia. No programa em questão este problema foi solucionado conforme mostrado na Figura 6.21.

O CLP fica sempre à espera de um tipo de sinal de sincronização (Sync1 ou Sync2) diferente do que foi recebido na última atualização. Caso ele receba o mesmo sinal de sincronização que está esperando do CLP que o antecede na seqüência de processamento sincronizado, significa que a mensagem recebida deste e dos demais CLPs já contém informações atualizadas. Desta forma, transfere-se as informações de cada um dos buffers que contêm mensagens provenientes de CLPs que formam canais de comunicação com o CLP em questão para variáveis internas deste CLP (função WOR_W). Esta parte da subrotina equivale ao estado S_k do autômato sincronizador. No exemplo da Figura 6.21 supõe-se que o CLP em questão forma apenas um canal de comunicação, com o CLP que o antecede na seqüência de processamento sincronizado. Feito isso, o CLP passa a esperar a variável de sincronização diferente da última que ele recebeu, além de setar a variável Sincronização_OK, que

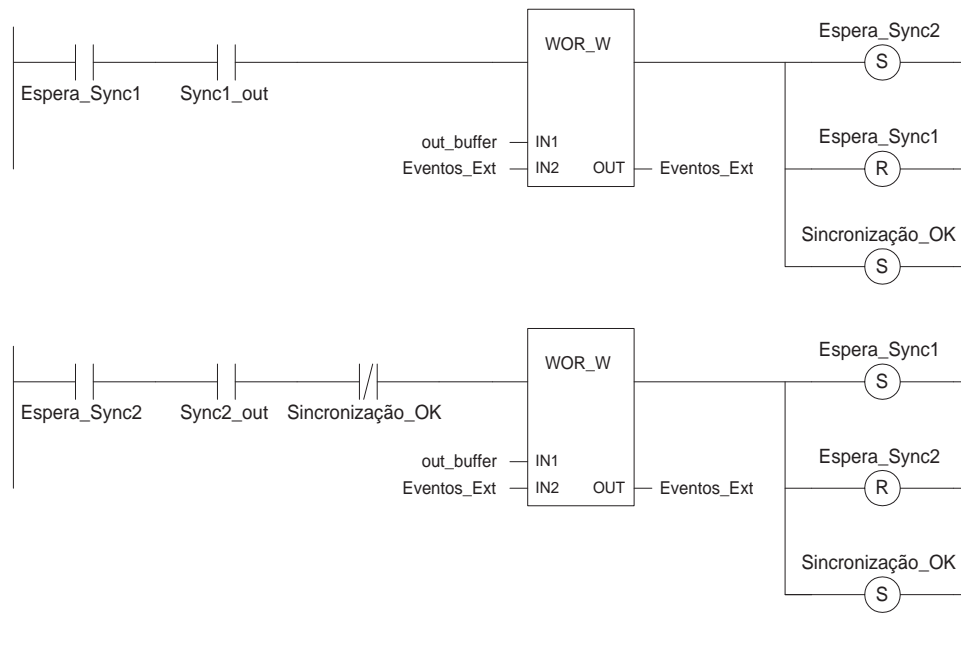


Figura 6.21: Lógica para sincronização

sinaliza que mais uma sincronização foi realizada com sucesso, permitindo que o programa continue com o seu fluxo normal.

Quando a variável **Sincronização_OK**, equivalente à transição CC_k do autômato sincronizador, está em nível lógico 1 permite-se que seja executada a subrotina *Super*. Esta subrotina equivale ao estado P_k do autômato sincronizador. Nela são implementadas as estruturas do sistema de controle (supervisores e módulos do sistema produto) conforme modelo mostrado na Seção 4.2.2. Quando o processamento de todas as estruturas é concluído seta-se a variável **p_envia**.

Com a variável **p_envia** em nível lógico 1 a subrotina *Envia* é executada. Nela, as variáveis de sincronização **Sync1** e **Sync2** são enviadas alternadamente para o CLP sucessor na seqüência de processamento sincronizado, possibilitando a manutenção da sincronização. Além disso, as informações atualizadas são enviadas para todos os CLPs que mantêm um canal de comunicação com o CLP em questão.

