



**Universidade Técnica de Lisboa**

**Instituto Superior Técnico**

# **Supervisão de Sistemas de Automação com aplicação a uma Célula de Fabricação Flexível**

Hélder Ribeiro 46543 Controlo e Robótica  
Ricardo Elvas 46676 Controlo e Robótica

**LICENCIATURA EM ENGENHARIA ELECTROTÉCNICA E DE COMPUTADORES**  
**Relatório de Trabalho Final de Curso**  
**15/2002/L**

Prof. Orientador: Professor Doutor João José dos Santos Sentieiro  
Prof. Acompanhante: Doutor Paulo Jorge Coelho Ramalho Oliveira

Janeiro de 2004



*Though this be madness, yet there is method in't.*  
William Shakespeare, Hamlet

***Aos nossos pais, irmãs e amigos.***



# Agradecimentos

Servem estas palavras, para agradecer a todos aqueles que, directa ou indirectamente, contribuíram para a realização deste trabalho.

Agradecemos ao prof. Paulo Oliveira a oportunidade que nos concedeu, confiando-nos este trabalho com total apoio e disponibilidade. Reconhecemos as sugestões, críticas e palavras de incentivo, sempre oportunas.

Um palavra especial ao Eng. João Serralha, pela inteira disponibilidade na resolução dos problemas que surgiram, pelos favores prestados e por todos os momentos de boa disposição.

Ao prof. João Sentieiro, agradecemos a disponibilização dos meios que tornaram possível a realização do trabalho.



# Resumo

Neste trabalho propõe-se uma arquitectura para a supervisão de sistemas de automação, com incidência na detecção e recuperação de falhas. O sistema supervisionado consiste numa célula de fabricação flexível, composta por dois manipuladores robóticos, três máquinas CNC, quatro tapetes de transporte e respectivos controladores.

A fim de possibilitar a implementação da arquitectura de supervisão, apresenta-se um estudo exaustivo do sistema de eventos discretos que modela a célula de fabricação flexível a supervisionar, bem como das propriedades estruturais e topológicas da mesma.

A arquitectura proposta baseia-se na enumeração e classificação de uma série de falhas que podem ocorrer. Desenvolve-se uma metodologia para a recuperação da célula de fabricação flexível baseada em resultados recentes de controlo de Redes de Petri. Estes resultados possibilitam a síntese directa de controladores de supervisão para Redes de Petri que obedecem a um conjunto de restrições convexas, em que se garantem propriedades da Rede de Petri total resultante.

A arquitectura proposta, onde se acrescenta um posto remoto de design e produção, foi implementada e validada em tempo real na supervisão da célula de fabricação flexível, mostrando-se que este é um passo sólido na utilização da metodologia proposta para a detecção e recuperação de falhas em sistemas de eventos discretos.



# Conteúdo

<b>Agradecimentos</b>	<b>iii</b>
<b>Resumo</b>	<b>v</b>
<b>Lista de Figuras</b>	<b>x</b>
<b>Lista de Tabelas</b>	<b>xi</b>
<b>1 Introdução</b>	<b>1</b>
<b>2 A CFF</b>	<b>5</b>
2.1 Os elementos da <i>CFF</i>	5
2.1.1 Braços Robot	6
2.1.2 Controladores	6
2.1.3 Tapetes	7
2.1.4 As Máquinas CNC da <i>CFF</i>	7
2.1.5 O PC - Aplicação de Controlo	8
2.2 Arquitectura Existente	8
2.3 Arquitectura Proposta	9
<b>3 Redes de Petri</b>	<b>11</b>
3.1 Conceitos Básicos	11
3.1.1 Notação e Definição	11
3.1.2 Dinâmica das RdP	12
3.1.3 Propriedades Comportamentais	12
3.1.4 Propriedades Estruturais	13
3.2 Modelação e Controlo de Sistemas de Eventos Discretos	14
3.2.1 Transições não controláveis e não observáveis	17
3.2.2 Síntese de controladores para RdPs com transições não controláveis e não observáveis	18
3.2.3 Restrições Lineares Generalizadas	20
3.2.4 Análise da Admissibilidade das Restrições Lineares Generalizadas	21
3.2.5 Transformação de Restrições Lineares Generalizadas	23
3.2.6 Algoritmo para a transformação de uma restrição numa restrição admissível	24
3.3 Métodos de Análise	25
3.3.1 Análise Algébrica Linear	25
3.3.2 Grafo de Cobertura	26
3.3.3 Técnicas de redução	26
<b>4 Modelação e Controlo da CFF</b>	<b>29</b>
4.1 Modelação dos Elementos da <i>CFF</i>	29
4.1.1 Tapetes	29
4.1.2 Máquinas CNC	31
4.1.3 Braços Robot	32

4.2	Imposição de Restrições de Controlo . . . . .	35
4.2.1	Interacção entre <i>B2</i> e os tapetes . . . . .	35
4.2.2	Interacção entre <i>B1</i> e os tapetes <i>T1</i> e <i>T2</i> . . . . .	36
4.2.3	Interacção entre o sequenciador e os restantes componentes . . . . .	37
4.2.4	Interacção entre robot <i>B1</i> e as máquinas . . . . .	39
4.3	Análise de Propriedades . . . . .	40
4.3.1	Análise dos Invariantes da Rede . . . . .	40
4.3.2	Grafo de Alcançabilidade . . . . .	41
<b>5</b>	<b>Supervisão da <i>CFF</i> - Modelação das Falhas</b>	<b>43</b>
5.1	Falhas da <i>CFF</i> . . . . .	44
5.1.1	Braços Robot . . . . .	44
5.1.2	Máquinas . . . . .	47
5.2	Análise da RdP resultante . . . . .	50
5.2.1	Análise dos Invariantes da Rede . . . . .	50
5.2.2	Grafo de Alcançabilidade . . . . .	51
<b>6</b>	<b>Supervisão da <i>CFF</i> - Implementação Prática</b>	<b>53</b>
6.1	Arquitectura de Comunicações . . . . .	53
6.2	Posto Remoto de <i>Design</i> e Produção . . . . .	54
6.3	Monitorização, Detecção e Recuperação de Falhas . . . . .	56
6.3.1	Monitor . . . . .	56
6.3.2	Detector de Falhas . . . . .	56
6.3.3	Recuperação . . . . .	58
6.4	Dificuldades de implementação . . . . .	60
<b>7</b>	<b>Conclusões e Desenvolvimentos Futuros</b>	<b>61</b>
<b>A</b>	<b>Modelação da <i>CFF</i></b>	<b>63</b>
<b>B</b>	<b>Modelação das Falhas</b>	<b>67</b>
<b>C</b>	<b>Implementação Prática</b>	<b>73</b>
<b>D</b>	<b>Fabrico Assistido por Computador CAD/CAM</b>	<b>77</b>
D.1	CAD . . . . .	77
D.2	CAM . . . . .	78
D.3	Tecnologia CNC . . . . .	80
<b>E</b>	<b>Braços Robot Scorbot - ER VII</b>	<b>83</b>
E.0.1	Robot . . . . .	83
E.0.2	Controlador . . . . .	83
<b>F</b>	<b>Máquinas da <i>CFF</i></b>	<b>87</b>
	<b>Bibliografia</b>	<b>89</b>

# Lista de Figuras

1.1	Estrutura do Relatório . . . . .	3
2.1	A <i>CFF</i> . . . . .	5
2.2	Aspecto geral da <i>CFF</i> . . . . .	5
2.3	Aspecto dos elementos constituintes da <i>CFF</i> . . . . .	6
2.4	Nova arquitectura da <i>CFF</i> . . . . .	9
3.1	Elementos de uma RdP . . . . .	12
3.2	RdPs básicas dos componentes . . . . .	16
3.3	RdP com o controlador . . . . .	17
3.4	RdP com o controlador . . . . .	22
3.5	Exemplo de uma Transformação-C . . . . .	23
3.6	Exemplo de uma Transformação-H . . . . .	24
3.7	4 simples transformações que preservam vivacidade, segurança e limitação. . . . .	27
3.8	RdP Simplificada . . . . .	27
3.9	Grafo de alcançabilidade da RdP da figura 3.8 . . . . .	27
4.1	RdP dos Tapetes <i>T1</i> e <i>T2</i> . . . . .	30
4.2	RdP dos Tapetes 3 e 4 . . . . .	30
4.3	RdP da máquina . . . . .	31
4.4	RdP do buffer <i>B2</i> . . . . .	32
4.5	RdP do braço <i>B2</i> . . . . .	32
4.6	RdP do braço <i>B1</i> . . . . .	33
4.7	RdP do Sequenciador . . . . .	34
4.8	Entrada de peças no sistema vindas do armazém . . . . .	35
4.9	Interação de <i>B2</i> com os tapetes . . . . .	36
4.10	Interação de <i>B1</i> com <i>T1/T2</i> . . . . .	37
4.11	Interação de <i>B1</i> com as máquinas . . . . .	40
4.12	Relação N° de peças/Nós do Grafo de alcançabilidade . . . . .	41
5.1	Extensões à RdP, correspondentes ao tratamento da falha $F_b1$ . . . . .	45
5.2	Grafo de alcançabilidade que descreve a ocorrência e recuperação de $F_b1$ . . . . .	45
5.3	Extensões à RdP correspondentes ao tratamento da falha $F_B2$ . . . . .	46
5.4	Grafo de alcançabilidade que descreve a ocorrência e recuperação de $F_B2$ quando o destino da matéria prima seria uma máquina . . . . .	47
5.5	Extensão à RdP correspondente ao tratamento da falha $F_M1$ . . . . .	49
5.6	Extensão à RdP correspondente ao tratamento da falha $F_M2$ . . . . .	49
5.7	Extensão à RdP correspondente ao tratamento da falha $F_M3$ . . . . .	49
5.8	Extensão à RdP correspondente ao tratamento da falha $F_M4$ . . . . .	49
5.9	Grafo de alcançabilidade que descreve a ocorrência e recuperação de $F_M1, F_M2, F_M3$ e $F_M4$ . . . . .	50
5.10	Relação N° de peças - Nós do Grafo de alcançabilidade . . . . .	51
6.1	Fluxo de informação entre o Controlador multitarefa e o Supervisor remoto . . . . .	53

6.2	Arquitectura de comunicações da <i>CFE</i> . . . . .	54
6.3	Diálogo de inicialização dos robots e/ou da <i>CFE</i> . . . . .	55
6.4	Janela da aplicação <i>Novo Modelo</i> . . . . .	55
6.5	Janela da aplicação <i>Sessão de Produção</i> . . . . .	56
6.6	Janela da aplicação <i>Posto de Design</i> . . . . .	56
6.7	RdP do robot <i>B2</i> modelada no <i>Monitor</i> . . . . .	57
6.8	Diagrama temporal do método de detecção de falhas associadas à ocorrência de <i>timeouts</i> . . . . .	57
6.9	Procedimentos do Detector de Falhas . . . . .	58
6.10	Seqüência de recuperação de $F_{B2}$ . . . . .	59
A.1	RdP controlada do sequenciador . . . . .	63
A.2	RdP controlada do sistema braços-tapetes . . . . .	64
A.3	RdP controlada das máquinas . . . . .	65
A.4	RdP controlada do robot <i>B1</i> . . . . .	66
B.1	Falha P-local . . . . .	67
B.2	Árvore de alcançabilidade da RdP da figura B.1 . . . . .	68
B.3	Falha local a um módulo . . . . .	69
B.4	Árvore de alcançabilidade da RdP da figura B.3 . . . . .	70
B.5	Falha Global . . . . .	71
B.6	Árvore de alcançabilidade da RdP da figura B.5 . . . . .	72
B.7	Grafo de alcançabilidade que descreve a ocorrência e recuperação de $F_{B2}$ quando o destino da matéria prima seria um <i>buffer</i> . . . . .	72
D.1	Diagrama das fases de um projecto CAD/CAM/CNC . . . . .	78
D.2	Vista de cima do modelo CAD da peça com as dimensões em milímetros . . . . .	79
D.3	Peça Bruta com as dimensões em milímetros . . . . .	79
D.4	Esquema do Suporte de fixação e da ferramenta (dimensões em milímetros) . . . . .	80
D.5	Caminho que a ferramenta irá efectuar no material - caminho feito pela ferramenta a preto . . . . .	80
D.6	Vista lateral da operação a efectuar pela ferramenta . . . . .	80
D.7	Simulador do Programa <i>Mcam9</i> . . . . .	81
D.8	Sistema de Controlo Numérico . . . . .	81
E.1	Limites da área de trabalho . . . . .	83
E.2	Esquema do Controlador do Braço . . . . .	84
E.3	Esquema do Controlador de cada eixo do Braço . . . . .	85

# Lista de Tabelas

4.1	Legenda da RdP dos Tapetes 1 e 2 . . . . .	30
4.2	Legenda da RdP dos Tapetes 3 e 4 . . . . .	30
4.3	Legenda da RdP dos Tapetes 1 e 2 . . . . .	31
4.4	Legenda da RdP do <i>buffer</i> . . . . .	32
4.5	Legenda a RdP de $B2$ . . . . .	32
4.6	Legenda da RdP de $B1$ . . . . .	33
4.7	Legenda da RdP do sequenciador . . . . .	34
4.8	Resultados da análise efectuada à RdP . . . . .	41
5.1	Falhas do Braço Robot . . . . .	44
5.2	Falhas nas Máquinas . . . . .	48
5.3	Resultados da análise efectuada à RdP . . . . .	51
6.1	Resultados da detecção e recuperação da falha $F_{B2}$ . . . . .	59
C.1	Eventos da <i>CFF</i> . . . . .	73
C.2	Acções do Supervisor . . . . .	74
C.4	Especificações das portas RS232 . . . . .	74
C.3	Sessão de Supervisão onde se recupera a falha $F_{B2}$ . . . . .	75
E.1	Sistema de coordenadas do robot . . . . .	83
E.2	Características do robot . . . . .	84
E.3	Legenda da figura E.2 . . . . .	85
E.4	Características do controlador do robot . . . . .	86
F.1	Especificações das portas RS232 . . . . .	87
F.2	Interface X22 - Interface de verificação e comando . . . . .	87
F.3	Especificações técnicas da fresadora EMCO-F1 . . . . .	88



# CAPÍTULO 1

## INTRODUÇÃO

Nas últimas décadas, assistiu-se ao aparecimento de um novo tipo de sistemas de manufactura, conhecidos por sistemas de fabricação flexível (SFF). A filosofia subjacente a este tipo de sistemas faz com que sejam indicados para soluções de baixo custo e de uma rápida adaptação ao tipo de produto e às dimensões dos lotes a produzir. Os sistemas de manufactura tradicionais proporcionam uma excelente relação custo-qualidade, sendo, no entanto, pouco versáteis. De acordo com os actuais padrões de competitividade, os sistemas de manufactura devem produzir de acordo com as conformidades e exigências dos clientes. Um *SFF* responde de forma adequada a estas necessidades, apresentando-se como um sistema de produção praticamente automatizado, com a capacidade de produzir uma grande variedade de diferentes produtos, apesar de utilizar o mesmo equipamento e o mesmo sistema de controlo. Exemplo real e concreto desta versatilidade é a *Motorola*, que, em 1999, produziu 29 milhões de variantes de um só produto.

Tipicamente, um *SFF* é constituído por várias células [Células de Fabricação Flexível (*CFF*)] espalhadas por diferentes espaços físicos. Cada *CFF* apresenta, pelo menos, três subsistemas:

- Sistema de Armazenamento e Processamento de material - constituído por equipamentos automatizados ou robotizados que fornecem e gerem material.
- Sistema de Processamento - conjunto de máquinas ferramenta de controlo numérico por computador (CNC).
- Sistema de Controlo por Computador - centro de decisão e controlo de todo o sistema.

A *CFF* do Laboratório de Sistemas de Controlo Numérico (LSCN), cedida ao Departamento de Engenharia Electrotécnica e de Computadores (DEEC) pelo Instituto de Sistemas e Robótica (ISR) e objecto da aplicação prática deste trabalho, inclui-se na classe dos *SFF* e possui todos os subsistemas acima referidos. Ao longo da última década, desenvolveram-se vários e meritórios esforços no sentido da evolução da *CFF* do LSCN. Ainda assim, ao contrário do que acontece na maioria das *CFFs* actuais, a arquitectura actual não contempla dois importantes componentes deste tipo de sistema: a supervisão do processo de fabrico e o desenho assistido por computador (CAD).

Neste trabalho propõe-se uma arquitectura para a supervisão de sistemas de automação, com incidência na detecção e recuperação de falhas. Pretende-se, também, implementar na *CFF*, do LSCN, a arquitectura proposta. Apresenta-se, à partida, uma dificuldade na implementação de um sistema deste tipo: o nível mais alto da hierarquia de controlo da *CFF* é "livre de erros", isto é, não prevê qualquer tipo de falha no funcionamento de qualquer dos seus elementos. A solução mais comum, para este tipo de problema, passa por acrescentar mais um nível à hierarquia de controlo ([16]). Neste novo topo de hierarquia são recebidos os eventos necessários à observação e monitorização do processo de fabrico e detectadas eventuais falhas. Depois de analisado o tipo de falha e decidida a recuperação, são enviados comandos para realização da mesma.

Para que a detecção e recuperação de falhas não altere as boas propriedades do sistema de controlo, já implementado, é imprescindível percorrer algumas etapas anteriores à implementação prática de um sistema deste tipo. O primeiro passo deve ser no sentido de estudar exaustivamente as sequências realizadas por cada um dos componentes da *CFF* e do sistema de controlo que os integra, para que o sistema seja modelado o mais fielmente possível. Com este conhecimento, é também possível melhorar alguns aspectos da *CFF*: actualização do sistema operativo do *PC* de controlo e um novo modelo para as comunicações série, entre outros. Depois disso, há que modelar o sistema, de forma a que, depois de analisado, se lhe possa acrescentar a detecção e recuperação de falhas.

O controlo de um sistema desta complexidade requer uma hierarquia multi-nível de controladores, desde os servo-mecanismos, no nível mais baixo, até a aplicações de *software* que implementem a necessária lógica de controlo, no nível mais alto. Os níveis mais baixos de controlo são, normalmente, baseados em modelos representados por Sistemas Dinâmicos de Variáveis Contínuas (*SDVC*), enquanto que, no nível mais alto, uma representação em termos de Sistemas de Eventos Discretos (*SED*) é, normalmente, mais adequada. A ligação

entre estes dois tipos de sistemas é feita por elementos intermédios de controlo, tipicamente PLC's ou outro tipo de controladores equivalentes, que efectuam a abstracção dos *SDVC* para os *SED*, catalogando o estado dos elementos do *SFF* em estados discretos que permitam ao controlador de alto-nível tomar as devidas decisões de controlo.

Um Sistema de Eventos Discretos é um sistema cujo espaço de estados é discreto e cujo estado só é alterado após ocorrência assíncrona de um evento. É a formalização que melhor se adapta a sistemas em que o espaço de estados é naturalmente descrito por um conjunto discreto de estados e a transição de estado só é observada em instantes de tempo discretos. É devido à ocorrência de um evento que se dá a transição de estado.

Os *SEDs* podem ser modelados fazendo uso de diversas metodologias como sejam os automatos, as linguagens e as Redes de Petri, entre outras. As Redes de Petri (RdP) permitem modelar, analisar, simular e observar um *SED*. Surgiram, no início dos anos sessenta, como uma alternativa ao *Automata* para avaliar o desempenho de *SEDs*. A grande vantagem deste paradigma é, para além da conveniente representação gráfica, o considerável conjunto de técnicas de análise desenvolvidas (apresentadas em [14] e [13]), baseadas numa teoria bem definida e sustentada, que permite o estudo das principais propriedades dos sistemas aos quais se aplica.

Até há alguns anos, a modelação de controladores para supervisão de *SED* baseados em RdPs era conseguida através de métodos mais ou menos intuitivos e, muitas vezes, baseada na adição de pequenas extensões de rede, bem conhecidas, para resolver os mais diversos problemas. Em [14] e [1] foram apresentadas algumas dessas técnicas, que permitem modelar sistemas em que vários componentes concorrem entre si, partilham os mesmos recursos e/ou apresentam sincronismo de várias acções. O propósito destes controladores é, generalizando, modificar e/ou restringir o comportamento de um *SED*, de forma a que cumpra um dado conjunto de especificações. Em [11], é apresentada uma nova forma de modelar e implementar o controlo de um *SED* modelado em RdPs. Este método, baseado no conceito de invariantes de marcação de uma RdP, formaliza matematicamente o que, muitas vezes, se obtém através de modelações empíricas. Esta técnica permite sintetizar, de forma automática, controladores para uma RdP, baseados nas suas propriedades estruturais e num conjunto de restrições lineares impostas ao seu comportamento. O resultado, é um modelo unificado do sistema e do seu controlador, que facilita a análise, a simulação, a verificação e a sua implementação.

Recorrendo a resultados recentes para a síntese de controladores baseados em invariantes de marcação ([11] e [9]), é possível modelar a *CFE* sob a forma de uma só RdP. Este modelo serve de base para a modelação da detecção e recuperação de falhas. Depois de estudadas e enumeradas as falhas mais frequentes da *CFE*, acrescentam-se, à RdP original, extensões que modelam a detecção e a recuperação das mesmas. Com isto, chega-se a um modelo unificado do sistema de controlo e da supervisão da *CFE*, integrando, numa mesma rede, os estados de funcionamento normal e os principais estados de falha e recuperação. Assim, é possível provar, teoricamente e por simulação exaustiva, a conservação das propriedades interessantes do sistema original e o correcto funcionamento das rotinas de detecção e recuperação de falhas.

A implementação de sistemas de monitorização e supervisão não é única. Em [4], descreve-se um método baseado na observação de eventos, que permite identificar a ocorrência de falhas através da observação das sequências de eventos que ocorrem no sistema ao longo do tempo. A principal desvantagem desta abordagem é a ausência de informação acerca do estado do sistema, que apresenta claras desvantagens na decisão das acções de recuperação. Outras abordagens, como em [16], utilizam observadores baseados em RdP que modelam o correcto funcionamento do sistema, detectando quando tal não acontece e impondo acções de recuperação do estado normal. À vantagem de se conhecer, em todos os instantes, o estado do sistema, contrapõe-se uma maior dificuldade na detecção de falhas em sistemas complexos com elevada interacção entre componentes. A solução implementada neste trabalho passa pela conjugação das duas abordagens. Desenvolveu-se um monitor baseado nas RdPs de cada um dos componentes e cuja função é apenas observar o estado do sistema. A detecção é feita por um detector de falhas baseado em sequências de eventos. Assim, juntando o melhor dos dois métodos, é possível detectar facilmente a ocorrência de uma falha e acrescentar, a esta, a informação de estado necessária para que sejam decididas as acções de recuperação a impor. Adicionalmente, pretende-se integrar na *CFE* um sistema de desenho assistido por computador. A evolução tecnológica permite integrar, nos *SFF*, postos de desenho e simulação através da tecnologia *CAD/CAM*.

O *CAD* (*Computer Aided Drafting & Design*) é parte integrante do subsistema da engenharia de concepção. Consiste num sistema de projecto assistido por computador e tem, como principais objectivos, definir os req-

uisitos do produto, determinar a estratégia de concepção e formalizar a concepção por meio de documentação diversa. O software *CAM* (*Computer Aided Manufacturing*) utiliza as especificações geradas no CAD, programa as operações de manufactura e permite simular o fabrico do produto, reduzindo, assim, os custos da fase de projecto. Este *software* está, normalmente, equipado com um pós-processador que gera automaticamente o programa CNC a ser usado pela máquina CNC. Sendo o *CAD/CAM* um dos meios de ligação dos departamentos de projecto e produção, não é de estranhar que possua extrema importância no panorama industrial actual e que seria, de facto, uma mais valia integrar um sistema deste tipo na *CFF* do *LSCN*. Um dos objectivos deste trabalho é dotar a *CFF* de uma integração vertical suficiente para que possua as principais fases do processo de produção. Para tal, será adicionado um módulo *CAD/CAM*, constituído pelo software *MasterCam 9*® (*MCam9*) e por uma aplicação (Posto Remoto de *Design* e *Produção*), capaz de comunicar remotamente com a *CFF*, através de uma ligação de rede *Ethernet*.

A arquitectura proposta, onde se acrescenta um posto remoto de design e produção, foi implementada e validada em tempo real na supervisão da célula de fabricação flexível, mostrando-se que este é um passo sólido na utilização da metodologia proposta para a detecção e recuperação de falhas em sistemas de eventos discretos.