No CLP da família S7-300 não é realizada distribuição vertical. Desta forma, implementa-se somente a parte do programa referente à distribuição horizontal do sistema de controle. Ele é basicamente o mesmo implementado nos CLPs da família S7-200 como mostra a Figura 6.22.

Percebe-se que a única diferença da estrutura deste programa para a parte do programa referente à implementação da distribuição horizontal no CLP S7-200 está na subrotina de inicialização. No caso do CLP S7-200 essa inicialização de variáveis era feita antes mesmo de serem executadas as subrotinas que realizavam a distribuição vertical. No caso do CLP S7-300 essa inicialização de variáveis também deve ocorrer, e a subrotina *Inicialização*, responsável por isso, é executada no primeiro ciclo de processamento do CLP, quando a variável **First_Scan** está em nível lógico 1.

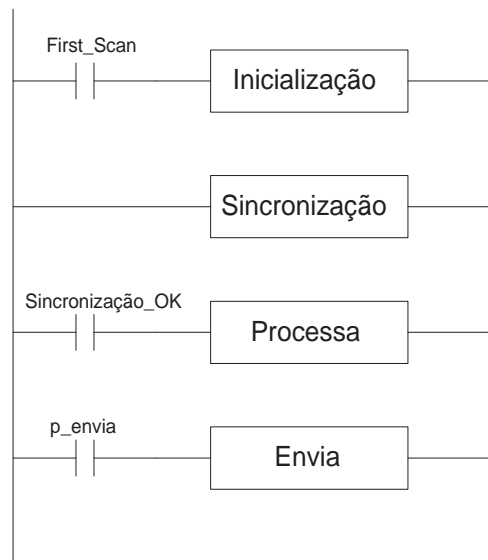


Figura 6.22: Estrutura básica do programa implementado nos CLPs S7-300

6.5 Conclusão do Capítulo

Este capítulo abordou todos os aspectos relacionados à implementação distribuída do controle supervisório na CEBE, planta utilizada neste trabalho. Em um primeiro momento foram levantadas as especificações do *hardware* disponível anteriormente e como está a configuração do *hardware* disponível atualmente. Foi feita também uma pequena explanação de tópicos básicos relativos às redes industriais do tipo Profibus.

Em um segundo momento foram propostos, resolvidos e implementados no sistema físico três problemas de distribuição com configurações distintas. Salienta-se que não se pode considerar alguma dessas configurações melhor que as outras, uma vez que cada uma delas atende bem a um determinado tipo de demanda, possuindo vantagens e desvantagens. Essas vantagens e desvantagens podem ser mensuradas por meio dos critérios de avaliação propostos para escolha da configuração de distribuição mais adequada. Ao final do capítulo, a estrutura dos programas implementados nos CLPs foi mostrada em linhas gerais.

O capítulo seguinte, último desta dissertação de mestrado, traz algumas considerações finais e as conclusões que puderam ser tiradas a partir deste trabalho.

Capítulo 7

Conclusões

A presente dissertação de mestrado apresenta a implementação distribuída bem sucedida de um sistema de controle supervisorio a um processo de manufatura real. Para isso foi realizada a distribuição do sistema de controle em vários dispositivos de controle. Tal procedimento implica em algumas considerações que devem ser feitas, principalmente no que diz respeito à comunicação dos dispositivos de controle e à sincronização dos programas de controle em execução em cada um dos dispositivos. Foram propostos também critérios para análise da configuração do sistema de controle supervisorio distribuído mais adequada aos mais variados tipos de problemas.

A resolução do problema proposto neste trabalho foi desenvolvida gradativamente, à medida que alguns conceitos relevantes iam sendo apresentados. Em um primeiro momento realizou-se a modelagem do sistema e das especificações de comportamento e em seguida a síntese dos supervisores modulares, segundo a abordagem de Controle Modular Local proposta por Queiroz [23]. Essa abordagem possibilita a redução dos esforços computacionais gastos no procedimento de síntese, assim como minimiza a quantidade de memória necessária para implementação dos componentes do sistema de controle nos dispositivos de controle. Isso se deve ao fato de a planta ser representada por um sistema produto e os supervisores serem calculados modular e localmente, resultando em supervisores com menor número de estados. Essa abordagem propicia também uma maior legibilidade da lógica de controle, o que facilita tanto a manutenção quanto o projeto destes sistemas.

Em um segundo momento aplicou-se o modelo para implementação do controle supervisorio à CEBE. Nesta etapa o auxílio da ferramenta CGLI foi de grande valia já que a transformação dos autômatos das estruturas do sistema de controle (módulos do sistema produto e supervisores) encontrados no procedimento de síntese (formato Grail) para linguagem de programação de CLPs (lista de instruções) foi realizada automaticamente e demandou bem menos tempo para ser realizada.

A distribuição vertical foi a primeira etapa de distribuição realizada sobre o sistema de controle supervisorio da CEBE. Na verdade, esta fase da distribuição já havia sido realizada de uma maneira informal e empírica. Como um dos objetivos deste trabalho era validar o modelo de distribuição proposto por Vieira [35], procurou-se descrever como esta distribuição havia sido feita adequando-a e adaptando-a ao modelo em questão.

Desejava-se que a comunicação entre os CLPs utilizados na implementação distribuída fosse realizada por meio da rede Profibus, já existente no Laboratório de Automação Industrial (LAI). Desta forma, buscava-se a utilização de tecnologias e ferramentas o mais próximas possíveis daquelas encontradas na indústria em aplicações reais. Contudo, isso só seria viabilizado caso um novo CLP, mestre da rede Profibus, fosse adquirido, uma vez que todos os CLPs disponíveis só trabalhavam como escravos da rede. Este CLP foi especificado e comprado e, devido à sua maior capacidade em relação aos já existentes, possibilita que futuramente possa-se realizar o controle de sistemas ainda mais complexos.

A partir destas novas configurações de *hardware* foi realizada a distribuição horizontal do sistema de controle. Neste caso, não foi feita apenas uma revisão bibliográfica do trabalho de Vieira [35] para que fosse realizada a implementação. Durante o processo de implementação foram notadas algumas restrições no referido modelo quanto à generalização de alguns conceitos e também quanto à forma com que algumas representações foram apresentadas. No intuito de contribuir com este trabalho, foi proposta uma forma mais abrangente de se representar os autômatos sincronizadores, completamente desvinculada da tecnologia utilizada para a comunicação. Propõe-se que a adequação à tecnologia seja realizada no próprio programa do dispositivo de controle, caso seja necessário. Além disso, a extração das informações referentes aos canais de comunicação do grafo de distribuição horizontal, por meio da proposta do grafo de comunicação, possibilitou maior abrangência de representação além de tornar o modelo mais simples e formalmente mais adequado.

Por fim, realizou-se a distribuição geral do sistema de controle supervisorio da CEBE segundo três configurações de distribuição distintas. Procurou-se abordar situações e escolher configurações que representassem uma classe de problemas de distribuição. A cobertura destas configurações com relação à gama possível de problemas de distribuição ficou restrita ao *hardware* disponível, mas acredita-se que para o porte do sistema tratado grande parte das configurações de distribuição factíveis foi abordada.