## Estrutura do relatório

- Capítulo DOIS: descrevem-se os componentes da *CFF* e suas características, a arquitectura de controlo existente, de [5], e a arquitectura proposta, para implementar o sistema de supervisão e a consola de desenho CAD remota.
- Capítulo TRÊS: é apresentada a teoria sobre Redes de Petri que serviu de base para o desenvolvimento do trabalho. Nesse contexto, são introduzidas as suas definições, propriedades e métodos de análise. É também descrito o método de síntese de controladores para supervisão, apresentado em [11], bem como os pressupostos que o sustentam.
- Capítulo QUATRO: descrevem-se todos os passos da modelação da *CFF* em Redes de Petri. A obtenção do sistema global é feita com a imposição de restrições de controlo entre os diversos componentes. Por fim, apresentam-se os resultados da análise rede, por forma a descortinar as propriedades que esta possui.
- Capítulo CINCO: são classificadas as falhas passíveis de estudo, com exemplos associados. Neste capítulo, modela-se a detecção e recuperação das falhas da *CFF*, com especial incidência naquelas de implementação prática factível.
- Capítulo SEIS: Descreve-se a implementação prática do sistema de *monitorização e supervisão* e do *posto de design e produção*.
- Capítulo SETE: são apresentadas as conclusões a reter acerca do estudo efectuado sobre um sistema de fabrico flexível, métodos utilizados, opções de projecto e propriedades do sistema. Apontam-se, também, algumas direcções no sentido da evolução e desenvolvimento do trabalho efectuado.

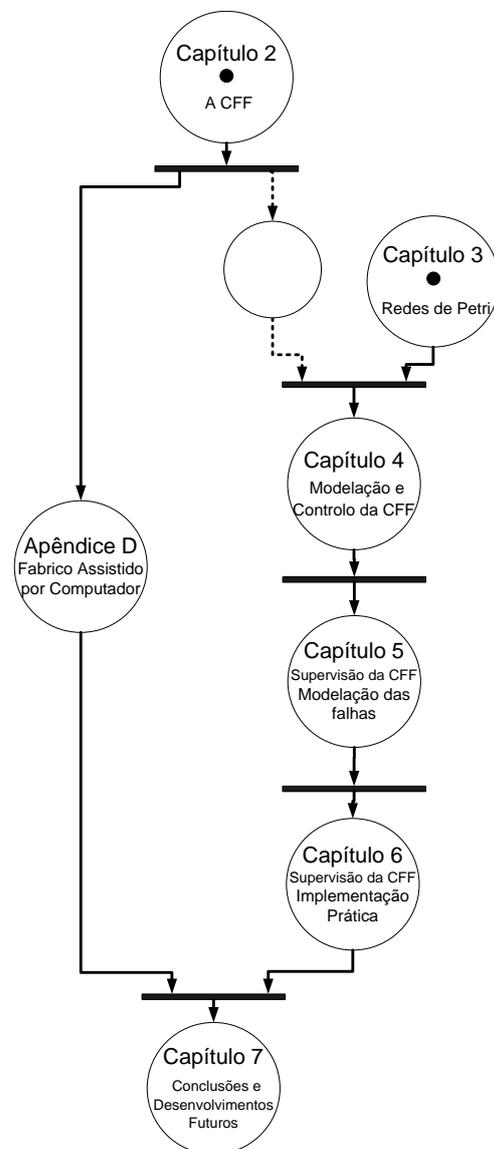


Figura 1.1: Estrutura do Relatório



## CAPÍTULO 2

### A CFF

Neste capítulo, é apresentada a Célula de Fabrico Flexível (*CFF*) do Instituto de Sistemas e Robótica, e por este cedida à secção de Sistemas e Controlo do Departamento de Engenharia Electrotécnica e de Computadores (DEEC).



Figura 2.1: A *CFF*

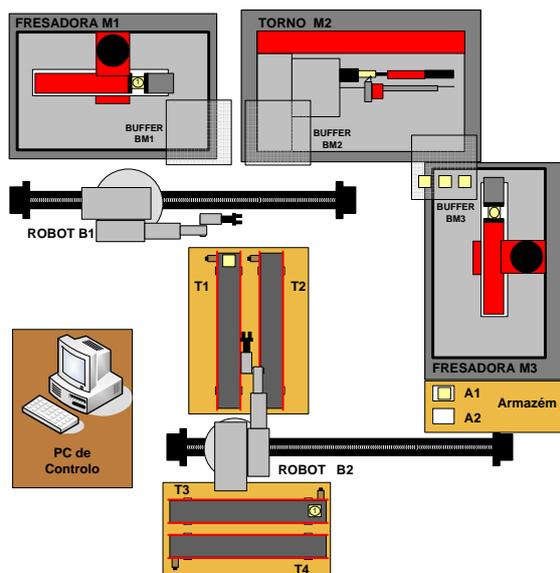


Figura 2.2: Aspecto geral da *CFF*

A *CFF* do referido laboratório, que tem vindo a evoluir ao longo dos anos, tem sido objecto de estudo e desenvolvimento nos vários trabalhos efectuados na última década. No trabalho descrito em [12], realizaram-se os primeiros esforços para a integração dos diversos elementos, ainda sem explorar todas as potencialidades dos mesmos, num sistema totalmente controlado pelos controladores dos braços robots. Em [10], explica-se a evolução desenvolvida no sentido do carregamento automático de programas de maquinação, da incorporação de armazéns temporários de entrada e da maquinação de matéria-prima em mais que uma máquina. Finalmente, no trabalho descrito em [5], desenvolveu-se um gestor para a *CFF* que permitiu viabilizar o controlo dos processos a partir de uma plataforma de alto-nível, de acordo com a filosofia do fabrico flexível. Foi a partir desta última configuração que evoluiu o presente trabalho.

### 2.1 Os elementos da *CFF*

Como se pode observar na figura 2.2, a *CFF* é constituída por:

- Duas fresadoras *EMCO F1-CNC*
- Um torno mecânico *EMCO Compact 5-CNC*
- Dois manipuladores robóticos *Scorbot ER VII ESHED ROBOTEC*, com as respectivas bases deslizantes e controladores
- Quatro tapetes rolantes *ESHED ROBOTEC*
- Um *buffer* local a cada máquina ferramenta CNC
- Dois armazéns
- Computador pessoal equipado com um processador Pentium a 266Mhz, da Intel®, com uma placa Bo- caBoard com 16 portas série RS232. Foi utilizado o sistema operativo *Windows 2000®*.



Figura 2.3: Aspecto dos elementos constituintes da CFF

Depois de apresentados, individualmente, cada um destes componentes, a fim de ilustrar características galvanizadoras ou limitativas a ter em conta, será descrita a nova arquitectura da CFF.

### 2.1.1 Braços Robot

Na figura 2.2 observa-se que a CFF está equipada com dois manipuladores robóticos *Scorbot ER VII* da *ESHED ROBOTEC*, designados neste trabalho por *B1* e *B2*. Estão assentes em bases rolantes, do tipo *sem-fim*, possuem cinco graus de liberdade e um efector terminal tipo pinça, que fixa as matérias a transportar. Todas as trajectórias e tarefas que os robots realizam são pré-programáveis.

Em cada sessão de fabrico, os armazéns *A1* e *A2* armazenam as matérias primas necessárias para produzir o número de peças encomendadas.

Como se observa pela distribuição física da CFF, descrita na figura 2.2, o robot *B2* transporta matérias existentes nos armazéns, *A1* ou *A2*, para os tapetes um ou dois (*T1* ou *T2*) e as máquinas são alimentadas pelo robot *B1*. No fim do processo de maquinação, o robot *B1* coloca as peças novamente nos tapetes *T1* ou *T2* que, depois de transportadas por estes, serão levadas pelo robot *B2* até aos tapetes três ou quatro (*T3* ou *T4*).

Na figura 2.3 observa-se a estrutura do braço robot. As principais especificações técnicas dos manipuladores são apresentadas na tabela E.2 do apêndice.

Cada braço robot tem um controlador associado, que é responsável pelo controlo das juntas do braço, tarefas de geração de trajectória e pela manutenção e execução dos programas que possui armazenados em memória. Cada junta de revolução tem o seu próprio sistema de controlo, ao qual é fornecida a posição através de um encoder óptico. A explicação do processo de controlo, de cada junta, remete-se para a secção E.0.2, em apêndice. A sincronização e o comando do movimento de todas as juntas, durante a execução das instruções do programa, cabe ao módulo de controlo e execução do controlador.

### 2.1.2 Controladores

A função dos controladores vai muito para além do controlo dos robots que lhes estão associados. O facto de estarem dotados de um sistema operativo *multi-tarefa*, que permite a execução de 20 programas em simultâneo, e de um interpretador, que permite a programação do robot em linguagem ACL (Advanced Control Language) [8], leva a que sejam usados como actuadores de todos os outros componentes da célula, servindo de *elo de ligação* entre estes e o PC de controlo. Os controladores são a base do modelo de integração adoptado e é, a partir deles, que se realiza a interacção entre os elementos da célula e o PC. A partir das suas portas série, é possível obter-se indicações de fim de curso, de choque ou de sobre-aquecimento. Estas mensagens irão desempenhar um papel fundamental na detecção e recuperação de falhas.

A inclusão de autómatos programáveis (PLC's) neste sistema é, desde logo, posta de parte. Esta opção justifica-se devido às características apresentadas anteriormente, que incorporam as funções mais comuns dos PLC's.

Os controladores possuem os seguintes interfaces de comunicação com o exterior:

- Teclado especializado - *Teach Pendant*
- Entradas e saídas digitais
- Comunicação série via RS232

O teclado especializado, de entre vários comandos, permite executar programas existentes e gravar posições na memória do controlador para que possam ser utilizadas nos comandos de movimento dos programas.

Cada controlador possui dezasseis entradas e outras tantas saídas digitais, que podem ser acedidas pelos programas em execução no controlador. Deste modo, é possível observar e detectar as acções efectuadas por dispositivos externos e actuar sobre os mesmos, consoante o que se pretenda. As características dos controladores encontram-se presentes na tabela E.4, em apêndice.

Cada controlador possui vinte e uma portas RS232: As primeiras dez comandam as juntas do braço, a base deslizante e os dois tapetes; as dez seguintes servem para reportar estados e receber comandos do PC, actuando nos componentes que lhes estão associados. A última porta reveste-se de primordial importância por duas razões: é a partir dela que se efectua toda a programação dos controladores ou, no caso de não se utilizar o teclado especializado, a execução de comandos; por outro lado, e relativamente ao objectivo final deste trabalho, é a partir desta porta de comunicações que se detectam mensagens de falhas ocorridas nos componentes controlados pelo controlador.

O carácter *multi-tarefa* do controlador, atrás referido, permite o lançamento de processos dedicados que comunicam através das entradas e saídas digitais e das portas séries com cada um dos elementos da célula e com o PC.

Os programas de inicialização e interacção dos componentes da CFF foram desenvolvidos em [5].

### 2.1.3 Tapetes

Os tapetes *T1* e *T2*, da figura 2.2, são encarregues do transporte de peças da entrada para a saída da CFF e vice-versa. Têm ambos exactamente a mesma função e são geridos de acordo com políticas de gestão de recursos a definir pelo utilizador.

Os tapetes *T3* e *T4*, da figura 2.2, estão encarregues do transporte das peças maquinadas até a um, hipotético, posto de controlo de qualidade. Têm ambos o mesmo papel, sendo a forma como são geridos definida pelo utilizador.

Os tapetes *T1* e *T2* estão ligados ao controlador do robot *B1* e os tapetes *T3* e *T4* ao controlador do robot *B2*. Os tapetes são comandados pelo controlador, respectivo, como se de uma junta se tratasse. As suas posições e movimentos são programadas e cada tapete tem um processo dedicado no respectivo controlador.

Os tapetes não têm nenhum tipo de sensor dedicado à detecção de peças, o que representa uma dificuldade acrescida na implementação de um sistema de detecção de falhas. Os controladores enviam mensagens apenas relativamente à sua posição, não ao que transportam. Os fins de trajectória são indicados quando percorrida uma distância predefinida. Neste caso, a matéria prima é transportada de uma ponta para outra do tapete.

Para se obter informação acerca dos materiais transportados, pelos tapetes, será necessário equipá-los com sensores de presença para detectar funcionamentos anómalos.

### 2.1.4 As Máquinas CNC da CFF

Como se observa na figura 2.2, a CFF é constituída por duas fresadoras (*M1* e *M3*), um torno mecânico (*M2*, figura 2.3) e três *buffers* acoplados a cada máquina. Estas máquinas têm como função maquinar as peças em bruto, isto é, remover matéria por desbaste ou polimento, alterando desta forma as dimensões e propriedades das superfícies até à obtenção do resultado final. Cada *buffer* tem como função disponibilizar posições livres às peças que aguardam operações de maquinação.

Tratam-se de máquinas de Controlo Numérico por Computador (CNC), também conhecidas por máquinas ferramenta, compostas por um controlador e pela máquina propriamente dita. As características técnicas das máquinas são apresentadas no apêndice F. O controlador é constituído por uma unidade de processamento de dados (DPU - data processing unit) e uma unidade de controlo de servomecanismos (CLU - control loops unit). A DPU desempenha, pelo menos, as seguintes funções:

- dispositivo de leitura
- circuitos de leitura e lógica de teste de paridade
- distribuição os dados entre os vários eixos
- fornecimento comandos de velocidade entre pontos sucessivos do desenho

A CLU executa as seguintes funções:

- controlo de posição em todos os eixos
- controlo de velocidade em cadeia fechada
- controlo de desaceleração e folga

Este controlo é efectuado com motores passo-a-passo, não existindo qualquer tipo de *encoders* ou de sensores de fim de curso, o que limita o funcionamento das máquinas em termos de posições absolutas. Se, por exemplo, ocorrer um falha de energia, a posição inicial da base e da fresa terá que ser atingida com o auxílio do operador humano, o que limitará a introdução de mecanismos de recuperação de falhas automáticos.

As máquinas são controladas através de um interface digital ( $x22$ ) que permite comandar, através dos controladores dos robots, as principais acções e observar o estado das mesmas. Na tabela F.2 do apêndice F descreve-se, em pormenor, este interface. De realçar que, nem o estado de emergência nem o comando de paragem da máquina são tratados na configuração anterior da CFF.

Para além do interface digital, as máquinas CNC da CFF estão equipadas com portas série RS232, em que a informação é transmitida em *half-duplex*. Estas portas são activadas através do interface digital e permitem enviar para as máquinas os programas CNC e recebê-los das mesmas.

Embora tecnologicamente ultrapassadas, as máquinas da CFF possuem um importante carácter pedagógico, visto que, mesmo as máquinas CNC mais actuais, se baseiam nos mesmos conceitos e princípios de funcionamento.

### 2.1.5 O PC - Aplicação de Controlo

Toda a decisão e controlo da CFF centra-se no PC da CFF. Trata-se de um *Pentium II 266Mhz* da *Intel*, equipado com uma placa *Bocboard* de dezasseis portas série RS232. Neste PC corre a aplicação que controla e gere toda a CFF, desenvolvida em *Visual C++* e baseada no conceito de programação por objectos.

A aplicação informática, desenvolvida e descrita em [5], é constituída por duas partes distintas, mas complementares: um sequenciador e um gestor multitarefa.

O primeiro, está encarregue do sequenciamento do fabrico de cada peça, políticas de gestão de recursos e estabelecimento de prioridades.

O segundo, é uma aplicação *multi-thread* que permite controlar, de forma concorrente, todos os recursos da célula. Possui tarefas (*threads*) dedicadas a cada um dos componentes da CFF. Estas tarefas comunicam directamente com os processos dos respectivos controladores, através das portas série de *Bocboard*, controlando, assim, cada um destes.

## 2.2 Arquitectura Existente

Na arquitectura prévia, a gestão do processo de fabrico é totalmente controlada pelo PC. É nele que se centra toda a decisão e controlo da CFF. Os controladores apenas funcionam como actuadores nos diversos componentes da célula. Cada recurso da célula possui um processo dedicado no respectivo controlador, que funcionará como uma pequena máquina de estados, limitando-se a receber e a distribuir informação entre os componentes da célula e as respectivas tarefas dedicadas da aplicação de controlo.

De notar que, na arquitectura existente, não está previsto qualquer tipo de funcionamento anormal, tanto ao nível das comunicações como da ocorrência de falhas nos componentes da CFF. A elevada interacção entre os componentes, controlada pelo gestor multi-tarefa, e a ausência de tratamento de falhas levam ao bloqueio parcial ou completo do processo de fabrico quando, por exemplo, uma peça é mal colocada no armazém.

Outra das limitações desta arquitectura, é o facto de estar completamente isolada do exterior, não prevendo qualquer comunicação com um sistema CAD/CAM nem pedidos remotos de produção. Esta limitação vai, de certa forma, contra a filosofia da fabricação flexível actual.

## 2.3 Arquitectura Proposta

A arquitectura, ora implementada, não retira nada à anteriormente desenvolvida. Foram mantidas todas as características, tanto ao nível de software e protocolos de comunicação, como ao nível do *hardware*. O trabalho desenvolvido, baseou-se na adição de dois novos módulos ao sistema: Um módulo de supervisão e monitorização e um módulo de CAD/CAM (Desenho e fabrico assistidos por computador) e controlo remoto. Estes dois módulos serão descritos individualmente nos próximos capítulos.

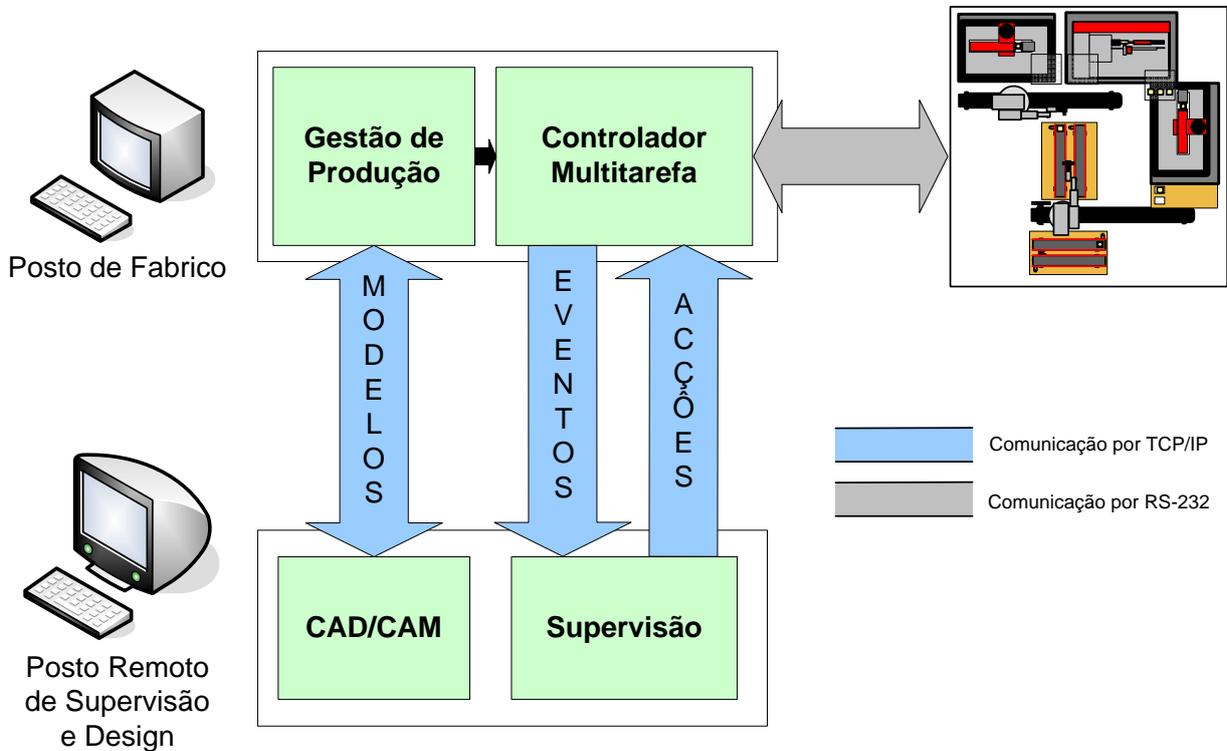


Figura 2.4: Nova arquitectura da CFF

Numa mesma aplicação remota, cuja comunicação com a CFF é feita por *TCP/IP*, foram reunidas várias funcionalidades: Gestão dos modelos em fabrico, com a possibilidade de envio de novos modelos desenvolvidos na ferramenta *CAD/CAM*; Criação e/ou ordenação de sessões de fabrico e, com maior importância no presente trabalho, a Supervisão e Monitorização do processo de fabrico com base na Teoria dos Sistemas de Eventos Discretos. Foram adicionados à aplicação local da CFF dois processos que controlam os próprios controladores dos braços, o que, além da inicialização automática da célula (até aqui apenas possível com recurso às teclados especializados de controlo), permite, também, a observação de eventos até aqui ignorados e que são preponderantes para a detecção de falhas. De realçar, ainda, o processo *supervisor local* que, recebendo ordens remotas, pode interagir com todos os elementos da CFF, o que será fundamental para a implementação de mecanismos de recuperação de falhas.

As etapas de estudo e desenvolvimento desta nova arquitectura são descritas nos capítulos seguintes.



# CAPÍTULO 3

## REDES DE PETRI

Um Sistema de eventos discretos (*SED*) é um sistema cujo espaço de estados é discreto e cujo estado é só alterado após a ocorrência assíncrona de um evento num certo instante de tempo. É o paradigma típico para representar um sistema de manufactura como a *CFE*.

Os *SEDs* podem ser modelados fazendo uso de diversas metodologias como sejam os automatos, as linguagens e as Redes de Petri, entre outras.. As Redes de Petri (RdP), introduzidas por Carl Adam Petri em 1962, são uma delas e permitem modelar, analisar, simular e observar um *SED*. A teoria até aqui desenvolvida permite que um *SED* seja modelado por uma RdP, que não é mais do que uma representação matemática do mesmo. A análise formal da rede pode revelar informações importantes acerca do comportamento estrutural e dinâmico do sistema e apontar mudanças que melhorem os mesmos.

Importa, por estas razões, apresentar as definições e conceitos básicos das RdPs usados para modelação do sistema de manufactura em causa. A apresentação destas definições não pretende ser uma demonstração exaustiva da formulação matemática das RdP, mas apenas introduzir o assunto e mostrar qual a formulação usada neste trabalho. A teoria envolvida está descrita com mais pormenor em [13] e [11].

Depois da formulação teórica, serão também apresentadas as ferramentas de análise mais importantes, com especial incidência para as usadas no estudo do sistema de manufactura em causa. Por fim, será descrita a teoria apresentada em [11] e [9], acerca da síntese de controladores para supervisão baseada nos invariantes de marcação de uma RdP.

### 3.1 Conceitos Básicos

#### 3.1.1 Notação e Definição

A definição de uma RdP divide-se em dois passos. No primeiro, é definida a estrutura da RdP, um grafo constituído por lugares e transições que pretende modelar o funcionamento do sistema. De seguida, junta-se a esta estrutura um estado inicial, um conjunto de lugares marcados e uma função de transição, estando, assim, completo o modelo da RdP. As definições destes e de outros conceitos serão descritos nesta secção.

##### Definição 3.1 (Estrutura da RdP)

A estrutura de uma RdP,  $C$ , é um 4-tuplo

$$C = (P, T, D^+, D^-)$$

onde

- $P$  é o conjunto finito de  $n$  lugares.
- $T$  é o conjunto finito de  $m$  transições.
- $D^+ e D^-$  são matrizes inteiras de elementos não negativos, que representam o fluxo entre os dois conjuntos  $P$  e  $T$ .

Os elementos das matrizes  $D^+ e D^-$  indicam os pesos dos arcos que ligam as transições a lugares e lugares a transições, respectivamente. ▲

Em termos gráficos, tal como se verifica na figura 3.1, convencionou-se representar os lugares por círculos e as transições por segmentos de recta, sendo estes ligados por arcos. Os arcos que ligam transições a lugares são descritos pela matriz  $D^+ \in \mathbb{Z}^{(n \times m)}$  e os arcos que ligam lugares a transições descritos por  $D^- \in \mathbb{Z}^{(n \times m)}$ .

Um lugar é um *acumulador* de marcas e não pode conter um número negativo de marcas. Quando uma transição é *disparada*, são retiradas marcas de uns lugares e colocadas noutros. Um arco com peso  $D_{ij}^+$  da transição  $j$  para o lugar  $i$  indica que, quando a transição  $j$  dispara, o lugar  $i$  recebe  $D_{ij}^+$  marcas. Um arco com



Figura 3.1: Elementos de uma RdP

peso  $D_{kj}^-$  do lugar  $j$  para a transição  $j$  indica que um lugar  $k$  tem que conter pelo menos  $D_{kj}^-$  marcas para que a transição  $j$  esteja permitida e, quando disparada, o lugar  $k$  perderá  $D_{kj}^-$  marcas. Para que uma transição seja permitida e possa disparar, todos os seus lugares devem ter um número mínimo de marcas.

A distribuição de marcas pelos lugares da rede num dado instante é chamada de *Estado* ou *Marcação da RdP*. A marcação é representada por um vector inteiro  $n$ -dimensional  $\mu$ . A *marcação inicial* da rede é  $\mu_0$ . As transições a disparar são representadas pelo vector  $m$ -dimensional  $q$ . O  $j$ -ésimo elemento de  $q$  é 0 se a transição  $j$  não for disparável e o  $j$ -ésimo elemento de  $q$  é 1 se a transição for disparável. A validade de um dado *vector de disparo*  $q$  pode ser determinada pela seguinte condição:

$$\mu \geq D^- q \quad (3.1)$$

A *Matriz de Incidência de uma RdP* é definida por:

$$D = D^+ - D^- \quad (3.2)$$

Quando uma RdP não contém transições com arcos de entrada e de saída para um mesmo lugar (*self-loops*), a condição de disparo pode ser expressa em termos da matriz de incidência. Assim, como todos elementos não nulos de  $D^+$  e  $D^-$  são mutuamente exclusivos e todos os vectores de  $D^+ q$  são maiores ou iguais a zero, a seguinte condição pode ser usada:

$$\begin{aligned} \mu + D^+ q &\geq D^- q \\ \mu + (D^+ - D^-) q &\geq 0 \\ \mu + D q &\geq 0 \end{aligned} \quad (3.3)$$

Na ausência de *self-loops*, a RdP pode ser unicamente definida por  $D$  e poderá ser usada apenas a condição 3.3. No método de síntese de controladores descrito em [11] assume-se que a RdP não contém *self-loops* pelo que, por conveniência, será evitado neste trabalho o uso dos mesmos. Este assunto será tratado com mais detalhe na secção 3.2.

### 3.1.2 Dinâmica das RdP

Quando as transições descritas por  $q$  disparam, o estado da RdP muda. A mudança de estado é descrita por:

$$\mu \Leftarrow \mu + Dq \quad (3.4)$$

Quando uma RdP não contém *self-loops*, pode ser descrita pelo seguinte sistema:

$$\begin{aligned} \mu(0) &= \mu_0 \\ \mu(k+1) &= \mu(k) + Dq(k) \quad D \in \mathbb{Z}^{n \times m}, \mu \in \mathbb{Z}^n, q \in \mathbb{Z}^m, (\mu, q \geq 0) \end{aligned} \quad (3.5)$$

Como facilmente se observa, este sistema é em tudo igual a um sistema linear em tempo discreto, com limitações no estado e nos vectores de entrada.

### 3.1.3 Propriedades Comportamentais

De seguida, são apresentadas as principais propriedades de uma Rede de Petri, definidas em [13]. Apresentam-se três propriedades comportamentais das RdP: a capacidade/limitação; a vivacidade e a alcançabilidade. Existem outras propriedades comportamentais. No entanto, apresentam-se somente as que permitem a caracterização correcta da classe de redes abordadas neste trabalho.

### Alcançabilidade

A alcançabilidade é base fundamental para o estudo de qualquer sistema dinâmico.

Uma marcação  $M_n$  é dita *alcançável* ou *atingível*, a partir de uma marcação inicial  $M_0$ , se existe uma sequência de disparos que transforma  $M_0$  em  $M_n$ . Uma sequência de disparos é definida por  $\sigma = t_1 t_2 \dots t_n$ .  $M_n$  é atingível a partir de  $M_0$  através de  $\sigma$ ,  $M_0[\sigma > M_n$ . O conjunto de todas as marcações atingíveis a partir de  $M_0$  é representado por  $R(M_0)$ . O conjunto de todas as sequências de disparos possíveis a partir de  $M_0$  é representado por  $L(M_0)$ .

Com estas definições, a *alcançabilidade de uma RdP* apresenta-se como o problema de saber se  $M_n \in R(M_0)$  para uma dada marcação  $M_n$  na RdP. Em algumas aplicações pode haver interesse apenas na marcação de um subconjunto de lugares. Isto leva ao problema da *alcançabilidade de uma submarcação*, isto é, saber se  $M'_n \in R(M_0)$ , onde  $M'_n$  é uma qualquer marcação cuja restrição a um dado subconjunto de lugares é coincidente com a de uma dada marcação  $M_n$ .

### Capacidade/Limitação

Um lugar  $P$  de uma RdP diz-se  $k$ -limitado ou, simplesmente, limitado (*bounded*), se o número de marcas em  $P$  não exceder um número inteiro positivo  $k$ , para todas as marcações  $M$  obtidas a partir de  $M_0$ .

#### Definição 3.2 (RdP $k$ -limitada)

Uma RdP diz-se  $k$ -limitada se  $M(p) \leq k$  para todos os lugares  $P$  e todas as marcações  $M \in R(M_0)$  da RdP.▲

Um lugar  $P$  diz-se **seguro** se é 1-limitado para todas as marcações da rede. Uma RdP diz-se segura se todos os seus lugares são, para a marcação inicial  $M_0$ , seguros.

### Vivacidade

O conceito de vivacidade de uma RdP possui uma relação estreita com a ausência de bloqueios (*deadlocks*). Em [13] definem-se vários níveis de vivacidade. Uma transição  $t$  numa RdP é dita:

- 1) **nível 0 (Morta)**, se  $t$  nunca pode ser disparada para nenhuma sequência de disparos em  $L(M_0)$ .
- 2) **Nível 1 (potencialmente disparável)**, se  $t$  pode ser disparada pelo menos uma vez numa sequência de disparos em  $L(M_0)$ .
- 3) **Nível 2**, se, dado um inteiro positivo  $k$ ,  $t$  pode ser disparada pelo menos  $k$  vezes numa sequência de disparos em  $L(M_0)$ .
- 4) **Nível 3**, se  $t$  pode ser disparada infinitamente, numa sequência em  $L(M_0)$ .
- 5) **Nível 4 (viva)**, se a transição  $t$  é de nível 1 para qualquer marcação em  $R(M_0)$ .

Se uma rede for de **Nível 4** é dita *viva*. Isto significa que a vivacidade de uma rede garante a não existência de bloqueios.

Se todas as transições da rede são vivas de nível  $k$ , com  $k = 0, 1, 2, 3, 4$ , a RdP é dita *viva de nível  $k$* .

#### 3.1.4 Propriedades Estruturais

As propriedades estruturais são aquelas que dependem das características topológicas das RdPs e são independentes da marcação inicial  $\mu_0$ . Estas propriedades podem ser, muitas vezes, caracterizadas pela matriz de incidência  $D$  da RdP e pelas equações ou desigualdades que lhe estão associadas. Estas propriedades são cruciais na análise das RdPs, já que tornam possível a investigação da estrutura de uma RdP independentemente do comportamento dinâmico da mesma.

### Invariantes de marcação ou p-invariantes

Os *invariantes de marcação* (*place invariants*) são conjuntos de lugares, cuja soma das marcas neles contidas é constante para todas as marcações possíveis. São representados por vectores inteiros de dimensão  $n$ , onde  $n$  é o número de lugares da rede. Os elementos não nulos correspondem aos lugares que pertencem a um determinado invariante.

#### Definição 3.3 (Invariante de marcação)

Um invariante de marcação é definido por todo o vector inteiro que satisfaz a seguinte condição:

$$x^T \mu = x^T \mu_0 \quad (3.6)$$

em que  $\mu_0$  representa a marcação inicial da rede e  $\mu$  uma qualquer marcação subsequente. ▲

Os invariantes de marcação podem ser obtidos encontrando as soluções inteiras de:

$$x^T D = 0 \quad (3.7)$$

Dado um qualquer vector de disparos  $q$ , tem-se:

$$\begin{aligned} x^T \mu(k+1) &= x^T (\mu_k + Dq(k)) \\ &= x^T \mu_k \end{aligned}$$

### Invariantes de disparo ou t-invariantes

Em dualidade com os invariantes de marcação estão os *invariantes de disparo* (*transition invariants* em [11]).

#### Definição 3.4 (Invariantes de Disparo)

Um vector  $y$  inteiro de dimensão  $m$  é um invariante de transição se:

$$Dy = 0 \quad (3.8)$$

▲

Usando a definição 3.5 observa-se que:

$$\begin{aligned} \mu(0) &= \mu_0 \\ \mu(1) &= \mu_0 + Dq(0) \\ \mu(N) &= \mu_0 + D(q(0) + \dots + q(N-1)) \\ &= \mu_0 + Dy \end{aligned}$$

Os *t-invariantes* representam sequências de disparos que reinicializam uma marcação. Eles realçam a componente cíclica de um processo.

## 3.2 Modelação e Controlo de Sistemas de Eventos Discretos

A modelação de sistemas de eventos discretos, através de Redes de Petri (RdP), baseiam-se normalmente em métodos mais ou menos intuitivos que têm como principal objectivo modelar e simular o comportamento de um sistema real. Em [14] e [1] são apresentadas várias formas de modelar sistemas em que vários componentes concorrem entre si, partilham os mesmos recursos e/ou apresentam sincronismo de várias acções. No entanto, nesta forma de abordar a modelação, o desenho da rede e dos controladores que impõem a interacção dos vários componentes do sistema são feitos de forma intuitiva e, muitas vezes, baseada em pequenas extensões de rede bem conhecidas para resolver os mais diversos problemas, mas que não garantem as boas propriedades para a Rede total resultante.

Em [11] é apresentada uma nova forma de modelar e implementar o controlo de um sistema de eventos discretos, modelados em RdPs. Este método, baseado no conceito de invariantes de marcação de uma RdP,

formaliza matematicamente o que, muitas vezes, se obtém através de modelações empíricas. Esta técnica permite sintetizar um controlador (ou lugar de controlo) para uma RdP, baseado nas suas propriedades estruturais e num conjunto de restrições lineares impostas ao seu comportamento. O resultado, é um modelo unificado do sistema e do seu controlador, que facilita a análise, a simulação, a verificação e a sua implementação.

A síntese de controladores, no método proposto, é bastante eficiente, uma vez que o número de lugares de controlo acrescentados à rede é igual ao número de restrições impostas e os cálculos para a sua obtenção envolvem pouco mais que álgebra de matrizes.

Os controladores obtidos por este método, garantem máxima permissividade (maximally permissivity), isto é, inibem somente o disparo de transições que violem uma ou mais restrições.

De seguida, apresenta-se um resumo do método de síntese de controladores apresentada em [11]. Um exemplo, extraído da CFF, constituído por um braço robot e um tapete acompanhará essa apresentação. Neste exemplo, pretende-se efectuar o transporte de uma peça de um armazém para um tapete e deste para a saída. Depois de obtidas as RdP que modelam o comportamento básico destes dois componentes, nesta tarefa específica, serão definidas as restrições estruturais de cada um deles e as restrições que deverão ser impostas para que haja sincronismo à correcta execução da tarefa. Com base nestas restrições será então sintetizado o controlador para a RdP.

O sistema a controlar é modelado por uma RdP com  $n$  lugares e  $m$  transições e conhecido em [11] como *processo*. A matriz de incidência do processo é  $D_p \in \mathbb{Z}^{n \times m}$ . A necessidade de se controlar o processo surge da possibilidade de este violar certas restrições no seu comportamento. A *rede de controlo* é, também ela, uma RdP, com a matriz de incidência  $D_c$ , ligada às transições do processo e com um conjunto novo de lugares de controlo. A *rede controlada*, com matriz de incidência  $D$ , não é mais do que a RdP original com o controlador adicionado. A rede controlada é também conhecida por sistema controlado ou sistema em malha fechada.

O propósito do controlador é forçar o processo a cumprir restrições no seu comportamento. Neste sentido, considere-se que o objectivo de controlo é o de restringir todas as marcações alcançáveis da rede  $\mu_p$  de modo que

$$l^T \mu_p \leq b \quad (3.9)$$

onde  $l \in \mathbb{Z}^n$  é um vector de inteiros e  $b$  é um escalar inteiro.

Introduzindo uma variável auxiliar não negativa  $\mu_c$  pode-se transformar a inequação 3.9 numa equação da seguinte forma

$$l^T \mu_p + \mu_c = b \quad (3.10)$$

Esta variável auxiliar representa um novo lugar  $c$  que, ao guardar temporariamente as marcas, garante o cumprimento da restrição. Este lugar é conhecido por *lugar de controlo*.

Quando existe mais do que uma restrição para o processo a equação 3.9, pode ser reescrita na seguinte forma matricial

$$L \mu_p \leq b \quad (3.11)$$

onde  $\mu_p$  é o vector de marcação do processo,  $L \in \mathbb{Z}^{n_c \times n}$ ,  $b \in \mathbb{Z}^{n_c}$  e  $n_c$  é o número de restrições do tipo 3.9.

Adicionando a variável auxiliar obtém-se

$$L \mu_p + \mu_c = b \quad (3.12)$$

onde  $\mu_c \in \mathbb{Z}^{n_c}$  é um vector de inteiros que representa a marcação dos lugares de controlo.

A matriz de incidência  $D \in \mathbb{Z}^{(n+n_c) \times m}$  do sistema em malha fechada é dada por

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} \quad (3.13)$$

e o vector de marcação  $\mu \in \mathbb{Z}^{n+n_c}$  e a marcação inicial são

$$\mu = \begin{bmatrix} \mu_p \\ \mu_c \end{bmatrix} \quad e \quad \mu_0 = \begin{bmatrix} \mu_{p0} \\ \mu_{c0} \end{bmatrix} \quad (3.14)$$

Os invariantes da equação 3.12 devem satisfazer a equação 3.7, uma vez que 3.12 está na forma da equação 3.6.

$$\begin{aligned} X^T &= [LI] \begin{bmatrix} D_p \\ D_c \end{bmatrix} = 0 \\ LD_p + D_c &= 0 \end{aligned} \quad (3.15)$$

onde  $I \in \mathbb{Z}^{n_c \times n_c}$  é a matriz de identidade.

De 3.15 e 3.12 pode-se então realizar a síntese do controlador.

### Teorema 3.1 (Síntese de Controladores [11])

Seja

$$b - L\mu_{p_0} \geq 0 \quad (3.16)$$

então a RdP do controlador,  $D_c \in \mathbb{Z}^{n_c \times m}$ , com a marcação inicial  $\mu_{c_0}$  é obtido através de

$$D_c = -LD_p, \quad (3.17)$$

$$\mu_{c_0} = b - L\mu_{p_0} \quad (3.18)$$

e impõe as restrições pretendidas. Assume-se que todas as transições de entrada na rede de controlo são controláveis.

Se a inequação 3.16 for falsa então as restrições não podem ser impostas, uma vez que as condições iniciais do processo se encontram fora da gama definida pelas restrições. ♦

### Exemplo 3.1 (Síntese de controladores para imposição de restrições estruturais)

As RdPs da figura 3.2 representam as redes original do braço e do tapete e é sobre esta que se pretende efectuar o controlo.

Com este exemplo, pretende-se ilustrar as vantagens do uso da técnica proposta no controlo de RdPs. A sua matriz de incidência e marcação inicial são:

$$D_p = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \end{bmatrix} \quad \mu_{p_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Neste exemplo, o objectivo é forçar a que o robot só possa transportar uma peça do armazém para o tapete rolante, ou seja, há a necessidade de impor uma restrição de ordem estrutural para que só exista uma peça no efector terminal do robot. De forma análoga, o tapete só poderá, também ele, transportar uma peça de cada vez. Por observação da rede, torna-se claro que as restrições a impor são:

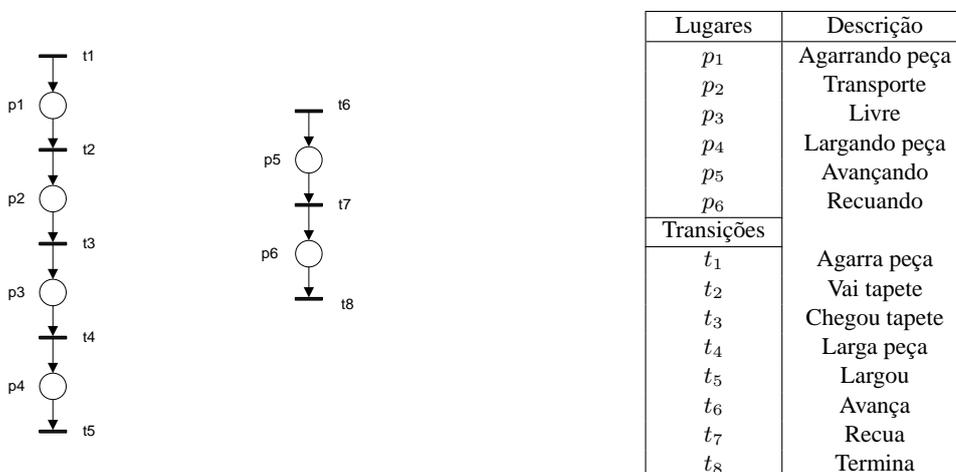


Figura 3.2: RdPs básicas dos componentes

$$\begin{aligned}\mu_5 + \mu_6 &\leq 1 \\ \mu_1 + \mu_2 + \mu_3 + \mu_4 &\leq 1\end{aligned}$$

ou seja,  $L$  e  $b$  são dados por:

$$L = \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

Há que verificar a condição 3.16, para garantir que o controlador que se obtém impõe as restrições pretendidas. Observando a marcação inicial das redes 3.16 e o vector  $b$ , conclui-se facilmente a veracidade da condição. Posto isto, usando a equação 3.17 pode-se obter o controlador a seguinte matriz de incidência do controlador:

$$D_c = -LD_p = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

A marcação inicial para o lugar de controlo  $\mu_{c_0}$  pode ser obtida a partir da equação 3.18

$$\mu_{c_0} = b - L\mu_{p_0} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

De 3.13 conclui-se que a rede controlada terá a seguinte matriz de incidência:

$$D = \begin{bmatrix} D_p \\ D_c \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Os lugares  $p_7$  e  $p_8$  representam os lugares de controlo sintetizados e obrigam a que o braço e o tapete apenas transportem uma peça de cada vez. Obteve-se uma RdP controlada, representada na figura 3.3. ■

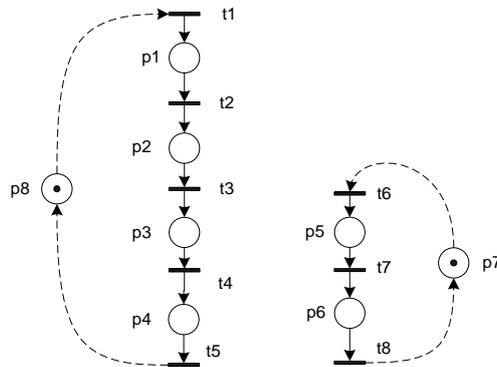


Figura 3.3: RdP com o controlador

### 3.2.1 Transições não controláveis e não observáveis

É comum, num SED, existirem eventos sobre os quais não se pode exercer qualquer influência ou que ocorrem sem que se consigam observar. À luz da definição clássica de controlabilidade e de observabilidade conclui-se que o mesmo é dizer que, numa RdP, certas transições podem não ser controláveis ou não observáveis.

Para que um controlador baseado em invariantes de marcação seja realizável, é necessário que, num processo que contenha transições não controláveis e não observáveis, a restrição que se quer impor seja *admissível*.

**Definição 3.5 (Transição não controlável)**

Uma transição de um processo é dita não controlável se o seu disparo não pode ser inibido por uma acção externa. ▲

Seja  $D_{uc} \in \mathbb{Z}^{n \times n_{uc}}$  a matriz de incidência composta pelas colunas de  $D_p$  correspondentes às transições não controláveis, sendo  $n_{uc}$  o número de transições não controláveis da RdP. A porção do controlador que corresponde às transições não controláveis é dada por  $-LD_{uc}$ . Por isso, para que uma restrição seja válida, no que toca à não controlabilidade, todos os elementos de  $LD_{uc}$  devem ser menores ou iguais a zero.

Prova-se também que, dado o processo com transições não controláveis descritas pela matriz de incidência  $D_{uc}$  e dada restrição  $l^T \mu_p \leq b$ , se

$$l^T D_{uc} \leq 0 \quad (3.19)$$

a restrição é admissível para esse processo.

**Definição 3.6 (Transição não observável)**

Uma transição de um processo é dita não observável se os seus disparos não puderem ser directamente detetados ou avaliados. ▲

Seja  $D_{uo} \in \mathbb{Z}^{n \times n_{uo}}$  a matriz de incidência composta pelas colunas de  $D_p$  correspondentes às transições não observáveis, sendo  $n_{uo}$  o número de transições não observáveis da RdP. A porção do controlador que corresponde às transições não controláveis é dada por  $-LD_{uo}$ . Por isso, para que uma restrição seja válida, no que toca à não observabilidade, todos os elementos de  $LD_{uo}$  devem ser iguais a zero (assumindo que uma transição não observável é também não controlável).

De forma análoga para a não observabilidade: para um dado processo com transições não observáveis e uma dada restrição  $l^T \mu_p \leq b$ , se

$$l^T D_{uo} = 0 \quad (3.20)$$

então a restrição é admissível.

**3.2.2 Síntese de controladores para RdPs com transições não controláveis e não observáveis**

Como se viu, a presença de transições não controláveis e não observáveis introduz dificuldades acrescidas na síntese de controladores. Quando restrições a impor violam as condições 3.19 e/ou 3.20, surge a necessidade de as adaptar de modo a garantir que seja possível impô-las sem que tal aconteça.

Em [11], mostra-se que é possível encontrar restrições admissíveis da forma  $L' \mu_p \leq b'$  que impõem as restrições não admissíveis  $L \mu_p \leq b$  e mantêm a máxima permissividade do controlador.

**Lema 3.1 (Estrutura para a Transformação da Restrição)**

$$\text{Seja } R_1 \in \mathbb{Z}^{n_c \times n} \text{ satisfazendo } R_1 \mu_p \geq 0 \quad \forall \mu_p \quad (3.21)$$

$$\text{Seja } R_2 \in \mathbb{Z}^{n_c \times n_c} \text{ uma matriz diagonal positiva} \quad (3.22)$$

Se  $L' \mu_p \leq b'$  onde

$$L' = R_1 + R_2 L \quad (3.23)$$

$$b' = R_2 (b + \mathbf{1}) - \mathbf{1} \quad (3.24)$$

e  $\mathbf{1}$  é um vector de dimensão  $n_c$  em que todos os elementos são iguais a 1, então  $L \mu_p \leq b$ . ○

Para que a restrição  $L' \mu_p \leq b'$  possa ser imposta à RdP, é preciso que se verifiquem as condições do lema seguinte.

**Lema 3.2 (Condição inicial para as restrições transformadas)**

A restrição  $L'\mu_p \leq b'$ , onde  $L' \neq 0$  e  $b'$  são definidas por 3.23 e 3.24, podem ser impostas numa RdP com marcação inicial  $\mu_{p_0}$  sse:

$$a \leq R_1\mu_{p_0} \leq R_2(b + \mathbf{I} - L\mu_{p_0}) - \mathbf{I} \quad (3.25)$$

◦

Com os dois lemas anteriores, sabe-se agora como transformar uma restrição não admissível noutra admissível que a imponha. O teorema seguinte apresenta uma técnica para sintetizar um controlador que impõe a restrição transformada  $L'\mu_p \leq b'$ .