Não se pode dizer que das três configurações propostas exista alguma que seja melhor que as outras. Pode-se dizer que uma configuração é mais adequada a um determinado problema. O que vai determinar quais são as configurações possíveis é, em primeiro lugar, o *hardware* disponível. Em um segundo momento analisa-se a necessidade ou não de se manter a modularidade do sistema físico e das especificações ou ainda de se implementar o sistema de controle modularmente. Deve-se sempre ter em mente que a introdução de dispositivos no sistema de controle implica em custos e, portanto, tais escolhas devem ser realizadas de maneira extremamente criteriosa. Por essa razão foram propostos critérios para uma análise crítica da configuração de distribuição mais adequada para o sistema a ser distribuído. Procurou-se destacar os fatores julgados mais importantes a partir da experiência de implementação distribuída do sistema de controle supervisorio realizada neste trabalho, sendo eles a simplicidade, os custos com equipamentos, os custos com cabos, o funcionamento parcial e o fluxo de dados. Contudo, acredita-se que estes critérios não são os únicos e que a partir de novas experiências de implementação de sistemas deste tipo é bem provável que outros fatores importantes para a escolha da configuração de distribuição mais adequada venham à tona. A contribuição deste trabalho neste sentido é de fornecer os critérios de análise capazes de direcionar a escolha da configuração de distribuição mais adequada.

Acredita-se que este trabalho cumpriu com o seu objetivo principal, que era de validar o modelo proposto por Vieira [35] através da implementação distribuída do controle supervisorio em uma célula flexível real. Durante esse processo alguns aspectos de ordem prática e até mesmo teórica foram discutidos, trazendo algumas contribuições ao referido trabalho. Desta forma foi possível mostrar a aplicabilidade da TCS e, em especial, do Controle Modular Local na resolução de um problema real. A proposição de critérios para análise crítica das soluções de implementação é também uma contribuição importante deste trabalho, por servir como um suporte para projetistas de sistemas distribuídos até então inexistente.

A CEBE, sistema tratado no desenvolvimento deste trabalho, pode ser considerada um processo de pequeno porte. Apesar da metodologia utilizada poder ser aplicada a sistemas de qualquer dimensão, supõe-se que algumas questões podem surgir no tratamento de problemas de grande porte. Uma destas questões diz respeito ao fluxo de dados na rede que realiza a comunicação dos dispositivos de controle. Em sistemas grandes pode ser que a troca de informações cause um aumento considerável do fluxo de dados na rede. Sugere-se, desta forma, a aplicação desta metodologia em sistemas de médio/grande porte no intuito de se avaliar praticamente tais questões. Já estão em desenvolvimento abordagens que consideram que uma seqüência operacional não está necessariamente associada a um módulo do sistema produto. Sugere-se que a metodologia de distribuição tratada nesta dissertação seja estendida para que atenda sistemas desenvolvidos segundo esta abordagem.

Referências Bibliográficas

- [1] S. Abdelwahed e W.M. Wonham. Interacting DES: Modelling and Analysis. In *IEEE International Conference on Systems, Man and Cybernetics*, volume 5, pages 4222–4229, Oct. 2003.
- [2] G. Bouzon, M.L.C. de Oliveira, M.B.R. Vallim, J.P.C. Lacombe, G.M. Freitas, J.E.R. Cury, e J.-M. Farines. CEBE: Uma Plataforma para Experimentação Real Aplicada ao Ensino de Sistemas a Eventos Discretos. In *Anais do XV Congresso Brasileiro de Automática, CBA 2004*, Gramado, RS, Brasil, Set. 2004.
- [3] B. A. Brandin e W. M. Wonham. Modular Supervisory Control of Timed Discrete-Event Systems. In *Proceedings of the 32nd Conference on Decision and Control*, Dec. 1993.
- [4] B.A. Brandin. The Real-Time Supervisory Control of an Experimental Manufacturing Cell. *IEEE Transactions on Robotics and Automation*, 12(1):1–14, 1996.
- [5] C.G. Cassandras e S. Lafortune. *Introduction to Discrete Event Systems*. Kluwer Academic Publishers, 1999.
- [6] F. Charbonnier, H. Alla, e R. David. The Supervised Control of Discrete-Event Dynamic Systems. *IEEE Transactions on Control Systems Technology*, 7(2):175–187, Mar. 1999.
- [7] E. Chen e S. Lafortune. On Nonconflicting Languages that Arise in Supervisory Control of Discrete Event Systems. *Systems & Control Letters*, 17:105–113, 1991.
- [8] Y.-L. Chen e S. Lafortune. Design of Nonblocking Modular Supervisors Using Event Priority Functions. *IEEE Transactions on Automatic Control*, 45(3):432–452, Mar. 2000.
- [9] J.E.R. Cury. Teoria de Controle Supervisório de Sistemas a Eventos Discretos. Minicurso no V Simpósio Brasileiro de Automação Inteligente, Nov. 2001. <http://www.das.ufsc.br/~cury/cursos/apostila.pdf>.
- [10] M. Fabian e A. Hellgren. PLC-based Implementation of Supervisory Control of Discrete Event Systems. In *Proceedings of the 37th IEEE Conference on Decision and Control, CDC'98*, Tampa, Florida, USA, Dec. 1998.
- [11] P. Gohari e W.M. Wonham. On the Complexity of Supervisory Control Design in the RW Framework. *IEEE Transactions on Systems, Man and Cybernetics - Part B*, 30(5):643–652, Oct. 2000.