**Teorema 3.2 (Síntese de controladores para restrições transformadas[9])**

Seja um processo modelado por uma RdP com uma matriz de incidência  $D_p$  com um conjunto de transições não controláveis descritas por  $D_{uc}$  e com um conjunto de transições não observáveis descritas por  $D_{uo}$ . Pretende-se impor à marcação da RdP um conjunto de restrições lineares  $L\mu_p \leq b$ . Assumindo que  $R_1$  e  $R_2$  verificam 3.21 e 3.22 com  $R_1 + R_2L \neq 0$  e sendo

$$\begin{bmatrix} R_1 & R_2 \end{bmatrix} \begin{bmatrix} D_{uc} & D_{uo} & -D_{uo} & \mu_{p_0} \\ LD_{uc} & LD_{uo} & -LD_{uo} & L\mu_{p_0} - b - \mathbf{I} \end{bmatrix} \leq \begin{bmatrix} 0 & 0 & 0 & -\mathbf{I} \end{bmatrix} \quad (3.26)$$

Então o controlador

$$D_c = -(R_1 + R_2L)D_p = -L'D_p \quad (3.27)$$

$$\mu_{c_0} = R_2(b + 1) - 1 - (R_1 + R_2L)\mu_{p_0} = b' - L'\mu_{p_0} \quad (3.28)$$

existe e força a que todas as marcações subsequentes da RdP controlada satisfaçam a restrição  $L\mu_{p_0} \leq b$  sem tentar inibir transições não controláveis ou detectar transições não observáveis. ♦

**Cálculo de  $R_1$  e  $R_2$** 

Uma das principais dificuldades na síntese de controladores na presença de transições não controláveis e não observáveis é a determinação das matrizes  $R_1$  e  $R_2$  que transformam a restrição linear  $L\mu_{p_0} \leq b$  numa restrição admissível. Em [11] são apresentados dois métodos para a determinação de  $R_1$  e  $R_2$ .

O primeiro baseia-se na **programação linear inteira**. Mostra-se em [11, pp.52] que é possível converter as condições do teorema 3.2 para a forma *standard* de um problema de programação linear inteira (ILP). Sendo  $R = [R_1 \ R_2' \ R_3]$ , em que  $R_2' = R_2 - 1$  e  $R_3$  é um vector de variáveis auxiliares usadas para converter a condição de admissibilidade 3.19 numa igualdade, pode definir-se o seguinte ILP:

$$\min_R \left( z(R) = R \begin{bmatrix} \mu_{p_0} \\ L\mu_{p_0} - b - 1 \\ 0 \end{bmatrix} \right) \quad (3.29)$$

$$s.t. \begin{cases} R \begin{bmatrix} D_{uc} & D_{uo} \\ LD_{uc} & LD_{uo} \\ I & 0 \end{bmatrix} = -L \begin{bmatrix} D_{uc} & D_{uo} \end{bmatrix} \\ R \geq 0 \text{ (inteiro)} \end{cases}$$

Depois de resolver 3.29, se o mínimo da função objectivo  $z^* = z(R^*)$  for maior do que  $b - L\mu_{p_0}$  então o problema não pode ser resolvido, pois não há valores de  $R_1$  e  $R_2$  que satisfaçam o lema 3.2. De notar que  $L$  corresponde a apenas uma restrição, isto é  $n_c = 1$ . Múltiplas restrições podem ser resolvidas individual e independentemente.

O segundo método determina as transformações mais apropriadas, através de algumas **operações nas linhas de uma matriz** que contém as colunas não observáveis e não controláveis da matriz de incidência. Este método é bastante menos exigente a nível computacional, já que envolve pouco mais do que triangularização de matrizes. Por esta razão, o primeiro método só será usado caso o segundo falhe.

### 3.2.3 Restrições Lineares Generalizadas

As restrições impostas no exemplo 3.1 são, como se demonstrou, directamente apresentadas sob a forma da restrição linear 3.11. Estas restrições são conhecidas como *restrições lineares de marcação*. No entanto, existem muitas outras restrições que podem ser impostas, mesmo não estando directamente na forma da equação 3.11. Em [9] mostra-se que outro tipo de restrições lineares, *restrições lineares generalizadas (RLG)*, podem ser efectivamente impostas sob a forma de restrições lineares de marcação.

$$L\mu + Hq + Cv \leq b \quad (3.30)$$

Esta restrição é válida se  $\mu_0$  satisfizer:

$$L\mu_0 \leq b$$

e se a transição  $t_i$  puder disparar a partir do estado actual  $(\mu, v)$  sse:

- a)  $\mu \xrightarrow{t_i} \mu'$
- b)  $L\mu + Hq + Cv \leq b$
- c)  $L\mu' + Cv'$ , onde  $v' = v + q$

onde  $q$  é um vector de disparo em que  $q_i$  é 1 se a transição  $t_i$  se pretender disparável a partir de  $\mu$  e  $v$  é o vector de *Parikh*[9], em que  $v_i$  conta o número de vezes que a transição  $t_i$  disparou desde  $\mu_0$ . Assim, as restrições podem envolver não só a marcação da rede, mas também o disparo de certas transições e o número de vezes que as mesmas disparam. Permite, assim, a imposição de restrições bem mais abrangentes. Através deste vector, por exemplo, pode-se restringir a diferença entre o número de vezes que duas transições disparam a um valor constante. A síntese do controlador para o caso em que a RdP é totalmente controlável e observável pode ser obtido sem recorrer a qualquer transformação da RdP.

#### **Teorema 3.3 (Síntese de Controladores - Controlo e observação total [9])**

Dada a restrição  $L\mu + Hq + Cv \leq b$  e fazendo:

$$\begin{aligned} D_{lc}^+ &= \max(0, -LD - C) \\ D_{lc}^- &= \max(0, LD + C) \end{aligned}$$

O controlador é dado pelas seguintes matrizes de incidência:

$$D_c^+ = D_{lc}^+ + \max(0, H - D_{lc}^-) \quad (3.31)$$

$$D_c^- = \max(D_{lc}^-, H) \quad (3.32)$$

A marcação inicial do controlador  $\mu_{c0}$  é:

$$\mu_{c0} = b - L\mu_0 \quad (3.33)$$

O controlador definido pelas matrizes de entrada e saída  $D_c^+$  e  $D_c^-$  resultantes de 3.31 e 3.32 e com marcação inicial  $\mu_{c0}$  dada por 3.33, impõe 3.30 e é imposição óptima (*optimally enforcing*). ♦

#### **Inclusão de self-loops na RdP**

O teorema 3.3 só garante a optimalidade dos controladores para RdP que não incluam *self-loops*. Isto deve-se ao facto de  $D_c^+$  e  $D_c^-$  do controlador serem obtidas a partir da matriz de incidência  $D$ . Como se viu em 3.1, a existência de *self-loops* numa RdP, não é reflectida por esta matriz. Por isso não se poderá garantir a optimalidade do controlador sintetizado. A solução para esta limitação, passa por transformar a transição que contém o *self-loop* num conjunto de duas transições e um lugar, designada por Transformação-C. Esta transformação será descrita em pormenor mais à frente. De notar também que, sendo o controlador obtido através do cálculo de  $D_c^+$  e  $D_c^-$ , poderá incluir *self-loops* para garantir a imposição de uma restrição.

### 3.2.4 Análise da Admissibilidade das Restrições Lineares Generalizadas

Na presença de transições não controláveis e/ou não observáveis, há que verificar admissibilidade deste tipo de restrições.

#### Definição 3.7 (Admissibilidade de uma RLG)

Seja  $(C, \mu_0)$  uma RdP. Assuma-se que se quer impor um conjunto de restrições 3.30. Considere-se o controlador descrito por 3.31, 3.32 e 3.33. Diz-se que as restrições 3.30 são admissíveis para todos os estados alcançáveis  $(\mu, v)$  da RdP controlada se as seguintes condições forem verdadeiras:

1. Se  $t$  é não controlável e  $t$  é permitida por  $\mu|_C^1$ , então  $t$  é permitida na RdP controlada.
2. Se  $t$  é não observável e  $t$  é permitida por  $\mu$ , o disparo de  $t$  não muda a marcação dos lugares de controlo.

Desta definição tira-se, imediatamente, uma **condição suficiente** para a admissibilidade de uma RLG:

#### Proposição 3.1

As restrições 3.30 são admissíveis se para todas as marcações iniciais se  $D_{c,uc}^-$  e  $D_{c,uo}$  forem matrizes nulas. •

#### Exemplo 3.2

Voltando à figura 3.3 do exemplo 3.1, a matriz de incidência do sistema e a marcação inicial são dadas por:

$$D_p = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \quad \mu_{p_0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \end{bmatrix}$$

Neste exemplo, pretende-se que haja uma correcta sincronização entre o braço robot (que larga a peça) e o tapete (que recebe e transporta a peça). Para que tal aconteça, é necessário impor que o tapete esteja livre (marca no lugar  $p_7$  ou disparo de  $t_8$ ) para que o robot possa largar a peça (disparo de  $t_4$ ) e, por outro lado, que o tapete só possa começar a transportá-la (disparo de  $t_6$ ) quando o braço a tiver largado (disparo de  $t_5$ ). Estas são restrições impostas ao disparo de transições, pelo que uma boa solução será usar a restrição linear generalizada e o vector de *Parikh*. Analisando as RdPs, conclui-se que é necessário recorrer às restrições 3.34 e 3.35. De notar que, em ambas as restrições, a transição que leva ao disparo da outra possui coeficiente negativo e que os valores de  $b$  se devem ao facto de  $t_4$  disparar mais uma vez do que  $t_7$ , o que não se verifica entre  $t_6$  e  $t_5$ .

$$v_4 - v_7 \leq 1 \quad (3.34)$$

$$v_6 - v_5 \leq 0 \quad (3.35)$$

Representando estas restrições sob a forma de 3.30:

$$L = 0 \quad H = 0 \quad C = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & -1 & 1 & 0 & 0 \end{bmatrix} \quad b = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

Neste sistema todas as transições são observáveis. No entanto, todas as transições que representam fins de acções são não controláveis. Analisando o sistema, conclui-se que  $t_2, t_3, t_5, t_7, t_8$  são não controláveis. Surge, por isso, necessidade de verificar se as restrições propostas são admissíveis. Para tal, assume-se que não existem transições não controláveis, calcula-se a matriz  $D_c^-$  e testa-se a proposição 3.1. De 3.32 resulta que:

$$D_c^- = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

e a partir desta:

$$D_{c,uc}^- = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

<sup>1</sup> $\mu|_C$  significa a restrição de  $\mu$  aos lugares de  $C$  (RdP original)

Uma vez que não há transições não observáveis e sendo a matriz  $D_{c,uc}^-$  nula conclui-se, a partir da proposição 3.1, que as restrições 3.34 e 3.35 são admissíveis.

Após o cálculo de  $D_c^+$ , a partir de 3.31,

$$D_c^+ = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

é possível obter a matriz de incidência do controlador  $D_c$ . De 3.13 e 3.18 obtém-se a matriz de incidência do sistema controlado e a marcação inicial dos lugares de controlo, respectivamente.

$$D = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 1 \\ -1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & -1 & 0 & 0 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$$

Todos os resultados apresentados neste exemplo podem ser obtidos com a função *Linenf.m* da *Matlab SPN-BOX*, desenvolvida pelos autores de [9] e [11], onde foram implementados os métodos por eles desenvolvidos.

Os lugares  $c_1$  e  $c_2$ , da figura 3.4, representam o controlador sintetizado que satisfaz as restrições 3.34 e 3.35. ■

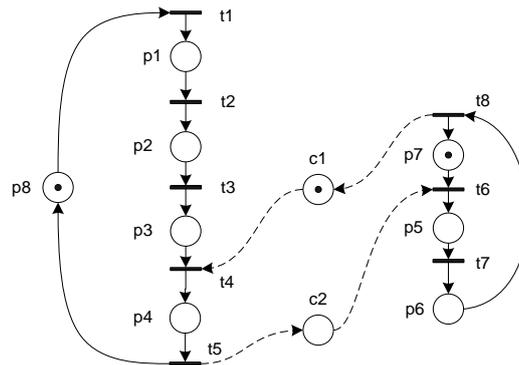


Figura 3.4: RdP com o controlador

### Determinação da admissibilidade de uma restrição

Caso não se verifiquem as condições da proposição 3.1, é necessário recorrer a outras formas de análise. Em [9] são apresentados dois métodos para verificar as condições da definição 3.7.

O primeiro consiste em percorrer exaustivamente a **árvore de alcançabilidade da RdP** em busca de:

1. Transições não permitidas em  $\mu$  mas permitidas em  $\mu|_C$ .
2. Mudanças na marcação de lugares de controlo devido ao disparo de transições não observáveis.

A ocorrência de, pelo menos, uma destas situações, revela a inadmissibilidade da respectiva restrição.

O segundo método é baseado na **análise estrutural da RdP** controlada. Seja  $T_{r,uc}$  o conjunto de transições não controláveis com colunas não nulas de  $D_{c,uc}^-$  e  $T_{r,uo}$  o conjunto de transições não observáveis com colunas não nulas de  $D_{c,uo}$ . O algoritmo seguinte pode ser usado para identificar restrições não admissíveis para alguma marcação inicial.

1. Assumir  $v = 0$  e verificar se existem marcações iniciais  $\mu$  do processo para as quais:
  - (a) a relação 3.30 é satisfeita.

- (b)  $\exists t \in T_{r,uc}$ :  $t$  é permitida pelo processo e inibida pelo controlador.

Se existem tais marcações  $\mu$ , sair e declarar que 3.30 como inadmissíveis para alguma marcação inicial.

2. Assumir  $v = 0$  e verificar se existem marcações iniciais  $\mu$  do processo para as quais:

- (a) 3.30 é satisfeita.

- (b)  $\exists t \in T_{r,u0}$ :  $t$  é permitida na RdP controlada.

Se existem tais marcações  $\mu$ , sair e declarar que 3.30 como inadmissíveis para alguma marcação inicial.

3. Repetir 1. e 2. para  $v \geq 0$ . Se não existir nenhum  $\mu$  e  $v$  para os quais nenhuma das condições de 1. e 2. são satisfeitas, declarar 3.30 admissíveis para todas as marcações iniciais que as satisfaçam.

### 3.2.5 Transformação de Restrições Lineares Generalizadas

Quando uma restrição é admissível, pode ser imposta como descrito no teorema 3.3. No entanto, quando uma restrição não é admissível ou nada se consegue concluir acerca da sua admissibilidade, é possível transformá-la para uma forma que se reconheça como tal. Pretende-se, dado um conjunto de restrições 3.30 a uma RdP  $(C, \mu_0)$ , encontrar um conjunto de restrições admissíveis:

$$L_a\mu + H_aq + C_av \leq b_a \quad (3.36)$$

tal que se  $P_c$  é um controlador óptimo que impõe 3.36 em  $(C, \mu_0)$ , então para  $\forall(\mu, v) \in R(C, \mu_0, P_c)$ : 3.30 é satisfeita.

O método adoptado consiste em transformar a RdP de modo a que as restrições 3.30 sejam mapeadas sob forma de restrições de marcação 3.9. Sob esta forma, as restrições podem ser transformadas em restrições admissíveis, como descrito em 3.2.2.

#### Transformação-C

Esta operação destina-se a transformar o termo  $Cv$  de 3.30 num termo de marcação. A ideia é adicionar à RdP um lugar por cada cada transição envolvida em  $Cv$ . Estes lugares vão receber uma marca cada vez que a respectiva transição for disparada, funcionando como um contador de disparos. O exemplo seguinte ilustra isto mesmo:

#### Exemplo 3.3

Dada a RdP da figura 3.5(a).

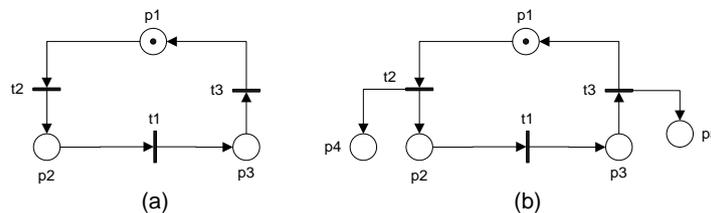


Figura 3.5: Exemplo de uma Transformação-C

Suponha-se que se quer impor uma restrição da forma 3.30:

$$\mu_1 + q_1 + v_2 - v_3 \leq 3 \quad (3.37)$$

Para proceder à Transformação-C, transforma-se a RdP da figura 3.5(a) na RdP da figura 3.5(b). Assim, a restrição 3.40 pode ser escrita de seguinte forma:

$$\mu_1 + q_1 + \mu_4 - \mu_5 \leq 3 \quad (3.38)$$

A inversa da Transformação-C também é possível. Dada a restrição:

$$\mu_1 - 3\mu_4 - 2\mu_5 + q_1 \leq 5 \quad (3.39)$$

por Transformação-C inversa, pode ser mapeada na figura 3.5 como:

$$\mu_1 + q_1 - 3v_2 - 2v_3 \leq 5. \quad (3.40)$$

Em [9] são formalizadas a Transformação-C e a sua inversa.

### Transformação-H

Esta operação destina-se a transformar o termo  $Hq$  de 3.30 num termo de marcação. A ideia é dividir a transição a que se refere a restrição em duas transições e um lugar, como descrito na figura 3.6 envolvida em  $Cv$ . Estes lugares vão receber uma marca cada vez que a respectiva transição for disparada, funcionando como uma memória do disparo da respectiva transição. O exemplo seguinte ilustra isto mesmo:

#### Exemplo 3.4

Dada a RdP da figura 3.6(a).

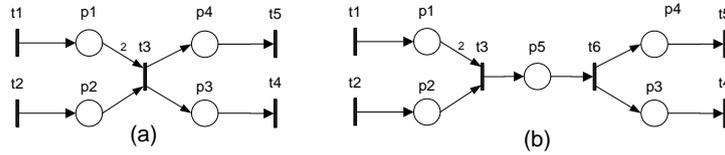


Figura 3.6: Exemplo de uma Transformação-H

Suponha-se que se quer impor uma restrição da forma 3.30:

$$\mu_1 + \mu_2 + 2\mu_3 + q_3 \leq 5 \quad (3.41)$$

Para proceder à Transformação-H, transforma-se a RdP da figura 3.6(a) na RdP da figura 3.6(b). Assim, a restrição 3.41 pode ser escrita da seguinte forma:

$$\mu_1 + \mu_2 + 2\mu_3 + 4\mu_5 \leq 5 \quad (3.42)$$

Onde o termo  $4\mu_5$  é obtido da seguinte forma: Considere-se o disparo de  $t_3$  na RdP transformada.  $\mu \xrightarrow{t_3} \mu'$  e  $a$  é o coeficiente de  $\mu_5$ , tem-se que:

$$a + \mu'_1 + \mu'_2 + 2\mu'_3 = 1 + \mu_1 + \mu_2 + 2\mu_3$$

onde o factor 1 é o coeficiente de  $q_3$  em 3.41. Daqui se obtém  $a = 4$ .

Em [9] são formalizadas as Transformação-H e a sua inversa.

### 3.2.6 Algoritmo para a transformação de uma restrição numa restrição admissível

O algoritmo apresentado em [9] baseia-se nas transformações definidas anteriormente. Dada uma restrição da forma 3.30 não admissível, efectuam-se as transformações C e H, de forma a obter uma restrição da forma  $L_{HC}\mu_{HC} \leq b$ . A esta, aplica-se o método descrito em 3.2.2 de modo a obter uma restrição admissível  $L_{HCa}\mu_{HC} \leq b_a$ . No caso de ser impossível o cálculo de tal restrição, o algoritmo declara impossível obter uma restrição admissível. Em caso de sucesso, resta aplicar as transformações H e C inversas para se chegar à desejada restrição admissível  $L_a\mu + H_aq + C_av \leq b_a$ . O teorema seguinte assegura a optimalidade do controlador obtido por este algoritmo.

#### Teorema 3.4

Assuma-se que o algoritmo não falha no cálculo de  $L_{HCa}\mu_{HC} \leq b_a$ . Então  $L_a\mu + H_aq + C_av \leq b_a$  é admissível e o controlador óptimo que a impõe, impõe também  $L\mu + Hq + Cv \leq b$ .

### 3.3 Métodos de Análise

Existem duas metodologias de análise das RdP podem ser classificadas em dois grupos:

1. método do grafo de alcançabilidade;
2. análise linear algébrica.

#### 3.3.1 Análise Algébrica Linear

As equações de estado 3.5 descrevem o comportamento dinâmico de sistemas concorrentes modelados por RdPs. A solvabilidade destas equações está, de certa forma, limitada pela obrigatoriedade de soluções inteiras não negativas.

#### Condição Necessária da Alcançabilidade

Supondo-se que uma marcação  $\mu_d$  é atingível a partir de  $\mu_0$  através de uma sequência de disparos  $q_1, q_2, \dots, q_d$ , de 3.5, obtém-se:

$$\mu_d = \mu_0 + D \sum_{k=1}^d q_k \quad (3.43)$$

que pode ser reescrita como:

$$Dv = \Delta\mu \quad (3.44)$$

onde  $v = \sum_{k=1}^d q_k$  e  $\Delta\mu = \mu_d - \mu_0$ .  $v$  será mais tarde definido como o vector de *Parikh* [9], vector coluna de dimensão  $n$  em que cada elemento corresponde ao número de vezes que a transição da respectiva posição disparou. Para existir solução para o sistema 3.44, é necessário que  $\Delta\mu$  seja ortogonal a todas as soluções  $y$  do seguinte sistema homogéneo [13]:

$$A^T y = 0 \quad (3.45)$$

Desta forma, é condição necessária, para que  $\mu_d$  seja acessível a partir de  $\mu_0$ , que a solução  $v$  exista e possua apenas valores inteiros não negativos. O facto de não existir essa solução, é suficiente para a não alcançabilidade da marcação.

A análise dos invariantes de marcação e de disparo, apresentados em 3.1.4, permite verificar várias condições relativamente à vivacidade e limitação de uma RdP. Estas condições são apresentadas e provadas em [3].

#### Proposição 3.2 (Condição Necessária para a Vivacidade)

Numa RdP viva sem lugares isolados, todos os  $p$ -invariantes  $x$  que não contiverem elementos negativos satisfazem  $x\mu_0 > 0$ . •

#### Proposição 3.3 (Condição Suficiente para a Limitação)

Toda a RdP com um  $p$ -invariante  $x$  satisfazendo  $x_i > 0$  para todos os lugares  $i$  é limitada. •

#### Proposição 3.4 (Condição Necessária para a Vivacidade e Limitação)

Toda a RdP viva e limitada possui um  $t$ -invariante  $y$  que satisfaz  $y_j > 0$  para cada transição  $j$  da rede. •

Este tipo de análise não apresenta, para algumas propriedades, respostas inequívocas acerca do comportamento de uma RdP. No entanto, pode ser um auxílio importante para análise das mesmas, quando usada a par de outras técnicas de análise.

### 3.3.2 Grafo de Cobertura

Dada uma RdP, é possível obter, a partir da marcação inicial, tantas marcações novas quantas as transições habilitadas. A partir das marcações novas obtidas, pode obter-se, de novo, mais marcações. A representação atrás descrita apresenta um problema: se a RdP for ilimitada, a representação crescerá infinitamente. No entanto, pode obter-se um grafo finito. Para tal, sempre que ocorra uma marcação repetida, termina esse ramo do grafo. Para uma marcação inicial  $M_0$ , diz-se que o estado  $M'$  está coberto se  $M'(i) = M(i)$  para todos os lugares pertencentes à rede.

Nesse sentido, introduz-se um símbolo especial  $\omega$ , que pode ser visto como "infinito". Em [13, p. 550] é apresentado um algoritmo para obtenção do grafo de cobertura.

Algumas propriedades podem ser estudadas a partir deste grafo:

- 1) Uma RdP é limitada e logo  $R(M_0)$  é finito sse  $\omega$  não aparece em nenhum dos nós do grafo.
- 2) Uma RdP é segura sse as marcações de todos os nós são zero ou um.
- 3) Uma transição é morta se não aparece como arco de um nó do grafo.
- 4) Se  $M$  é atingível a partir de  $M_0$ , então existe um nó chamado  $M'$  tal que  $M \leq M'$ .

Para uma RdP limitada, o grafo de cobertura é chamado grafo de alcançabilidade já que contém todas as possíveis marcações alcançáveis.

Apesar das enormes potencialidades, este método é muitas vezes difícil de aplicar devido à sua natureza fortemente combinatória, resultando numa "explosão" de estados. No entanto, pode-se simplificar a RdP para contornar este problema.

A redução de uma RdP é um procedimento que transforma uma RdP para outra rede simplificada, mantendo as propriedades desejáveis da rede original. As técnicas de redução de RdPs encontram-se na secção 3.3.3.

Com o intuito de minimizar o número de estados, existem algumas soluções que reduzem o tamanho do grafo durante a sua construção:

- Nós duplicados - quando um estado já existente é inserido, ele é considerado apenas um caminho para a primeira ocorrência desse estado, não sendo gerados novamente os seus estados sucessores.
- *Stubborn Set* - trata transições disparáveis que são independentes entre si, ou seja, transições que podem ser disparadas em qualquer ordem antes de ser obtido o próximo estado. Ao invés de serem considerados todas as possíveis combinações dessas transições, apenas uma das possibilidades é considerada.
- Conjunto de componentes - são apenas considerados alguns componentes do sistema modelado, durante a construção do grafo de alcançabilidade.

### 3.3.3 Técnicas de redução

Quando um sistema apresenta uma dimensão tal, que torna mais complicado o uso das técnicas anteriormente descritas, é comum reduzir o modelo do mesmo a um outro mais simples. Estas técnicas de redução devem, naturalmente, preservar as propriedades do sistema original que se pretendem estudar. As simplificações da figura 3.7 foram apresentadas em [13] e asseguram vivacidade, segurança e limitação. Na figura 3.7 estão representadas as seguintes reduções:

- (a) Fusão de lugares em série.
- (b) Fusão de transições em série
- (c) Fusão de lugares em paralelo
- (d) Fusão de transições em paralelo.

Para além deste tipo de reduções existe outro tipo de transformação que simplificam a análise e interpretação de uma RdP, as **transformações hierárquicas**. A introdução de *macro-lugares* em substituição de algumas sub-redes sequenciais e a remoção de alguns lugares concorrentes são dois exemplos deste tipo de transformações que preservam a vivacidade e a limitação.

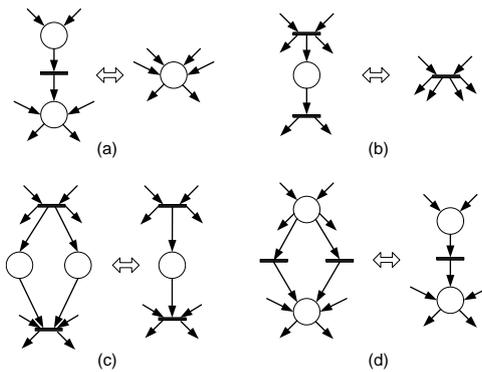


Figura 3.7: 4 simples transformações que preservam vivacidade, segurança e limitação.

**Exemplo 3.5 (Análise de uma RdP)**

Imagine-se que a RdP da figura 3.4 é o sistema no qual se quer efectuar a supervisão. Então, antes da modelação das falhas e adição das mesmas, é preciso analisar a RdP de modo a identificar e classificar as suas propriedades essenciais. Esta análise permitirá avaliar o efeito da adição das falhas ao sistema.

Antes da análise da rede, é possível efectuar algumas das transformações de simplificação descritas em 3.3.3. A subrede constituída pelos lugares  $p_1, p_2, p_3$  e pelas transições  $t_1$  e  $t_2$  é uma rede sequencial. Por isso, torna-se possível a substituição dessa subrede por um *macro-lugar*. O mesmo acontece, olhando para  $p_5, p_6$  e  $t_6$  como uma subrede. De notar que esta transformação hierárquica poderá, também, ser usada na modelação das falhas, visto que dentro de cada subrede referida ocorrem o mesmo tipo de falhas e as recuperações são idênticas. Trata-se de uma abstracção para um nível hierárquico superior. Tem-se, então, a rede da figura 3.8, em que os macro-lugares são representados pelos círculos de maior diâmetro.

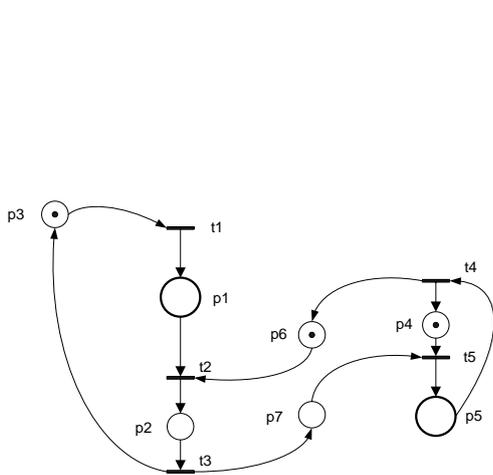


Figura 3.8: RdP Simplificada

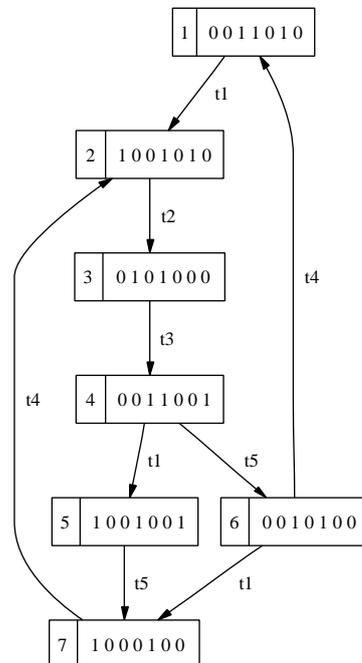


Figura 3.9: Grafo de alcançabilidade da RdP da figura 3.8

A matriz de incidência e a marcação inicial são agora:

<sup>1</sup>os nomes dos lugares e transições foram alterados de modo a facilitar a interpretação da RdP

$$D = \begin{bmatrix} 1 & -1 & 0 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 \\ 0 & 0 & 0 & -1 & 1 \\ 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & -1 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$

A RdP possui, agora, uma dimensão que torna menos complexa a análise através do respectivo grafo de alcançabilidade, apresentada na figura 3.9.

Directamente do grafo de alcançabilidade concluem-se as seguintes propriedades:

1. A RdP é 1-limitada.
2. A RdP é segura, já que é 1-limitada.
3. A RdP não possui transições mortas.
4. A marcação inicial  $\mu_0$  é atingível a partir de todas as marcações da árvore
5. A partir da marcação inicial  $\mu_0$ , a RdP não possui *deadlocks*.

A propriedade **4.** é especialmente interessante, já que assegura o carácter cíclico da tarefa, isto é, a marcação/estado inicial da RdP/sistema é sempre retomado depois da execução da tarefa. Esta é uma característica que será fundamental manter depois da modelação das falhas.

A RdP possui os seguintes invariantes de disparo e marcação:

$$y = [1 \quad 1 \quad 1 \quad 1 \quad 1] \quad x = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \end{bmatrix}$$

A primeira coluna de  $x$  é o invariante correspondente ao funcionamento do braço, a segunda coluna é o invariante do tapete e a terceira coluna corresponde ao invariante inerente à interacção dos dois componentes. O carácter cíclico do sistema é, mais uma vez, realçado, já que toda a rede é coberta por um só invariante de disparo. Nenhum destes invariantes viola as proposições 3.2, 3.3 e 3.4 para a vivacidade e limitação de uma RdP. A partir da análise da árvore de alcançabilidade e dos invariantes da RdP, não se concluiu a existência de bloqueios.

Estas serão as ferramentas com que se efectuará a modelação do *SED* associado à *CFE* a estudar.

As metodologias aqui apresentadas serão aquelas que permitirão a síntese de controladores para efectuarem a supervisão do *SED* resultante.

## CAPÍTULO 4

# MODELAÇÃO E CONTROLO DA *CFE*

Antes de proceder à supervisão da *CFE* e da consequente modelação das falhas que podem ocorrer, é necessário obter um modelo em RdP da *CFE* e do seu sistema de controlo. Este é um procedimento que influenciará a possibilidade ou a impossibilidade de recuperar certas falhas e tem, por isso, crucial importância.

A modelação pretende aproximar-se, o mais possível, do sistema implementado. Ainda assim, há que ter em conta as limitações inerentes à modelação de sistemas por RdPs e que a tornam mais pobre, principalmente ao nível do sequenciamento.

Todas as RdPs são obtidas recorrendo à síntese de controladores baseados em invariantes de marcação apresentado no capítulo 3. A admissibilidade foi analisada para todas as restrições. No entanto, será apenas referida para os casos em que as restrições se revelem não admissíveis, estando, por isso, implícita a admissibilidade de todas as outras.

Obtidas as RdPs que modelam cada um dos componentes da *CFE*, procede-se à sincronização das mesmas para, assim, se obter uma RdP que modele todo o sistema. No final deste capítulo analisa-se a RdP obtida para a caracterizar quanto às suas propriedades fundamentais e definir quais destas será imprescindível manter após a adição das extensões que modelam as falhas e permitem efectuar a supervisão do sistema.

### 4.1 Modelação dos Elementos da *CFE*

Nesta secção, serão modelados individualmente todos os elementos da *CFE*. As RdPs resultantes baseiam-se nas sequências básicas de funcionamento de cada componente e na imposição de restrições estruturais de controlo inerentes às rotinas de controlo de cada um deles, usadas na configuração actual da *CFE*.

Depois de obtidas as redes básicas, procedeu-se à síntese dos controladores de cada um dos componentes da *CFE*. Estes controladores têm como função impor restrições estruturais e físicas do próprio componente e restrições de sequenciamento de acções impostas pelas rotinas de controlo associadas a cada um deles. Não serão ainda apresentados quaisquer tipos de sincronismo entre componentes, sendo a principal preocupação a correcta modelação de cada um.

Os controladores foram obtidos com base nos métodos apresentados na secção 3.2 do capítulo 3. Foi usada a *toolbox* de *Matlab SPNBOX*, desenvolvida pelos autores de [9] e [11], onde se aplicam os métodos por eles desenvolvidos. Os arcos que interagem com os lugares de controlo serão representados a linha tracejada.

Note-se que nesta subsecção a modelação dos componentes é local e individual. Devido a esse facto, a nomenclatura usada na modelação das RdP somente diz respeito ao componente sujeito a análise.

#### 4.1.1 Tapetes

Os tapetes, da figura 2.2, podem ser vistos como recursos. Por isso, sendo o tapete *T1* estruturalmente igual ao tapete *T2* e o tapete *T3* estruturalmente igual ao tapete *T4*, podemos representar estes quatro componentes em apenas duas redes, impondo as respectivas restrições nos invariantes de marcação. Este tipo de restrições é bastante comum em problemas de gestão de recursos.

#### T1 e T2

Os tapetes *T1* e *T2* têm, exactamente, as mesmas sequências de funcionamento. Estes tapetes são solicitados para efectuarem o transporte das peças da entrada da *CFE* para a saída e no sentido contrário. Estas duas acções nunca são efectuadas na mesma sequência. Por isso, as RdPs de *T1* e de *T2* são constituídas por duas sequências distintas. A RdP equivalente é apresentada na figura 4.1. Para estes dois componentes, modelados pela RdP da figura 4.1, concluíram-se necessárias as seguintes condições:

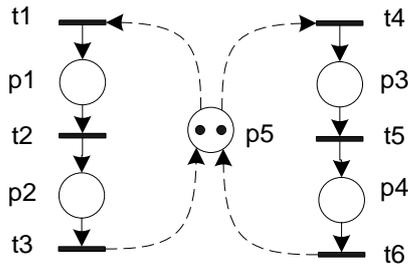
Figura 4.1: RdP dos Tapetes  $T1$  e  $T2$ 

Tabela 4.1: Legenda da RdP dos Tapetes 1 e 2

Lugares	Descrição
$p_1, p_3$	em transporte
$p_2, p_4$	em espera
Transições	
$t_1$	avança
$t_2$	avançou
$t_3, t_6$	fim de transporte*
$t_4$	recua
$t_5$	recuou*

\*transições não controláveis

1. Não se podem executar as duas sequências de funcionamento ao mesmo tempo
2. Existem dois tapetes deste tipo
3. Só uma peça é transportada em cada sequência

Estas três condições são simultaneamente impostas por uma só restrição linear:

$$\mu_1 + \mu_2 + \mu_3 + \mu_4 \leq 2.$$

Pode provar-se que a restrição é admissível e o que controlador sintetizado é representado pelo lugar  $p_5$  da RdP da figura 4.1. A marcação inicial de  $p_5$  é 2 e, como se esperava, é sinónimo da existência de dois tapetes do mesmo tipo.

### T3 e T4

Tal como os anteriores, os tapetes  $T3$  e  $T4$  apresentam iguais comportamentos. A sua função é transportar a peça até a um hipotético controlo de qualidade e trazê-la de volta, acabando aí o processo de fabrico da mesma. Estes componentes apenas possuem uma sequência de funcionamento e deverão ser impostas as seguintes restrições à RdP da figura 4.2:

1. Só uma peça é transportada de cada vez
2. Existem dois tapetes deste tipo

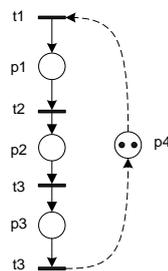


Figura 4.2: RdP dos Tapetes 3 e 4

Tabela 4.2: Legenda da RdP dos Tapetes 3 e 4

Lugares	Descrição
$p_1$	em transporte
$p_2$	em espera
$p_3$	em transporte
Transições	
$t_1$	avança
$t_2$	avançou*
$t_3$	recua
$t_4$	fim de transporte*

\*transições não controláveis

Estas duas condições são impostas sob forma de uma só restrição linear:

$$\mu_1 + \mu_2 + \mu_3 \leq 2.$$

Da imposição desta restrição, resulta o controlador associado ao lugar  $p_4$  da RdP da figura 4.2.

A marcação inicial de duas marcas em  $p_4$  deve-se ao facto de este mesmo lugar funcionar como um gestor de recursos, isto é, dois tapetes com funcionamento paralelo.

### 4.1.2 Máquinas CNC

A CFF é constituída, como se viu em 2.1.4, por duas fresadoras e um torno CNC. Embora efectuem operações de maquinação claramente distintas, as fresadoras e o torno têm seqüências de funcionamento idênticas, isto é, do ponto de vista do controlo, os seus comportamentos são exactamente iguais. Daí que seja possível obter uma RdP igual para as três máquinas.

As máquinas só possuem uma seqüência de funcionamento, ou seja, quando solicitadas executam sempre a mesma seqüência de operações, com excepção para o caso em que a mesma peça está sujeita a vários programas de maquinação. As acções executadas são:

1. Carregamento do programa CNC
2. Fecho do aperto da peça
3. Maquinação
4. Abertura do aperto da peça

Com estas acções, analisando a rotina de controlo dedicada às máquinas, chegou-se à RdP da figura 4.3.

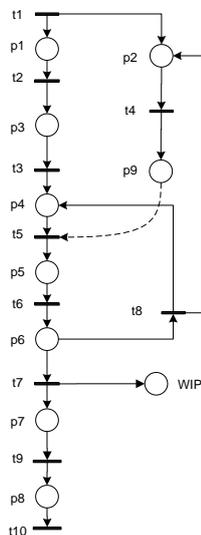


Figura 4.3: RdP da máquina

Tabela 4.3: Legenda da RdP dos Tapetes 1 e 2

Lugares	Descrição
$p_1, p_4, p_7$	em espera
$p_2$	carregando programa
$p_3$	fechando aperto
$p_5$	em maquinação
$p_6$	encaminha peça
$p_8$	abrindo aperto
Transições	
$t_1$	Carrega Programa
$t_2$	Fecha aperto
$t_3$	Aperto fechado*
$t_4$	Programa carregado*
$t_5$	Inicia Maquinação
$t_6$	Fim de maquinação*
$t_7$	Encaminha Peça
$t_8$	Novo programa
$t_9$	Abre aperto
$t_{10}$	Aperto fechado*

\*transições não controláveis

O encaminhamento da peça, após a maquinação, é feito pelo sequenciador. Para isso, após o fim da maquinação, é acrescentada uma marca à lista de peças em fabrico (*WIP - Work In Progress*) através do disparo de  $t_7$ .

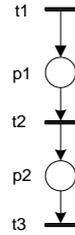
Como se observa na figura 4.3, a máquina é o único componente da célula que apresenta concorrência de acções. O carregamento do programa ( $p_2$ ) é efectuado em simultâneo com a recepção da peça e o consequente fecho do aperto ( $p_3$ ). Para assegurar que a máquina só inicia a maquinação da peça ( $t_5$ ) depois de o programa ter sido correctamente carregado ( $t_4$ ), recorre-se à síntese de um controlador que imponha a restrição

$$v_5 - v_4 \leq 0.$$

Daqui resultou o lugar de controlo  $p_9$ .

### Buffers das máquinas

Como se viu no capítulo 2, há um buffer associado a cada máquina. Se a máquina estiver ocupada, as peças são depositadas no *buffer* até que esta fique livre. A rede que representa o buffer está descrita na figura 4.4. A interação dos *buffers* com as respectivas máquinas será tratada mais à frente, neste capítulo, juntamente com a sincronização destas com o sequenciador.

Figura 4.4: RdP do buffer *B2*Tabela 4.4: Legenda da RdP do *buffer*

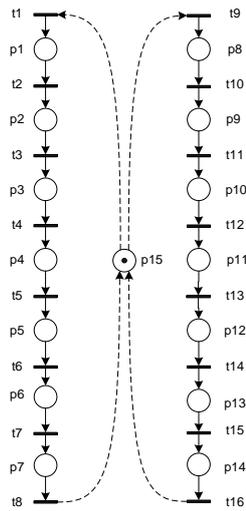
Lugares	Descrição
$p_1$ $p_2$	posições ocupadas a desocupar
Transições	
$t_1$	ocupa posição
$t_2$	liberta posição
$t_3$	posição libertada

### 4.1.3 Braços Robot

A função dos manipuladores robóticos da *CFF*, descritos em 2.1.1, é o transporte de peças. Existindo unicamente esta função, as sequências de funcionamento de ambos podem ser divididas em duas etapas: Agarrar uma peça na sua origem e largá-la do respectivo destino. Depois de enumerados e analisadas todas as possíveis origens e destinos, procedeu-se à correspondência entre ambos, com o auxílio de controladores.

#### Robot *B2*

Na figura 4.5 verifica-se que os dois ramos da rede equivalem às duas tarefas *B2* efectua. No primeiro ramo

Figura 4.5: RdP do braço *B2*Tabela 4.5: Legenda a RdP de *B2*

Lugares	Descrição
$p_1$	em deslocamento
$p_2, p_4, p_6, p_9, p_{11}, p_{13}$	em espera
$p_3, p_{10}$	a agarrar
$p_5, p_8, p_{12}$	em deslocamento
$p_7, p_{14}$	a largar
Transições	
$t_1$	vai ao armazém
$t_2, t_6, t_{10}, t_{14}$	chegou*
$t_3, t_{11}$	agarra peça
$t_4, t_{12}$	agarrou*
$t_5$	vai para o T1/T2
$t_7, t_{15}$	larga peça
$t_8, t_{16}$	largou*
$t_9$	vai para o T1/T2
$t_{13}$	vai para o T3/T4

\*transições não controláveis

( $t_1$  a  $t_5$ ) observa-se que, desde que seja solicitado pelo robot *B1*, o robot está habilitado para transportá-la até aos tapetes *T1* ou *T2*. Os restantes nós da rede modelam o transporte dos tapetes *T1* ou *T2* para os tapetes *T3* ou *T4*.

É necessário assegurar a exclusão mútua destas duas trajectórias, o que se consegue impondo um invariante de marcação que abranja todos os lugares da rede:

$$\mu_1 + \mu_2 + \dots + \mu_{14} \leq 1$$

Este invariante é controlado pelo lugar de controlo  $p_{15}$

#### Robot *B1*

O robot *B1* tem uma tarefa bastante mais complexa, na medida em que tem 8 componentes que, na prática, se traduzem como possíveis origens ou destinos. Deste facto, resulta uma vasta gama de combinações entre os componentes, que será tratada mais à frente. A RdP da figura 4.6 modela o comportamento *B1*.

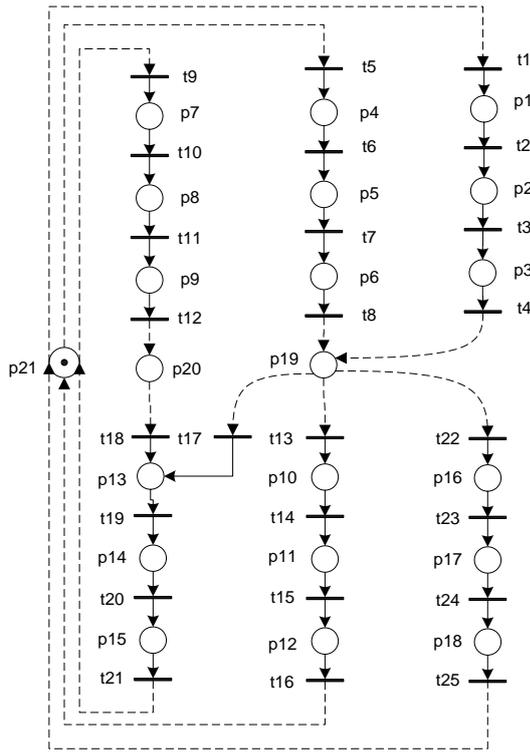


Figura 4.6: RdP do braço *BI*

Tabela 4.6: Legenda da RdP de *BI*

Lugares	Descrição
$p_1, p_7$	em deslocamento
$p_2, p_5, p_8$	em espera
$p_3, p_4, p_9$	a agarrar
$p_6$	a retirar
$p_{10}, p_{16}$	em deslocamento c/carga
$p_{11}, p_{14}, p_{17}$	em espera
$p_{12}, p_{15}, p_{18}$	a largar
$p_{13}$	alimenta máquina
Transições	
$t_1, t_{13}$	vai para o tapete 1/2
$t_2, t_{10}$	chegou*
$t_3, t_{11}$	agarra peça
$t_4, t_6, t_{12}$	agarrou*
$t_5$	agarra peça na máquina
$t_7$	retira peça
$t_8^*$	retirou
$t_9, t_{22}$	vai para o <i>buffer</i>
$t_{14}, t_{19}, t_{23}$	chegou*
$t_{15}, t_{24}$	larga peça
$t_{16}, t_{25}$	largou*
$t_{17}, t_{18}$	alimenta máquina
$t_{19}$	Alimentou
$t_{20}$	sai da máquina
$t_{21}$	saiu*

\*transições não controláveis

Como visto anteriormente, há que restringir o comportamento do robot na realização de certas trajectórias de transporte de peças. É, então, necessário impor as seguintes condições, que se podem reformular sob forma de restrições lineares generalizadas:

1. O único destino para as peças que saem do buffer é a respectiva máquina.

$$v_{18} - v_{12} \leq 0$$

2. Todas as outras combinações origem/destino ficam ao critério do sequenciador, sendo apenas preciso assegurar o fluxo entre todos eles:

$$(v_{13} + v_{17} + v_{22}) - (v_4 + v_8) \leq 0$$

Todas estas restrições são admissíveis e, a partir delas, sintetizaram-se os controladores que as impõem. As restrições 1., 2. são impostas pelos lugares de controlo  $p_{20}$ ,  $p_{19}$ , respectivamente.

Há ainda que garantir que apenas é executada uma tarefa de cada vez. Isto traduz-se na imposição de um invariante de marcação que cubra todos os lugares da rede:

$$\mu_1 + \mu_2 + \dots + \mu_{19} + \mu_{20} \leq 1$$

Esta restrição é imposta pelo lugar de controlo  $p_{21}$ .

A RdP da figura 4.6 modela o comportamento do robot *BI*.

### Sequenciador

O sequenciamento de acções e a gestão dos recursos disponíveis são elementos preponderantes em qualquer sistema de produção. A aplicação de controlo da *CFF*, descrita no capítulo 2, faz uso de um sequenciador que, de acordo com políticas predefinidas, gere todos os recursos da produção e encaminha, de acordo com a disponibilidade destes, todas as peças em fabrico.

As RdPs usadas neste trabalho não permitem a modelação exacta de um sistema deste tipo, pela sua complexidade e elevado número de componentes envolvidos. Por esta razão, optou-se por modelar um sequenciador que garanta apenas um correcto fluxo de peças pela CFF e comande as acções dos robots e das máquinas. As políticas de sequenciamento poderão ser implementadas na modelação desenvolvida recorrendo à atribuição de diferentes prioridades no disparo das transições do sequenciador. No entanto, do ponto de vista da análise e da posterior detecção de falhas, é suficiente que se garanta o fluxo das peças pela CFF, já que neste modelo estão incluídas todas as possíveis políticas de sequenciamento.

O sequenciador escolhe uma peça de duas listas, armazém ou peças em fabrico (WIP), define o seu destino e ordena aos robots o seu transporte. Para que o robot *B1* efectue o transporte de uma peça, é necessário obter a origem e o destino da mesma, pelo que são necessárias as quatro transições de saída, referentes às quatro trajetórias de transporte possíveis. A RdP da figura 4.7 apresenta a RdP do sequenciador.