- [12] A. Hellgren, M. Fabian, e B. Lennartson. Modular Implementation of Discrete Event Systems as Sequential Function Charts Applied to an Assembly Cell. In *Proceedings of the 2001 IEEE Conference on Control Applications*, Mexico City, Mexico, Sep. 2001.
- [13] A. Hellgren, M. Fabian, e B. Lennartson. On the Execution of Discrete Event Systems as Sequential Function Charts. In *Proceedings of the 2001 IEEE Conference on Control Applications*, Mexico City, Mexico, Sep. 2001.
- [14] A. Hellgren, B. Lennartson, e M. Fabian. Modelling and PLC-based Implementation of Modular Supervisory Control. In *Proceedings of the 6th Workshop on Discrete Event Systems, WODES'02*, pages 371–376, Zaragoza, Spain, 2002.
- [15] IEC. International Standard IEC 61158, Digital Data Communications for Measurement and Control - Fieldbus for Use in Industrial Control Systems, 2003.
- [16] ISO/IEC. International Standard IEC 61131-3, Programmable Controllers - Part 3: Programming Languages, 2003.
- [17] S.C. Lauzon, J.K. Mills, e B. Benhabib. An Implementation Methodology for the Supervisory Control of Flexible Manufacturing Workcells. *Journal of Manufacturing Systems*, 16(2):91–101, 1997.
- [18] R. Malik. On the set of certain conflicts of a given language. In *Proceedings of the 7th Workshop on Discrete Event Systems, WODES'04*, Sep. 2004.
- [19] F. A. C. Silva Neto. Projeto e Implementação de Controles de Sistema a Eventos Discretos em uma Unidade de Potência Hidráulica. Relatório de estágio, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, 2004.
- [20] F.A.C. Silva Neto, R.B. Souto, e V.J. de Negri. Comparação entre Métodos de Projeto de Controle de Sistemas a Eventos Discretos e sua Implementação em CLP. In *Anais do XI CREEM - UERJ*, Nova Friburgo, RJ, Brasil, 2004.
- [21] M. Nourelfath e E. Niel. Modular Supervisory Control of an Experimental Automated Manufacturing System. *Control Engineering Practice*, 12(2):205–216, Feb. 2004.
- [22] P.N. Pena. *Explorando Aspectos Estruturais na Síntese de Supervisores para Sistemas Compostos*. Monografia de qualificação, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC, Brasil, 2005.
- [23] M.H. Queiroz. Controle Supervisório Modular de Sistemas de Grande Porte. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Mar. 2000.
- [24] M.H. Queiroz. *Controle Supervisório Modular e Multitarefa de Sistemas Compostos*. Tese de Doutorado, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC, Brasil, Maio 2004.
- [25] M.H. Queiroz e J.E.R. Cury. Modular Control of Composed Systems. In *Proceedings of the American Control Conference, ACC'00*, pages 4051–4055, June 2000.