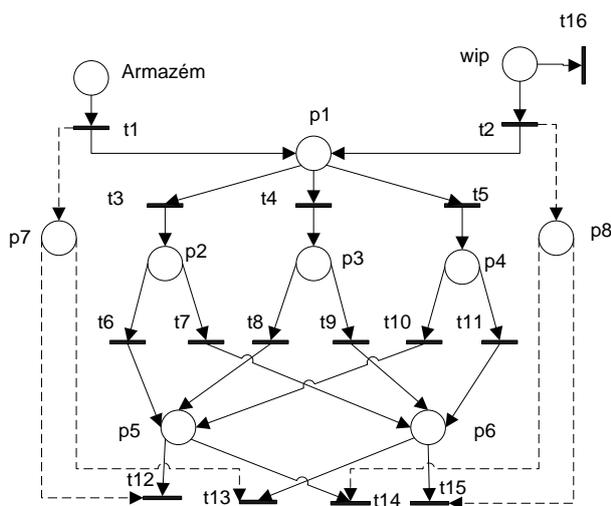


Figura 4.7: RdP do Sequenciador

Tabela 4.7: Legenda da RdP do sequenciador

Lugares	Descrição
$p_{armazem}$	Armazém
$p_{wip}$	WIP (peças em fabrico)
$p_1$	escolhe peça
$p_2, p_3, p_4$	máquinas 1, 2 e 3
$p_5$	destino buffers
$p_6$	destino máquinas
Transições	
$t_1$	escolhe do armazém
$t_2$	escolhe do WIP
$t_3, t_4, t_5$	vai para máquina 1, 2 ou 3
$t_6, t_8, t_{10}$	destino buffers 1, 2 ou 3
$t_7, t_9, t_{11}$	destino maquinação 1, 2 ou 3
$t_{12}$	armazém → buffer
$t_{13}$	armazém → máquina
$t_{14}$	máquina → buffer
$t_{15}$	máquina → máquina
$t_{15}$	máquina → saída

É necessário impor na rede da figura 4.7 duas restrições a esta rede: a primeira assegura que  $t_{12.seq}$  e  $t_{13.seq}$  só disparam se a peça tiver sido escolhida do armazém; a outra assegura que  $t_{13.seq}$  e  $t_{14.seq}$  só disparam se a peça tiver sido escolhida do armazém da lista WIP. Estas restrições podem ser reformuladas sob a forma de restrições ao vector de *Parikh*:

$$\begin{aligned} (v_{12} + v_{13}) - v_1 &\leq 0, \\ (v_{15} + v_{14}) - v_2 &\leq 0 \end{aligned}$$

Estas restrições são admissíveis e impostas pelos lugares  $p_7$  e  $p_8$ , respectivamente.

Na secção seguinte deste capítulo, procede-se à sincronização entre o sequenciador e os diversos componentes da célula, com recurso à síntese de controladores baseados em invariantes de marcação.

### Armazém

O armazém aloca as matérias brutas que serão maquinadas. Deste facto, depreende-se a sua fácil modelação, que resulta num lugar que contém o número de peças a maquinar.

Tal como foi apresentado no capítulo 2, existem os armazéns *A1* e *A2*. Em cada armazém existe uma única posição para *B2* apanhar as peças em bruto. Desta forma, e à semelhança do sistema físico, a entrada de peças no sistema modelado só se deve verificar quando as posições de *A1* e *A2* forem vagas por *B2* (disparo de  $t_{5.b2}$ ). Para garantir este sincronismo no sistema é necessário impor a seguinte restrição:

$$v_{1.seq} - v_{5.b2} \leq 2$$

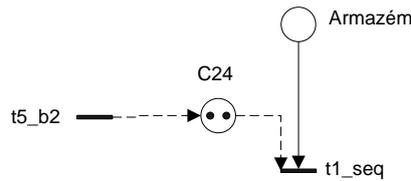


Figura 4.8: Entrada de peças no sistema vindas do armazém

Esta restrição é imposta pelo lugar  $C_{24}$ . Na figura 4.8 está representado o armazém e o lugar de controlo que permite a entrada de peças no sistema.

## 4.2 Imposição de Restrições de Controlo

Obtidas as redes que modelam cada um dos componentes da *CFE* é, então, possível juntá-las numa só rede, de acordo com o sistema de controlo implementado.

O seguimento rígido da estrutura de controlo já implementado é bastante complexo e muitas vezes impossível através deste tipo de Redes de Petri. Outro tipo de redes, como as *Redes de Petri Coloridas*, permitiriam modelar sistemas de automação baseados em controlo por software de forma bem mais eficiente. Além disto, o sequenciamento das peças é, como em todas as células de fabrico flexível, dinâmico e com elevado número de variáveis. Por exemplo, uma peça pode ser maquinada repetidamente por mais que uma máquina e sem uma ordem rígida. Isto torna, se não impossível, altamente complexa a modelação de todas as possíveis sequências de fabrico. Por estas razões, optou-se por modelar correctamente o fluxo das peças entre os diversos componentes e sequências de fabrico de forma mais geral, tornando a RdP mais abrangente do que o sistema real. Isto é, na RdP há sequências de fabrico que não ocorrem na realidade e que poderão até corresponder a comportamentos erróneos do processo de fabrico. Apelando à teoria de linguagens, apresentada em [7], conclui-se que as propriedades de uma linguagem gerada são válidas para uma linguagem marcada. O mesmo é dizer que uma vez provadas as propriedades para a rede modelada, estarão também provadas as propriedades da rede real, uma vez que esta, em analogia às linguagens, está incluída na primeira. Para outras propriedades em que tal não seja possível, há ainda a possibilidade de restringir a análise apenas às sequências válidas.

A sincronização entre os diversos componentes foi efectuada de forma "incremental", ou seja, foram-se juntando os componentes até que a *CFE* estivesse modelada por uma só rede. Desta forma, não só se obteve maior detalhe na modelação, como também se facilitou a tarefa de obtenção dos controladores. A correcção deste processo é conseguida com base na *máxima permissividade* dos controladores, baseados em invariantes de marcação, assegurando, assim, que apenas as restrições especificadas são impostas pelos controladores, não havendo por isso o risco de alterarem o comportamento global do sistema.

De seguida, são apresentadas as restrições a impor para o correcto sincronismo entre os componentes e as subredes daí resultantes, até se obter uma rede global. De notar que, nas figuras apresentadas, se evitou a descrição das redes na sua totalidade. Para tornar mais legíveis as figuras, apresentam-se apenas as subredes dos componentes que interagem entre si.

### 4.2.1 Interação entre B2 e os tapetes

O braço B2, pelo que já foi referido, basicamente larga e agarra peças dos tapetes T1/T2 e larga peças nos tapetes T3/T4. Há que assegurar o correcto sincronismo destas tarefas:

1. B2 só larga a peça nos tapetes T3/T4 ( $t_{4.T34}$ ) se um deles estiver livre ( $t_{15.b2}$ ). Esta restrição é semelhante à primeira restrição do exemplo 3.2. De forma a usar apenas o vector de *Parikh*, diz-se que a diferença entre a transição que marca  $p_{4.T34}$  ( $t_{4.T34}$ ) e  $t_{15.b2}$  deve ser 2 (2 tapetes):

$$v_{4.T34} - v_{15.b2} \leq 2$$

Esta restrição é imposta pelo lugar  $C_1$ .

2.  $T3/T4$  só avançam ( $t_{1\_T34}$ ) depois de  $B2$  ter largado a peça ( $t_{16\_b2}$ ):

$$v_{1\_T34} - v_{16\_b2} \leq 0$$

Esta restrição é imposta pelo lugar  $C_2$ .

3. Assim que  $T1/T2$  ( $t_{5\_T12}$ ) recuam transportando uma peça,  $B2$  pode agarrá-la ( $t_{11\_b2}$ ):

$$v_{11\_b2} - v_{5\_T12} \leq 0$$

Esta restrição é imposta pelo lugar  $C_3$ .

4. Depois de  $B2$  agarrar a peça ( $t_{12\_b2}$ ),  $T1/T2$  ficam livres ( $t_{6\_T12}$ ):

$$v_{6\_T12} - v_{12\_b2} \leq 0$$

Esta restrição é imposta pelo lugar  $C_4$ .

5.  $T1/T2$  só avançam ( $t_{1\_T12}$ ) depois de  $B2$  ter largado a peça ( $t_{8\_b2}$ ):

$$v_{1\_T12} - v_{8\_b2} \leq 0$$

Esta restrição é imposta pelo lugar  $C_5$ .

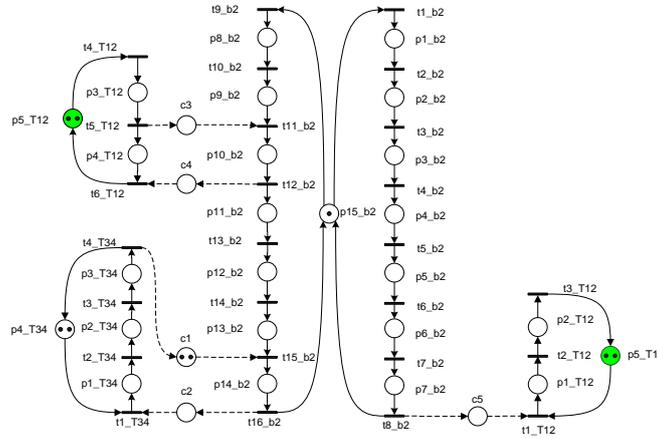


Figura 4.9: Interação de B2 com os tapetes

#### 4.2.2 Interação entre B1 e os tapetes T1 e T2

O braço  $B1$  apenas interage com  $T1$  e  $T2$ , largando ou agarrando peças. O sincronismo entre estes componentes é assegurado pelas seguintes restrições:

1.  $B1$  agarra peça ( $t_{3\_b1}$ ) se  $T1/T2$  avançarem ( $t_{2\_T12}$ ):

$$v_{3\_b1} - v_{2\_T12} \leq 0$$

Esta restrição é imposta pelo lugar  $C_6$ .

2.  $T1/T2$  só ficam livres ( $t_{3\_T12}$ ) depois de  $B1$  ter agarrado a peça ( $t_{4\_b1}$ )

$$v_{3\_T12} - v_{4\_b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_7$ .

3.  $T1/T2$  recuam ( $t_{4\_T12}$ ) assim que  $B1$  largar uma peça ( $t_{16\_b1}$ ).

$$v_{4\_T12} - v_{16\_b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_8$ .

Na figura 4.10 está representada a interação entre B1 e os tapetes T1 e T2, através dos lugares de controlo obtidos anteriormente.

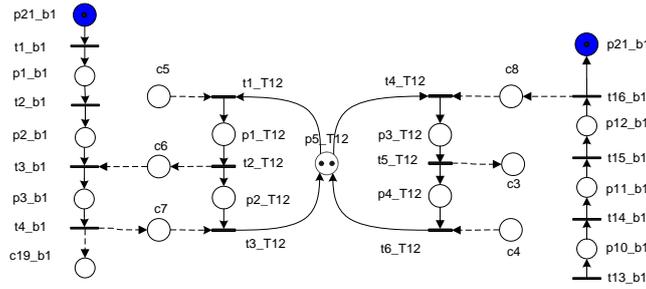


Figura 4.10: Interação de B1 com T1/T2

### 4.2.3 Interação entre o sequenciador e os restantes componentes

#### Interação entre o sequenciador e as máquinas

Como se viu anteriormente, é o sequenciador que decide qual a peça a maquina e em que máquina será maquina ou, se esta estiver ocupada, o encaminhamento para o respectivo *buffer*. Essa decisão é, na RdP do sequenciador, modelada pelo disparo de transições. Essas transições podem, recorrendo a controladores baseados em restrições no vector de disparos, comandar o envio de peças para as máquinas ou respectivos *buffers*.

Uma vez nos *buffers*, o único destino possível para as peças é a maquinação na respectiva máquina. Por esta razão, será o disparo de  $t_{3\_buffi}$  a encaminhar a peça para a máquina  $i^1$ . Podem-se reformular estas restrições no vector de disparos sob a forma de restrições lineares generalizadas.

A maquinação de peças ( $t_{1\_maqi}$ ) é habilitada pelo sequenciador ( $t_{7\_seq}, t_{9\_seq}$  ou  $t_{11\_seq}$ ) ou pelo *buffer* ( $t_{3\_buffi}$ ), da respectiva máquina.

$$v_{1\_maq1} - (v_{7\_seq} + v_{3\_buff1}) \leq 0$$

$$v_{1\_maq2} - (v_{9\_seq} + v_{3\_buff2}) \leq 0$$

$$v_{1\_maq3} - (v_{11\_seq} + v_{3\_buff3}) \leq 0$$

Estas restrições são impostas pelos lugares de controlo  $P_{in\_maq1}, P_{in\_maq2}$  e  $P_{in\_maq3}$ , respectivamente.

Se a máquina estiver ocupada, o sequenciador decide o envio para o *buffer* ( $t_{6\_seq}, t_{8\_seq}$  ou  $t_{10\_seq}$ ) e permite a entrada de peças no mesmo ( $v_{1\_buffi}$ ):

$$v_{1\_buff1} - v_{6\_seq} \leq 0$$

$$v_{1\_buff2} - v_{8\_seq} \leq 0$$

$$v_{1\_buff3} - v_{10\_seq} \leq 0$$

Estas restrições são impostas pelos lugares de controlo  $P_{in\_buff1}, P_{in\_buff2}$  e  $P_{in\_buff3}$ , respectivamente. É preciso, agora, controlar o acesso às máquinas e aos buffers.

1. Uma só peça em maquinação:

$$\mu_{2\_buffi} + \mu_{in\_maqi} + \mu_{1\_maqi} + \mu_{3\_maqi} + \mu_{4\_maqi} + \mu_{5\_maqi} + \mu_{6\_maqi} + \mu_{7\_maqi} + \mu_{8\_maqi} \leq 1, \text{ para } i = 1, 2, 3.$$

Estas restrições são impostas pelos lugares de controlo  $P_{10\_maqi}$  com  $i=1,2,3$ .

2. *Buffer* com capacidade para 3 peças:

$$\mu_{in\_buffi} + \mu_{1\_buffi} + \mu_{2\_buffi} \leq 3, \text{ para } i = 1, 2, 3.$$

Estas restrições são impostas pelos lugares de controlo  $P_{3\_buffi}$  com  $i=1,2,3$ .

3. Uma peça só vai para o *buffer* se a máquina estiver ocupada:

$$q_{6\_seq} - (\mu_{2\_buff1} + \mu_{in\_maq1} + \mu_{1\_maq1} + \mu_{3\_maq1} + \dots + \mu_{8\_maq1}) \leq 0$$

$$q_{6\_seq} - (\mu_{2\_buff2} + \mu_{in\_maq2} + \mu_{1\_maq2} + \mu_{3\_maq2} + \dots + \mu_{8\_maq2}) \leq 0$$

$$q_{6\_seq} - (\mu_{2\_buff3} + \mu_{in\_maq3} + \mu_{1\_maq3} + \mu_{3\_maq3} + \dots + \mu_{8\_maq3}) \leq 0$$

<sup>1</sup> $i$  corresponde ao número da máquina em causa

De notar que estas restrições não são admissíveis, já que controlam directamente  $t_{10\_maq_i}$ , transições não controláveis por representarem um fim de acção. Aplicando o algoritmo da transformação de restrições, descrito em 3.2, obtiveram-se as restrições admissíveis equivalentes. Estas restrições remetem o controlo de  $t_{10\_maq_i}$  para as transições  $t_{9\_maq_i}$ , estas sim controláveis, garantindo, da mesma forma, a optimalidade dos controladores. Estas restrições são impostas pelos lugares de controlo  $P_{11\_maq1}$ ,  $P_{11\_maq2}$  e  $P_{11\_maq3}$ , respectivamente.

Os controladores obtidos encontram-se nas figuras A.1 e A.3.

### Interacção entre o sequenciador e os robots

Depois de definidos os trajectos da peça escolhida pelo sequenciador, é necessário comandar os braços robots para que estes efectuem o transporte da mesma. Isto pode ser feito recorrendo, mais uma vez, a restrições no vector de disparos da RdP e sintetizando controladores para essas restrições.

Se a origem da peça for o armazém ( $t_{12\_seq}$  ou  $t_{13\_seq}$ ), permite-se os robots efectuar as seguintes acções:

1. O robot *B2* transporta uma peça do armazém para os tapetes *T1* ou *T2*.
2. O robot *B1* vai buscar uma peça aos tapetes *T1* ou *T2*.

Estas restrições podem ser apresentadas sob forma das restrições lineares generalizadas

$$\begin{aligned} v_{1\_b2} - (v_{12\_seq} + v_{13\_seq}) &\leq 0 \\ v_{1\_b1} - (v_{12\_seq} + v_{13\_seq}) &\leq 0 \end{aligned}$$

Estas restrições são admissíveis e são impostas pelos lugares de controlo  $C_9$  e  $C_{11}$ , respectivamente.

Se a origem da peça for uma máquina ( $t_{14\_seq}$ ,  $t_{15\_seq}$  ou  $t_{16\_seq}$ ), habilita-se o robot *B1* a ir a essa mesma máquina buscar a peça. Isto é conseguido impondo a restrição

$$v_{5\_b1} - (v_{14\_seq} + v_{15\_seq} + v_{16\_seq}) \leq 0.$$

Esta restrição é admissível e é imposta pelo lugar de controlo  $C_{12}$ .

Depois de efectuadas as acções relativas à origem da peça, há que assegurar o correcto encaminhamento para o destino escolhido. Se o destino for um dos *buffers* ( $t_{12\_seq}$  ou  $t_{14\_seq}$ ), habilita-se o ramo da RdP do robot *B1* correspondente à colocação de peça no buffer. Disto resulta a restrição

$$v_{22\_b1} - (v_{12\_seq} + v_{14\_seq}) \leq 0$$

Esta restrição é admissível e é imposta pelo lugar de controlo  $C_{13}$ .

Se o destino, de uma peça, for uma máquina ( $t_{13\_seq}$  ou  $t_{15\_seq}$ ), o procedimento é semelhante, habilitando-se o ramo do braço correspondente à colocação de peça na máquina. Temos, então, a restrição admissível

$$v_{17\_b1} - (v_{13\_seq} + v_{15\_seq}) \leq 0,$$

imposta pelo lugar  $C_{14}$ .

Finalmente, quando o sequenciador decide o fim de fabrico de uma peça ( $t_{16\_seq}$ ), é necessário:

1. Definir, como destino do robot, *B1* os tapetes *T1* ou *T2*.
2. Permitir ao robot *B2* o transporte da peça dos tapetes *T1* ou *T2* para os tapetes *T3* ou *T4*

As restrições que impõem, à RdP, este comportamento, são:

$$\begin{aligned} v_{13\_b1} - v_{16\_seq} &\leq 0 \\ v_{9\_b2} - v_{16\_seq} &\leq 0 \end{aligned}$$

Ambas admissíveis e asseguradas pelos lugares de controlo  $C_{15}$  e  $C_{16}$ , respectivamente.

Sempre que um dos braços vai colocar uma peça nos tapetes *T1* ou *T2*, o sequenciador terá que reservar uma posição num desses tapetes. Sob a forma de restrições no vector de disparo, isto é dado por:

$$(v_{12\_seq} + v_{13\_seq} + v_{16\_seq}) - (v_{3\_T12} + v_{6\_T12}) \leq 2,$$

que origina o lugar de controlo  $C_{10}$ .

Os controladores obtidos, nesta subsecção, estão representados nas figuras A.1, A.2 e A.4.

#### 4.2.4 Interacção entre robot *B1* e as máquinas

##### Interacção entre robot *B1* e os *buffers* das máquinas

1. Se o robot *B1* largou a peça no buffer, ocupa-se uma posição "física" do *buffer*:

$$\sum_{i=1}^3 v_{1.bufferi} - v_{25.b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{17}$ .

2. Se o *buffer* libertou uma posição (disparo de  $t_{2.bufferi}$ ), então o robot pode ir buscar uma peça (disparo de  $t_{9.b1}$ ):

$$v_{9.b1} - \sum_{i=1}^3 v_{2.bufferi} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{18}$ .

3. O *buffer* só liberta efectivamente a posição (disparo de  $t_{3.bufferi}$ ) quando o robot tiver agarrado a peça (disparo de  $t_{12.b1}$ ):

$$\sum_{i=1}^3 v_{3.bufferi} - v_{12.b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{19}$ .

Os controladores obtidos, nesta subsecção, encontram-se representados nas figuras A.3 e A.4.

##### Interacção o robot *B1* e as máquinas

O transporte de peças para as máquinas é, como se viu, efectuado pelo robot *B1*. Para tal, depois de habilitado para proceder a esse transporte, há que garantir o completo sincronismo entre as acções de abrir e fechar o aperto, por parte das máquinas, e o agarrar e largar a peça, por parte do braço. Para isto, impõem-se as seguintes restrições:

1. Para retirar a peça da máquina, *B1* deve informar quando a agarrou, disparo de  $t_{6.b1}$ .

$$v_{9.maqi} - v_{6.b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{20}$ .

2. *B1* só pode retirar a peça quando o aperto da máquina estiver aberto, disparo de  $t_{10.maqi}$ .

$$v_{7.b1} - \sum_{i=1}^3 v_{10.maqi} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{21}$ .

3. As máquinas só fecham o aperto depois de *B1* ter colocado a peça no mesmo, disparo de  $t_{20.b1}$ .

$$\sum_{i=1}^3 v_{2.maqi} - v_{19.b1} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{22}$ .

4. *B1* sai da máquina quando a peça estiver fixa pelo aperto, disparo de  $t_{3.maqi}$ .

$$v_{20.b1} - \sum_{i=1}^3 v_{3.maqi} \leq 0$$

Esta restrição é imposta pelo lugar  $C_{23}$ .

A *CFE* está agora modelada por uma só RdP. A complexidade da rede não permite a sua representação gráfica num só bloco. Por isso, optou-se por apresentar, no apêndice A, cada um dos módulos constituintes.

Resta agora proceder à análise das propriedades da RdP, para que depois se modelem, a partir da mesma, algumas das mais significativas falhas e respectivas recuperações.



### Condição Necessária para a Vivacidade e Limitação

A rede é coberta por t-invariantes. Logo, esta condição (3.4) é satisfeita. Nada se pode, no entanto, concluir acerca da vivacidade da RdP.

Com base nas condições atrás expostas, infere-se que a rede é limitada e que a vivacidade da mesma não é garantida. Não se conseguindo, desta forma, garantir a vivacidade da rede, resta proceder à análise exaustiva da rede. Esta análise, efectuada com o cálculo do grafo de alcançabilidade da rede, fornecerá todas as marcações alcançáveis a partir do estado inicial da mesma e a informação acerca da vivacidade das mesmas,

#### 4.3.2 Grafo de Alcançabilidade

Na elaboração do grafo de alcançabilidade desta RdP, a "explosão" de estados apresentou-se como uma dificuldade acrescida. Uma vez a rede reduzida, foi necessário recorrer às soluções descritas em 3.3.2 para minimizar o número de estados durante a construção do grafo de alcançabilidade. A construção deste grafo, utilizando o método *Stubborn Set*, só foi possível porque este preserva o carácter cíclico da rede e garante, caso eles existam, a detção de todos os estados mortos da rede.

Na tabela 5.3, encontram-se compilados os resultados da análise à rede, para diferentes ordens de produção e utilizando, ou não, técnicas de redução na construção do grafo de alcançabilidade. A figura 5.10 serve para mostrar a elevada redução que se obtém utilizando o método *Stubborn Sets*.

As análises foram efectuadas num Computador Intel Pentium IV a 2400Mhz, com 256 Megabyte de memória RAM.

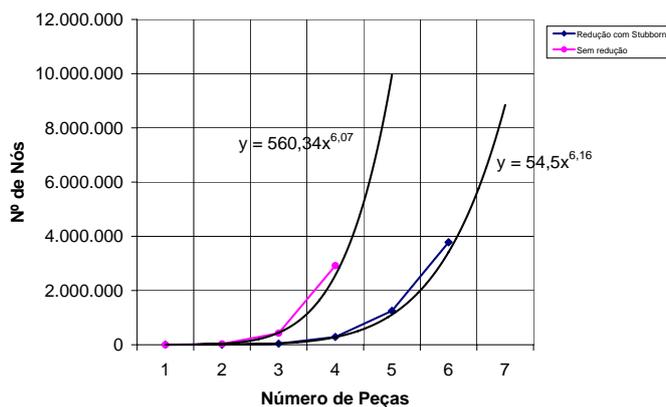


Tabela 4.8: Resultados da análise efectuada à RdP

Peças	$R$	$R_{stubb}$
1	622	70
2	313.25	2.788
3	426.646	40.758
4	2.919.694	288.322
5	$> 9 \times 10^6$	1.250.706
6	-	3.780.684

Figura 4.12: Relação N° de peças/Nós do Grafo de alcançabilidade

Utilizando a ferramenta acima mencionada, calculou-se o grafo de alcançabilidade da respectiva rede e, até lotes de seis peças, não foi detectado qualquer nó terminal. Isto equivale a dizer que, a partir da marcação inicial, a rede não alcança qualquer marcação morta. Daqui se conclui que, até lotes de produção de seis peças, a RdP não apresenta, a partir da marcação inicial, qualquer bloqueio (*deadlock*).

O cálculo do grafo de alcançabilidade, para lotes de mais de seis peças, tornou-se computacionalmente incontrolável pelo que, para lotes de maiores, nada se poderá concluir acerca da vivacidade da rede.



## CAPÍTULO 5

# SUPERVISÃO DA *CFE* - MODELAÇÃO DAS FALHAS

O objectivo principal do supervisor é acrescentar à *CFE* a capacidade de reconhecimento e recuperação de falhas sem a intervenção do operador ou, sempre que isto seja impossível, sem que seja interrompida a sessão de fabrico. Não estando qualquer falha prevista no controlador actual, é necessário enumerar e descrever as falhas a detectar e, seguidamente, modelar a sua detecção e consequente recuperação.

Esta modelação leva a apender à RdP original um conjunto de lugares, arcos e transições que conservam todas as propriedades interessantes já analisadas e provadas no capítulo 3. Baseia-se na adição de extensões constituídas por duas transições (uma de falha e outra de recuperação) e um lugar de recuperação. Este lugar modela um estado de recuperação em que, no sistema real, serão enviadas acções de recuperação para a *CFE*, até que se retorne ao funcionamento normal. Esta retoma do estado de funcionamento normal é, em termos de RdPs, modelado pelo disparo da transição de recuperação.

A conservação de propriedades reveste-se de capital importância, uma vez que garantirá a correcta modelação das mesmas e da sua integração no modelo existente. É possível garantir que, mediante determinadas condições, a rede resultante conserva as propriedades da original. Para esse efeito apresenta-se o seguinte Lema, sem demonstração<sup>1</sup>:

### **Lema 5.1 (Conservação de Propriedades de Vivacidade)**

*Seja C uma RdP com marcação inicial  $\mu_0$  e F o conjunto de todas as marcações de falha acrescentadas. Se se garantir que existe uma marcação  $M_n \in R(\mu_0)$  tal que*

$$Mf[t_r > M_n, \forall Mf \in F,$$

*então, a RdP resultante conserva as propriedades de vivacidade da RdP original.* ◻

Garantindo que, em todos os estados/marcações de falha introduzidos pelo disparo da respectiva transição de falha, há sempre uma transição de recuperação ( $t_r$ ) disparável que leva a uma marcação alcançável a partir da marcação inicial da RdP original, então a RdP resultante conserva as propriedades da original, no que diz respeito à vivacidade.

No que respeita à modelação das falhas, é conveniente caracterizá-las quanto à sua abrangência. Assim, optou-se por dividir as falhas em três principais grupos: falhas locais a um lugar ou transição, locais a um elemento/módulo e falhas globais.

### **P-locais**

Falhas P-locais são aquelas que ocorrem devido à terminação anormal de um estado ou devido à não ocorrência de um determinado evento e cuja recuperação passa, apenas, pelo regresso a esse mesmo estado, depois de um processo de recuperação que apenas intervém na entidade a que falha está associada.

### **Locais a um módulo**

Falhas locais a um módulo são semelhantes às falhas P locais, mas em que não é possível retomar o estado imediatamente anterior à falha. O processo de recuperação apenas intervém na entidade/módulo a que falha está associada.

---

<sup>1</sup>notação introduzida no capítulo 3

## Globais

Falhas globais são aquelas que afectam mais do que uma entidade/módulo. O processo de recuperação intervém em duas ou mais entidades da célula. Este tipo de falhas apresenta maior complexidade, uma vez que obriga ao sincronismo entre a recuperação de diversos elementos.

A elevada complexidade da RdP da CFF torna imperativa a aplicação deste método de modelação a um sistema menos complexo, a fim de validar e motivar o mesmo. No apêndice B, são apresentados alguns exemplos simples de falhas e procede-se à análise detalhada de cada tipo de falha.

## 5.1 Falhas da CFF

De um variado conjunto de falhas passíveis de serem modeladas, escolheram-se as que se seguem, por duas principais razões: permitem uma demonstração elegante e têm uma expressividade evidente, além de possuírem uma componente pedagógica do método introduzido anteriormente. Correspondem, também, a falhas da CFF em que a detecção não obriga à presença de sensores ou canais de comunicação adicionais.

### 5.1.1 Braços Robot

Escolheu-se, para modelação e detecção, uma falha associada a cada um dos robots: colisão quando o robot *B1* alimenta a máquina ( $F_{B1}$ ); colisão quando o robot *B2* tenta retirar a matéria-prima de armazém ( $F_{B2}$ ).

Tabela 5.1: Falhas do Braço Robot

Falha	Quando ocorre	Detecção	Recuperação	Tipo de falha
$F_{B1}$	entrada de matéria-prima na máquina	evento	<ol style="list-style-type: none"> <li>1. Colocar peça em local de segurança ou posto de CQ (controlo de qualidade)</li> <li>2. Recuperar o robot da colisão</li> <li>3. Reiniciar o robot e a máquina</li> </ol>	Global
$F_{B2}$	Saída de matéria prima do armazém	evento	<ol style="list-style-type: none"> <li>1. Recuperar o braço da colisão</li> <li>2. Esperar recolocação da matéria-prima em armazém</li> <li>3. Voltar ao estado anterior à falha</li> </ol> ou <ol style="list-style-type: none"> <li>1. Recuperar o braço da colisão</li> <li>2. Ignorar a posição de armazém</li> <li>3. Reiniciar elementos envolvidos</li> </ol>	P-Local ou Global*

\*depende do procedimento de recuperação adoptado

### Impacto na entrada de peça na máquina - $F_{B1}$

Pequenas variações na colocação da peça no armazém podem provocar a colisão desta com a base de apoio da máquina. Consequentemente, poderá verificar-se o alarme de impacto por parte do controlador do robot e originar, assim, a falha. Acontecendo numa altura de interacção com a máquina, esta falha é de abrangência global e a sua recuperação implicará dois elementos da CFF.

No que concerne às RdPs, o procedimento de recuperação passa por acrescentar extensões de detecção e recuperação da falha a todos os componentes envolvidos e sincronizá-las com a transição que modela, efectivamente, a ocorrência do evento de falha. Neste caso, a falha pode ocorrer quando o lugar  $P_{13.b1}$  estiver marcado. Depois de efectuadas as acções de recuperação ( $P_{f1.b1}$ ), é reinicializada a marcação da rede de *B1*. A cada máquina acrescenta-se, também, uma extensão ( $t_{fb\_maqi}$ ,  $P_{fb\_maqi}$  e  $t_{rb\_maqi}$ ) que procede à reinicialização da respectiva rede, quando esta está à espera de uma acção do braço em falha. Se considerarmos que uma das

acções de recuperação é a reinicialização da máquina, pode ser restringido o vector de disparos da rede sob a forma de uma restrição linear generalizada:

$$\sum_{i=1}^3 v_{fb\_maqi} - v_{r1\_b1} \leq 0. \tag{5.1}$$

Esta restrição é admissível, uma vez que o controlador resultante irá controlar a transição  $t_{fb-maqi}$  que, por não estar directamente ligada a um evento de falha, é controlável. A figura 5.1 apresenta as extensões associadas a esta falha, bem como os lugares da RdP com as quais interagem directamente.

Para análise do impacto desta adição à RdP, interessa analisar o grafo de alcançabilidade das submarcações envolvidas. Este grafo é apresentado na figura 5.2. A apresentação das marcações no grafo é feita de forma ligeiramente diferente. Para simplificação do próprio grafo, optou-se por representar, em cada nó, apenas os lugares que estão marcados e, entre parêntesis, o número de marcas que possuem. O disparo da transição de

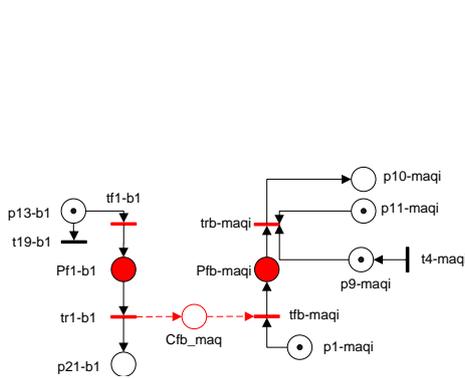


Figura 5.1: Extensões à RdP, correspondentes ao tratamento da falha  $F_{b1}$

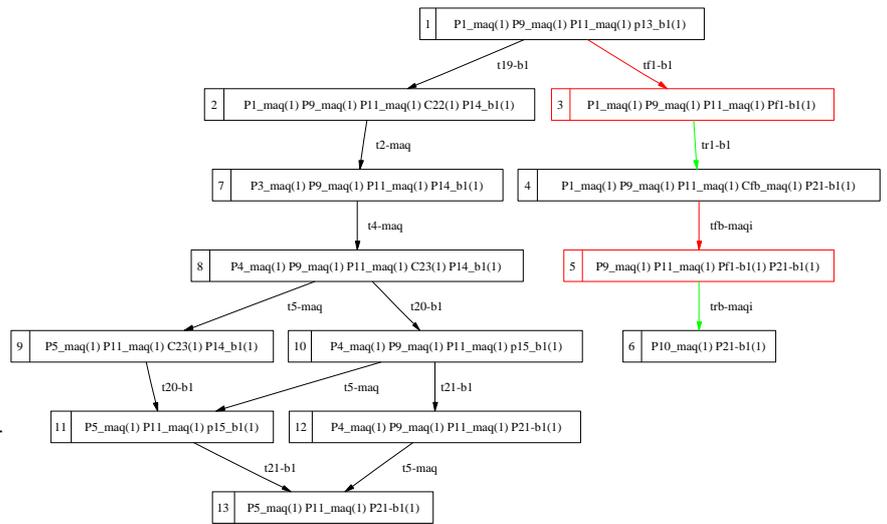


Figura 5.2: Grafo de alcançabilidade que descreve a ocorrência e recuperação de de  $F_{b1}$

falha adiciona ao conjunto de marcações alcançáveis estados de falha (a vermelho). Após a recuperação, são sempre alcançados estados/marcações já existentes na RdP original. Por esta razão, a RdP resultante, para além de manter todas as outras propriedades, continua livre de bloqueios.

### Impacto do robot $B2$ no armazém - $F_{B2}$

A falta de rigor no posicionamento da matéria prima no armazém, por parte do operador ou de um sistema de transporte a montante da célula, pode provocar o impacto do robot  $B2$  e, conseqüentemente, a sua paragem.

Do ponto de vista da RdP, esta falha pode ocorrer quando o lugar  $p_{2\_b2}$  estiver marcado, sinal que o robot  $B2$  se encontra a agarrar a matéria prima do armazém. A caracterização desta falha, quanto à sua abrangência, depende da estratégia de recuperação escolhida. Optou-se por incluir, na modelação, duas formas de recuperação para esta falha:

1. Local - Com intervenção do operador, interrompendo a sessão de fabrico até que este reposicione a matéria prima no armazém. Depois da intervenção do operador, retoma-se o funcionamento a partir do estado imediatamente anterior à ocorrência da falha. Assim, como se poderá ver pelo exemplo apresentado no apêndice B, a transição de falha ( $t_{f\_b2}$ ) retira uma marca ao lugar  $p_{2\_b2}$  e a transição de recuperação ( $t_{r1\_b2}$ ) devolve-a.
2. Global - Sem intervenção do utilizador, ignorando o fabrico da peça em questão e indisponibilizando a respectiva posição de armazém até ao fim da sessão de fabrico terá, também, que ser anulada a reserva dos tapetes  $T1$  ou  $T2$  (marcando  $c_{10}$ ). Na altura em que a falha ocorre (marcação de  $p_{2\_b2}$ ), o

sequenciador já ordenou o encaminhamento para uma das máquinas ou respectivos *buffers*. Por isso, terá que se adicionar, a cada um deles, as devidas extensões para que não permaneçam em bloqueio, à espera que lhes seja depositada a matéria prima bloqueada em armazém. Também o robot *B1* precisará de ser reiniciado, uma vez que já lhe foi ordenado o transporte da matéria prima dos tapetes para uma das máquinas ou respectivo buffer.

Na figura 5.3 estão apresentadas as extensões necessárias ao tratamento destas falhas, bem como os lugares e transições da RdP com os quais interagem.

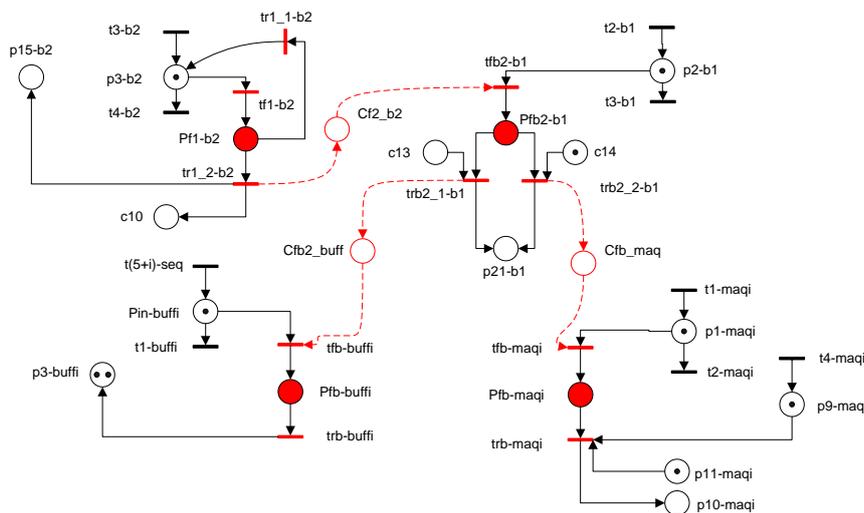


Figura 5.3: Extensões à RdP correspondentes ao tratamento da falha  $F_{B2}$

De reparar que a acção de recuperação das máquinas para esta falha é a mesma que a que se modelou para a falha  $F_{B1}$ . Por isso, é possível aproveitar o controlador  $C_{fb\_maq}$  de modo a que controle, também, a recuperação para esta falha. Para isso, impõe-se uma nova restrição, que irá substituir a restrição 5.1:

$$\sum_{i=1}^3 v_{fb\_maqi} - (v_{r1\_b1} + v_{rb2\_b1}) \leq 0.$$

Quando o destino da matéria-prima for o *buffer*, então a restrição a impor ao disparo das transições é:

$$\sum_{i=1}^3 v_{fb\_buffi} - v_{rb2\_b1} \leq 0.$$

Esta restrição ao vector de *Parikh* é imposta pelo controlador  $C_{fb2\_buff}$ .

A reinicialização do robot *B1* devido à falha do robot *B2* é dada pela restrição:

$$v_{tfb2\_b1} - v_{r1\_b2} \leq 0.$$

Esta restrição ao vector de *Parikh* é imposta pelo controlador  $C_{f2\_b2}$ .

O grafo de alcançabilidade da figura 5.4 correspondem à situação em que o destino da matéria-prima, mal colocada em armazém, é uma das máquinas. Dada a sua semelhança, optou-se por apresentar o grafo de alcançabilidade, para o caso em que o destino seria um dos *buffers*, na figura B.7 do apêndice B. Como se pode observar, também esta falha respeita as propriedades da rede original, uma vez que todos os lugares de falha alcançam lugares alcançáveis a partir da marcação inicial da rede original.

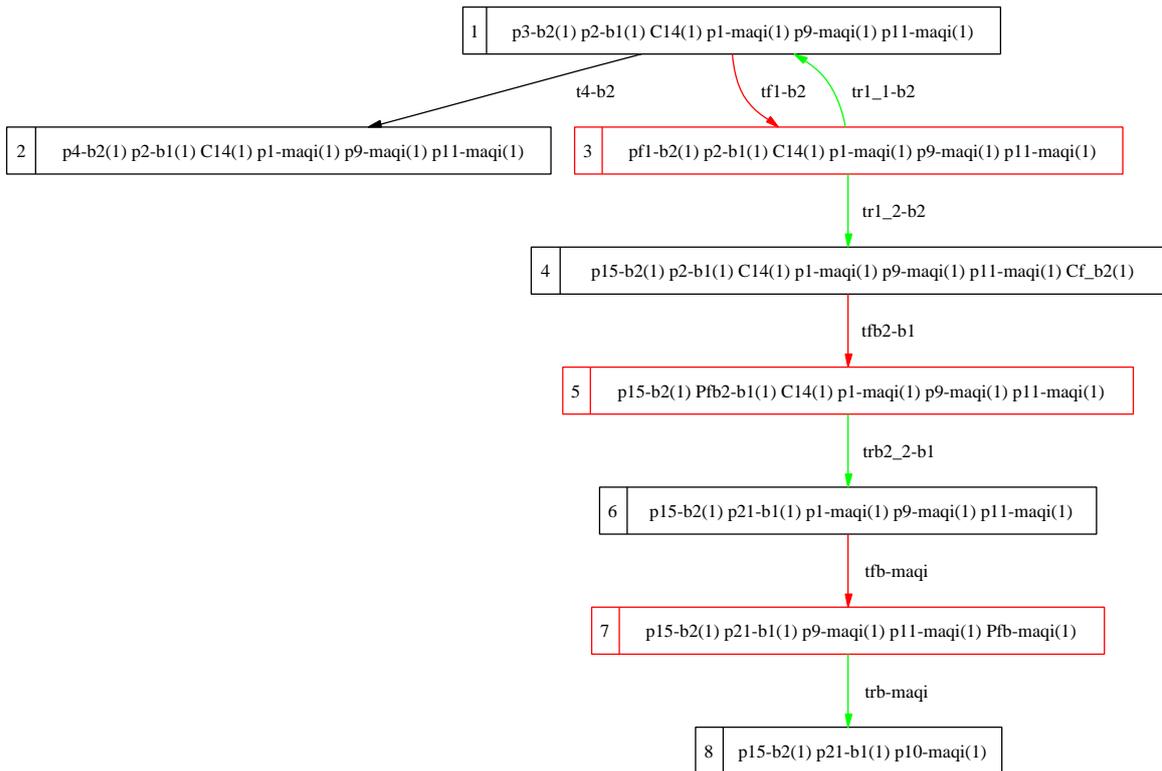


Figura 5.4: Grafo de alcançabilidade que descreve a ocorrência e recuperação de  $F_{B2}$  quando o destino da matéria prima seria uma máquina

### 5.1.2 Máquinas

Às máquinas podem associar-se várias falhas: Erro de carregamento do programa  $F_{M1}$ ; erro de execução de programa  $F_{M2}$ ; paragem de emergência por parte do operador local  $F_{M3}$ ; alarme de segurança  $F_{M4}$ . A tabela 5.2 descreve, em pormenor, cada uma destas falhas. De notar que todas as máquinas possuem o mesmo tipo de falhas, pelo menos do ponto de vista do controlo, pelo que se evita a descrição do processo de modelação cada uma delas.

Identificadas as falhas e caracterizadas as acções que as recuperam, procede-se, agora, à modelação das mesmas nas respectivas RdPs.

#### Erro no carregamento do programa CNC - $F_{M1}$

Durante o carregamento dos programas CNC nas máquinas, podem ocorrer erros de dois tipos: falha da comunicação série do *PC*, ou da máquina, ou formato inválido do ficheiro que contém o código *CNC* a carregar. Ainda que minimizados pelo uso de uma ferramenta *CAD/CAM*, os erros do segundo tipo tornam-se frequentes, se o pós-processador não for suficientemente robusto ou não estiver bem adaptado às máquinas em causa. Os erros devidos a falhas na comunicação não são tão frequentes. No entanto, quando acontecem, é imprescindível a sua detecção e recuperação.

Qualquer que seja a origem do erro, a consequência é, invariavelmente, o bloqueio do sistema. Daí, resulta a necessidade de detectar e resolver a ocorrência desta falha. Optou-se por incluir, na modelação, duas recuperações distintas:

1. Carregar novamente o programa
2. Desistir da maquinação e reencaminhar a matéria prima para a saída

Do ponto de vista da RdP, a falha pode ocorrer quando o lugar  $p_{2.maq}$  de qualquer uma das máquinas estiver marcado, sinal que o programa está a ser carregado. É a este lugar que se acrescentará a extensão de detecção

Tabela 5.2: Falhas nas Máquinas

Falha	Quando ocorre	Deteção	Recuperação	Tipo de falha
$F_{M1}$	Durante o carregamento do programa	<i>timeout</i>	<ol style="list-style-type: none"> <li>1. Tentar carregar novamente</li> <li>2. Se novo erro, eliminar a produção de peças que usem o programa.</li> <li>3. Se não, continuar a produção.</li> </ol>	P-Local ou Local ao Módulo
$F_{M2}$	Durante a maquinação (por erro de código)	<i>timeout</i>	<ol style="list-style-type: none"> <li>1. Pedir a intervenção do operador (tem que ser reinicializada a máquina)</li> <li>2. Esperar o fim da intervenção</li> <li>3. Eliminar a produção de peças que usem o programa com erro</li> </ol>	local ao módulo
$F_{M3}$	Durante a maquinação	evento	<ol style="list-style-type: none"> <li>1. Pedir a intervenção do operador (não é possível parar o estado de emergência automaticamente)</li> <li>2. Esperar o fim da intervenção</li> <li>3. Reiniciar a máquina</li> <li>4. Eliminar a peça das listas de tarefas</li> </ol>	Local ao Módulo
$F_{M4}$	Durante a Maquinação	evento	<ol style="list-style-type: none"> <li>1. Interromper a maquinação</li> <li>2. Esperar a desactivação do alarme de segurança</li> <li>3. Retomar a produção</li> </ol>	P-Local

e recuperação. Para a recuperação 1., esta falha é *P-Local* e para a recuperação 2., a falha é *local ao módulo* máquina. A figura 5.5 apresenta, de forma simplificada, a rede resultante.

Para verificação da validade e do correcto funcionamento desta modelação, é interessante analisar o grafo de alcançabilidade relativo à sub-marcação da máquina. Para isto, é preciso assumir que o sincronismo entre o braço e a máquina está assegurado, de acordo com a modelação efectuada no capítulo 4. Então, como forma de simplificação, ignoram-se os lugares de fusão relativos ao transporte da peça e têm-se em conta apenas aqueles que intervêm directamente na recuperação da falha. Sendo, esta, uma falha local ao módulo, não será necessária a análise de toda a rede. O grafo resultante está representado na figura 5.9 e mostra a evolução da RdP da máquina a partir da marcação onde pode ocorrer a falha ( $\mu_{6\_maq} = 1$ ). Como seria de esperar, o disparo da transição  $t_{f1\_maq}$  leva a rede a um estado de falha (desenhado a vermelho), de onde só sairá após o disparo de uma das transições de recuperação  $t_{r1.1\_maq}$  ou  $t_{r1.2\_maq}$ . Se a recuperação indicar o recarregamento do programa, é atingida a marcação imediatamente anterior à ocorrência da falha. Caso contrário, a transição de recuperação encaminha a peça para fora da máquina, através da marcação no lugar *WIP* e evita a maquinação da mesma.

### Erro de código/maquinação CNC - $F_{M2}$

A geração de código *CNC*, por parte do pós-processador, não possui nenhum processo de validação. Por isso, podem ocorrer erros de maquinação, inerentes a erros no código gerado. Por exemplo: um deslocamento da fresa, que se situe fora da zona de trabalho da mesma, gera um erro que interrompe e bloqueia a maquinação. Estes erros só são detectados, pela unidade de processamento da máquina, já no decorrer da maquinação. A recuperação passa pela reinicialização manual da *CNC*.

Como se viu, esta falha pode ocorrer durante a maquinação. Este estado está associado à marcação do lugar  $p_{5\_maq}$  de cada uma das máquinas. É deste lugar que a transição de falha  $t_{f2\_maq}$  retirará uma marca, aquando da ocorrência da mesma. Na figura 5.6, está representada a extensão associada a esta falha e os lugares da rede da máquina, cujas marcações são alteradas pela mesma.

É possível, através do grafo de alcançabilidade, analisar o comportamento da rede depois da adição da referida extensão. Na figura 5.9, está representado o grafo de alcançabilidade da RdP das máquinas, depois de acrescentadas todas as extensões.

### Paragem de emergência - $F_{M3}$

A paragem de emergência é uma acção que resulta da intervenção do operador. As máquinas *CNC* estão equipadas com um botão de pressão que, quando premido, desliga completamente todas as funções das mesmas.