- [26] M.H. Queiroz e J.E.R. Cury. Modular Supervisory Control of Large Scale Discrete Event Systems. In *Proceedings of the 5th Workshop on Discrete Event Systems, WODES'00*, volume 1, pages 103–110, Ghent, Belgium, Aug. 2000. Massachusetts : Kluwer Academic Publishers.
- [27] M.H. Queiroz e J.E.R. Cury. Controle Supervisório Modular de Sistemas de Manufatura. *Revista Controle & Automação*, 13(2):115–125, Ago. 2002.
- [28] M.H. Queiroz, E.A.P. Santos, e J.E.R. Cury. Síntese Modular do Controle Supervisório em Diagrama Escada para uma Célula de Manufatura. In *Anais do V Simpósio Brasileiro de Automação Inteligente*, Gramado, RS, Brasil, Jun. 2001.
- [29] P.J. Ramadge e W.M. Wonham. Supervisory Control of a Class of Discrete Event Processes. *SIAM Journal Control and Optimization*, Jan. 1987.
- [30] P.J.G. Ramadge e W.M. Wonham. The Control of Discrete Event Systems. In *Proceedings IEEE, Special Issue on Discrete Event Dynamic Systems*, volume 77, pages 81–98, Jan. 1989.
- [31] K. Rohloff e S. Lafortune. On the Complexity of the Verification of Modular Discrete-Event Systems. In *Proceedings of the 41st IEEE Conference on Decision and Control, CDC'02*, Las Vegas, Nevada, USA, Dec. 2002.
- [32] R.B. Souto. Projeto de Sistemas Automáticos com Modelagem e Controle da Comunicação com o Ambiente Externo. Dissertação de Mestrado, Universidade Federal de Santa Catarina, Florianópolis, SC, Brasil, Mar. 2005.
- [33] A.S. Tanenbaum. *Computer Networks*. Prentice Hall, Upper Saddle River, USA, fourth edition, 2003.
- [34] A.F. Vaz e W.M. Wonham. On Supervisor Reduction in Discrete-Event Systems. *International Journal of Control*, 44(2):475–491, 1986.
- [35] A.D. Vieira. *Contribuições à Implementação de Estruturas de Controle Supervisório*. Monografia de qualificação, Universidade Federal de Santa Catarina, UFSC, Florianópolis, SC, Brasil, 2004.
- [36] A.D. Vieira, J.E.R. Cury, e M.H. de Queiroz. Distribuição da Estrutura de Controle Supervisório de Sistemas a Eventos Discretos. In *Anais do XV Congresso Brasileiro de Automática, CBA 2004*, Gramado, RS, Brasil, Set. 2004.
- [37] A.D. Vieira, J.E.R. Cury, e M.H. de Queiroz. Distributed Supervisory Control of Discrete Event Systems Based on Local Modular Control. In *Anais do VI Induscon*, Joinville, SC, Brasil, Out. 2004.
- [38] Y. Willner e M. Heymann. Supervisory Control of Concurrent Discrete-Event Systems. *International Journal of Control*, 54:1143–1169, 1991.
- [39] K.C Wong, J.G. Thistle, R.P Malhamé, e H.-H Hoang. Supervisory Control of Distributed Systems: Conflict Resolution. *Discrete Event Dynamic Systems: Theory and Applications*, 10: 131–186, 2000.

-
- [40] K.C. Wong e W.M. Wonham. Modular Control and Coordination of Discrete-Event Systems. *Discrete Event Dynamical Systems*, 8(3):247–297, 1998.
- [41] W.M. Wonham. Notes on Control of Discrete-Event Systems. Dept. of Electrical & Computer Engineering, University of Toronto, 2003.
- [42] W.M. Wonham e P.J. Ramadge. On the Supremal Controllable Sublanguage of a Given Language. *SIAM Journal Control and Optimization*, May 1987.
- [43] W.M. Wonham e P.J. Ramadge. Modular Supervisory Control of Discrete Event Systems. *Math. Control, Signals Systems*, 1:13–30, 1988.