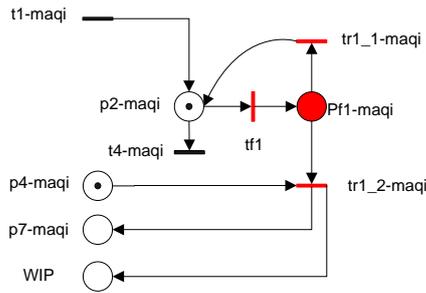


Figura 5.5: Extensão à RdP correspondente ao tratamento da falha  $F_{M1}$

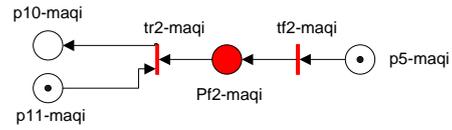


Figura 5.6: Extensão à RdP correspondente ao tratamento da falha  $F_{M2}$

Como se viu na tabela 5.2, a recuperação desta falha também só poderá efectuar-se por intervenção do operador, reiniciando a máquina.

A recuperação desta falha é, sob o ponto de vista da modelação, em tudo idêntica a  $F_{M2}$ . Por isso, é possível usar o lugar e a transição de recuperação de  $F_{M3}$ , colocando  $t_{f3-maqi}$  em concorrência com  $t_{f2-maqi}$ . Daí resulta a subrede da figura 5.7.

Observa-se na figura 5.9 que a ocorrência de qualquer das falhas  $F_{M2}$  ou  $F_{M3}$  origina a reinicialização do circuito da máquina, correspondendo à reinicialização efectuada pelo operador. De notar que, na altura em que elas ocorrem, não há mais nenhum módulo da RdP em interacção com o módulo da máquina. Por esta razão, é suficiente garantir a atingibilidade da marcação inicial do módulo máquina ( $\mu(P_{10-maq}) = 1$ ), demonstrada no grafo.

**Alarme de segurança -  $F_{M4}$**

Um alarme de segurança pode ser encarado como uma falha no sistema. Admitindo que as barreiras de protecção das máquina devem estar sempre fechadas, então, a abertura destas, pelo operador, constitui uma falha no sistema.

Trata-se de uma falha local a um lugar, uma vez que, depois de recuperada, se marca novamente o lugar a partir do qual foi detectada. A figura 5.8 mostra isso mesmo

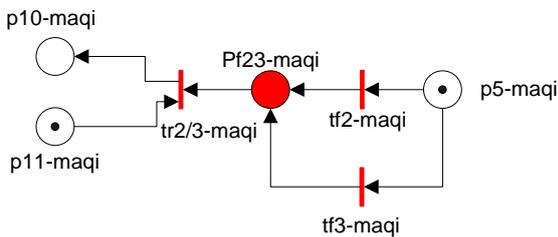


Figura 5.7: Extensão à RdP correspondente ao tratamento da falha  $F_{M3}$

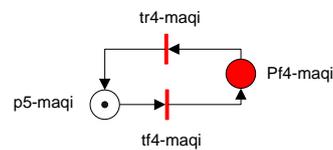


Figura 5.8: Extensão à RdP correspondente ao tratamento da falha  $F_{M4}$

No grafo da figura 5.9 observa-se a influência desta extensão na evolução da marcação da rede da máquina.

Analisando o grafo correspondente às falhas das máquinas, observa-se que todas as recuperações associadas às falhas (ramos a verde) alcançam marcações de funcionamento normal. Então, uma vez que a rede original é livre de bloqueios, também esta nova rede o será, já que, como se viu, a recuperação das falhas atinge sempre marcações válidas da rede original.

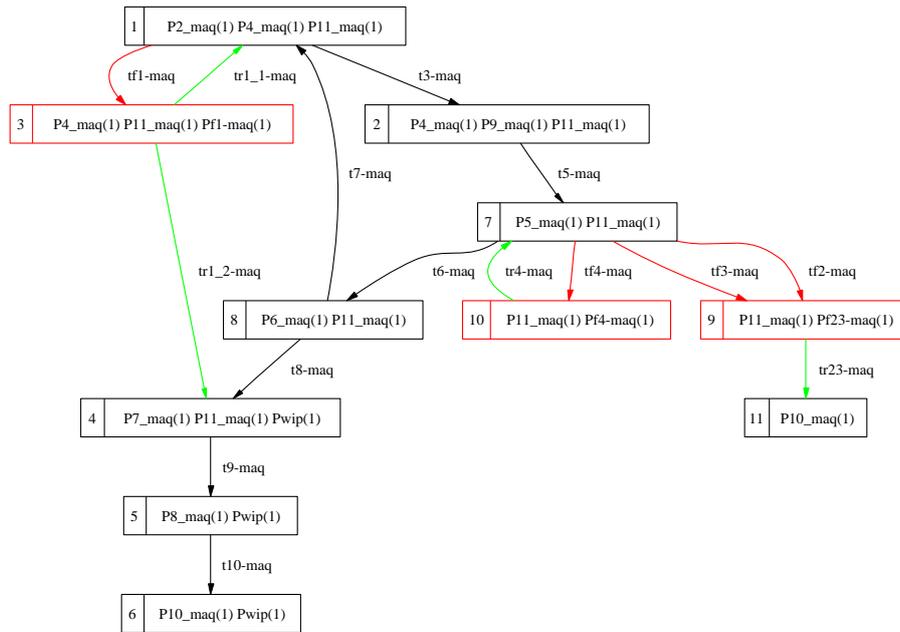


Figura 5.9: Grafo de alcançabilidade que descreve a ocorrência e recuperação de  $F_{M1}, F_{M2}, F_{M3}$  e  $F_{M4}$

## 5.2 Análise da Rdp resultante

Modeladas as falhas, é agora necessário confirmar as propriedades da nova Rdp. Depois de acrescentadas todas as extensões que tratam a ocorrência de falhas, resultou uma Rdp constituída por cento e quarenta e sete lugares (147) e cento e quarenta e quatro (144) transições.

Continua-se, neste caso, a realimentar a rede, a fim de evitar o bloqueio inerente ao fim de fabrico. Cada peça que é produzida volta ao armazém. Esta opção assegura que todas as situações possíveis de fabrico, em relação ao tamanho dos lotes, estão incluídas no modelo e na respectiva análise.

A introdução das extensões aumentou, não só o número de componentes da rede, mas também aumentou, ainda mais, a componente combinatória da rede. Note-se que, para todas as redes das máquinas, dos buffers e dos dois robots, foram acrescentadas alternativas ao fluxo normal de marcas. Isto aumentará, inevitavelmente, a complexidade da análise exaustiva da Rdp, no que diz respeito ao seu grafo de alcançabilidade.

Estes factores deixam antever uma excessiva complexidade na análise das propriedades da rede. Por isso, antes de analisar a rede, há que proceder à sua simplificação. As técnicas de redução usadas, descritas em 3.3.3, preservam as duas propriedades que interessa analisar: a vivacidade e a limitação. A aplicação destas técnicas, reduziu a rede a setenta e sete (77) lugares e sessenta e seis (66) transições. É sobre esta rede reduzida que incide a análise descrita.

Utilizou-se, também neste caso, o analisador *INA*, descrito em [15]. Todos os resultados, de seguida apresentados, foram obtidos com o auxílio deste analisador.

### 5.2.1 Análise dos Invariantes da Rede

Tal como em 5, antecedeu-se a análise exaustiva da Rdp, pela análise algébrica, na tentativa de auferir, desde já, algumas propriedades da Rdp.

A análise dos invariantes de marcação e de disparo será efectuada à luz do apresentado em 3.3.1.

### Condição Necessária para a Vivacidade

Esta condição (3.2) continua a verificar-se para todos os p-invariantes não negativos da rede.

### Condição Suficiente para a Limitação

A nova RdP é, também, coberta por p-invariantes, ou seja, todos os lugares da rede são limitados. Tal como na RdP original, conclui-se que esta condição (3.2) se verifica e que, como tal, a **RdP é limitada**.

### Condição Necessária para a Vivacidade e Limitação

Também esta condição (3.4) é satisfeita, uma vez que toda a rede é coberta por *t-invariantes*.

Com base nas condições atrás expostas, infere-se que a rede é limitada e que a vivacidade da mesma não é garantida. Não se conseguindo, desta forma, garantir a vivacidade da rede, resta proceder à análise exhaustiva da mesma. Esta análise, efectuada com o cálculo do grafo de alcançabilidade da rede, fornecerá todas as marcações alcançáveis a partir do estado inicial da mesma e a informação acerca da vivacidade das mesmas,

### 5.2.2 Grafo de Alcançabilidade

Na elaboração do grafo de alcançabilidade desta RdP, a "explosão" de estados ainda foi mais notória do que para a RdP original. Usaram-se as mesmas reduções que em 4.3.2 para minimizar o número de estados durante a construção do grafo de alcançabilidade.

Na tabela 5.3, encontram-se compilados os resultados da análise à rede, para diferentes ordens de produção e utilizando, usando sempre o método de redução *Stubborn Sets*. A figura 5.10 serve para mostrar o aumento da complexidade do grafo com o tamanho dos lotes.

As análises foram efectuadas num Computador Intel Pentium IV a 2400Mhz, com 256 Megabyte de memória RAM.

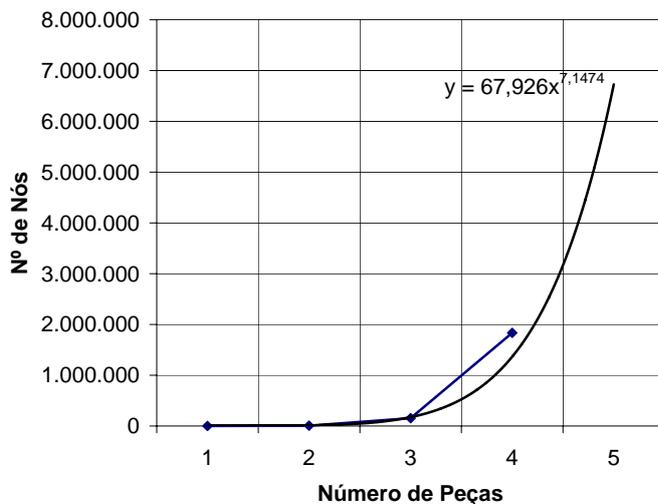


Tabela 5.3: Resultados da análise efectuada à RdP

Peças	$R_{stubb}$
1	85
2	6.468
3	154.676
4	1.834.268
5	> 5.699.698

Figura 5.10: Relação Nº de peças - Nós do Grafo de alcançabilidade

Utilizando a ferramenta acima mencionada, calculou-se o grafo de alcançabilidade da respectiva rede e, até lotes de quatro (4) peças, não foi detectado qualquer nó terminal. Isto equivale a dizer que, a partir da marcação inicial, a rede não alcança qualquer marcação morta. Daqui se conclui que, até lotes de produção de quatro peças, a RdP não apresenta, a partir da marcação inicial, qualquer bloqueio (*deadlock*).

Como se previa, o substancial aumento da complexidade da rede, tornou o cálculo do grafo de alcançabilidade, para lotes de mais de quatro peças, computacionalmente incontrolável, pelo que, para lotes maiores, nada se poderá concluir acerca da vivacidade da rede.



## CAPÍTULO 6

# SUPERVISÃO DA *CFF* - IMPLEMENTAÇÃO PRÁTICA

Neste capítulo, apresentam-se as alterações efectuadas à *CFF*, no sentido de implementar a nova arquitectura, apresentada em 2.3. Na figura 6.1 está descrito o diagrama funcional desta implementação.

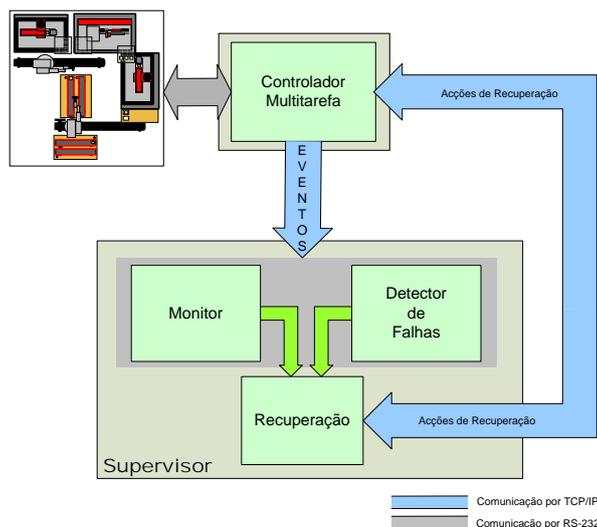


Figura 6.1: Fluxo de informação entre o Controlador multitarefa e o Supervisor remoto

Descrevem-se as potencialidades do *Posto de Design e Produção Remota* e a forma como se procede, remotamente, à *detecção e recuperação* de falhas da *CFF*, com base na modelação e análise efectuada nos capítulos 4 e 5.

Todas as aplicações foram desenvolvidas em *Visual C++ 6* e correm com o sistema operativo *Windows 2000* da *Microsoft*.

### 6.1 Arquitectura de Comunicações

A figura 6.2 apresenta a nova arquitectura de comunicações da *CFF*.

#### Ligações *RS232*

A arquitectura geral de funcionamento, da *CFF*, assenta na troca de mensagens entre as *threads* da aplicação de controlo do PC e os processos dos controladores dedicados a cada um dos componentes. Cada *thread* de controlo é responsável pela gestão do recurso que modela. A comunicação, com cada recurso, é efectuada por intermédio de uma porta *RS232* dedicada, que comunica com o respectivo processo do controlador. Destas comunicações, tal como se verifica na figura 6.2, resulta um total de nove ligações *RS232* entre o PC de controlo e os controladores. As três ligações restantes, entre o PC e as máquinas ferramenta, servem única e exclusivamente para o carregamento dos programas CNC.

Neste trabalho, afim de satisfazer os objectivos já referidos, acrescentaram-se duas ligações *RS232*. Isto permitiu a comunicação entre aplicação de controlo e o *sistema operativo* dos controladores, através das *portas de controlo*. Esta ligação permite configurar o controlador, programar rotinas de controlo, efectuar todos os

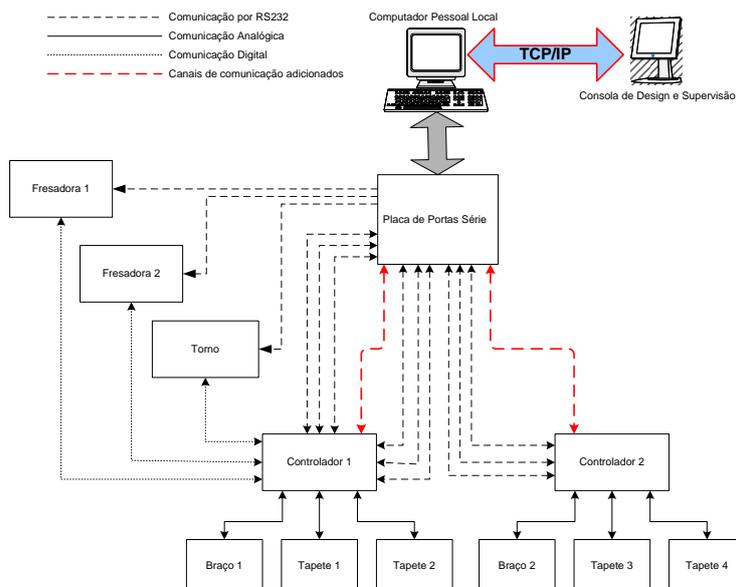


Figura 6.2: Arquitectura de comunicações da CFF

comandos que o teclado especializado disponibiliza, executar ou abortar programas e obter mensagens de falha dos recursos controlados. Esta porta de comunicações, para além de fundamental na detecção de falhas, possui a capacidade de reinicialização dos recursos ligados ao controlador em causa.

A utilização das Portas de Controlo originou a criação de duas *threads* dedicadas a cada uma delas, na aplicação de controlo e gestão da CFF.

Aproveitando-se as potencialidades destas portas, desenvolveu-se o diálogo, da figura 6.3, com a função iniciar ou reinicializar os braços robots e/ou a CFF. Assim, a aplicação fica dotada de uma maior autonomia e os teclados especializados deixam de ser necessários para o desempenho destas funções. Na tabela C.4, apresentam-se as especificações que permitem o correcto funcionamento das portas RS232 dos controladores.

### Ligações Digitais

Os controladores, tal como referido em 2.1.2, possuem dezasseis entradas e saídas digitais. Os processos dedicados às máquinas ferramenta recebem mensagens do PC de controlo e, ou activam as saídas digitais para efectuarem acções pretendidas, ou verificam ciclicamente as entradas digitais para poderem reportar alterações do estado das máquinas.

### Ligações TCP/IP

Para a implementação dos dois módulos remotos (*Design e Supervisão*), foi necessário recorrer a uma ligação *ethernet*, por TCP/IP. É a partir desta que se processa toda a troca de informação entre o Posto de *Design e Supervisão* e o PC de Controlo, da CFF. O facto de apenas dois PCs, de controlo e de supervisão, estarem ligados na mesma rede *ethernet*, exclui a hipótese de colisões e reduz de, forma significativa, a hipótese de se perderem mensagens ou de se alterar a ordem de recepção/envio das mesmas (*latência*), o que se possibilita a implementação da supervisão *on-line*. Não foram, no entanto, tidos em conta erros de comunicação.

## 6.2 Posto Remoto de *Design e Produção*

Um dos objectivos propostos, era o de conferir à CFF a capacidade de produzir peças que pudessem ser desenhadas com a ferramenta CAD/CAM descrita no apêndice D. Para tal foi desenvolvida (*Visual C++ 6*) uma aplicação remota que comunica por TCP/IP com a CFF, a fim de enviar novos tipos de peças desenhados pela ferramenta CAD e de receber modelos já em fabrico, permitindo a alteração dos mesmos. O natural

distanciamento físico entre os postos de design e de fabrico, muitas vezes até em diferentes países, conferem grande importância a esta opção. Além disto, as limitações do PC que controla a *CFF* tornou imperativa a utilização do TCP/IP para ligação a um PC, suficientemente avançado, que permitisse a utilização do *MCam9*. Assim, como de resto acontece no mercado profissional, munui-se a *CFF* de um Posto de Design remoto.

Esta mesma aplicação servirá, também, de base para a supervisão e monitorização, pontos fundamentais deste trabalho. Na figura 6.6, pode ver-se o aspecto da caixa de diálogo desta aplicação.

Depois de desenhada a peça e obtidos os ficheiros correspondentes a cada operação CNC, o utilizador, acedendo remotamente à base de dados da *CFF*, poderá efectuar 4 operações diferentes::

- Alterar/editar modelos existentes.
- Criar novo modelo.
- Apagar modelos existentes.
- Definir e/ou iniciar a Ordem de Produção.

### Criação/Alteração de Modelos de Fabrico

A figura 6.4 mostra a janela de diálogo correspondente à criação de um novo modelo.

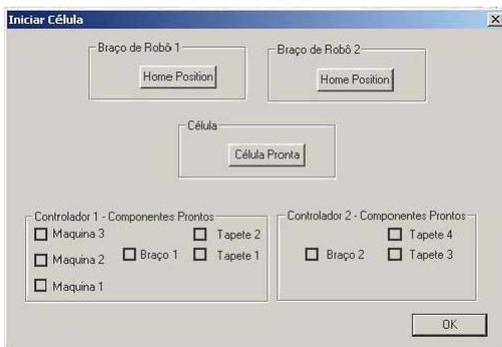


Figura 6.4: Janela da aplicação *Novo Modelo*

Figura 6.3: Diálogo de inicialização dos robots e/ou da *CFF*

À criação e alteração de modelos está associada a introdução dos seguintes dados:

- Operações a efectuar, com a informação de qual a máquina e ficheiro CNC associados.
- Armazém onde se encontra a matéria prima.
- Nome do novo modelo.

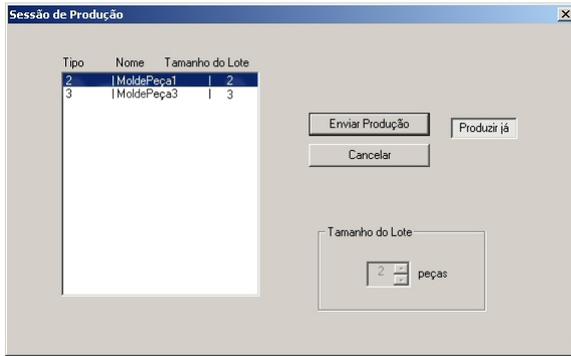
Estes dados serão então enviados para a *CFF* e poderá assim ser produzido um novo modelo de peça.

A aplicação remota envia à *CFF* todos os ficheiros CNC necessários à produção do modelo, assim como recebe da mesma todos os ficheiros de modelos já em produção. Trata-se de uma aplicação simples, mas que confere à *CFF* uma enorme versatilidade e actualidade.

### Ordem de Produção

A Ordem de Produção define um ciclo de fabrico da *CFF*. No início de cada um destes ciclos, terão que ser definidos os tipos de peças a fabricar e as quantidades de cada um deles. Esta operação poderá agora ser efectuada remotamente, podendo, também, ser dada a ordem de início da mesma. A figura 6.5 mostra o aspecto geral da caixa de diálogo da ordem de produção remota.

A ordem remota de início de produção inicia automaticamente a monitorização e supervisão da *CFF*. Por razões óbvias de segurança, a produção só começará com a confirmação do operador.

Figura 6.5: Janela da aplicação *Sessão de Produção*Figura 6.6: Janela da aplicação *Posto de Design*

### 6.3 Monitorização, Detecção e Recuperação de Falhas

Na figura, 6.1, está representado o fluxo de informação entre o PC de controlo, ou controlador multitarefa, e a aplicação remota que monitoriza e supervisiona as acções efectuadas na CFF. A supervisão da CFF divide-se entre a aplicação local de controlo e a aplicação remota. Foi adicionado, à aplicação local, um processo responsável pela comunicação com o supervisor. Este processo, o *supervisor local*, recebe acções do supervisor e envia-as para os componentes da CFF, comportando-se, por isso, como um *interface* entre o supervisor e a célula. Os eventos que permitem a monitorização são enviados por um *socket TCP/IP* e, por outro *socket*, são recebidas as acções de recuperação. O *supervisor local* possui acesso a todas as estruturas de dados e *threads* de controlo. Pode, também, comunicar directamente com os controladores dos robots e, por isso, permite ao supervisor controlar directamente cada um dos componentes da CFF.

#### 6.3.1 Monitor

Na aplicação cliente, foi criado um *monitor*, que recebe eventos da aplicação de controlo e gestão, da CFF. Esses eventos foram associados e mapeados à imagem da modelação efectuada aos diversos componentes no capítulo 4, incluindo a extensões que correspondem ao tratamento das falhas. Na tabela C.1 do apêndice C, são apresentados os eventos recebidos pelo monitor.

A recepção de todos os eventos associados às RdP modeladas, permite construir um observador baseado na rede modelada que servirá, não só para informar o utilizador acerca do estado do sistema (através de um interface gráfico), como para detectar falhas e recuperar a CFF.

Após a recepção de cada evento, actualiza-se a marcação/estado do componente a que se refere. A mudança de estado relativa à ocorrência de falhas será solicitada pela *recuperação*, após terem sido detectadas pelo *detector*. Recorrendo às equações de estado 3.5, obtêm-se, para cada componente  $i$  da CFF, as seguintes equações:

$$\begin{aligned} \mu(0)_i &= \mu_{0_i}, \\ \mu(k+1)_i &= \mu(k)_i + D_i q(k)_i \quad D_i \in \mathbb{Z}^{n_i \times m_i}, \mu \in \mathbb{Z}^{n_i}, q \in \mathbb{Z}^{m_i}, (\mu_i, q_i \geq 0). \end{aligned} \quad (6.1)$$

Sendo a marcação inicial  $\mu_{0_i}$  conhecida no início da sessão de fabrico, a evolução da marcação de cada rede vai ser dada pela recepção, sequencial, de cada evento. O vector de disparo  $q_i$  possui apenas um elemento não negativo, igual à unidade, na posição relativa ao evento recebido. A matriz de incidência  $D_i$  é a matriz de incidência relativa a cada componente. Na figura 6.7 apresenta-se o exemplo da forma como o *monitor* modela RdP dos componentes da CFF. O facto de não ocorrerem falhas devido a erros de funcionamento conjunto entre um ou mais componentes, torna dispensável a observação dos lugares de controlo.

#### 6.3.2 Detector de Falhas

Como se viu, existem duas formas de ocorrência de falhas: uma baseada na ocorrência de eventos de falha e outra baseada em *timeouts* após a ocorrência dos mesmos. A primeira não requer, à partida, um algoritmo

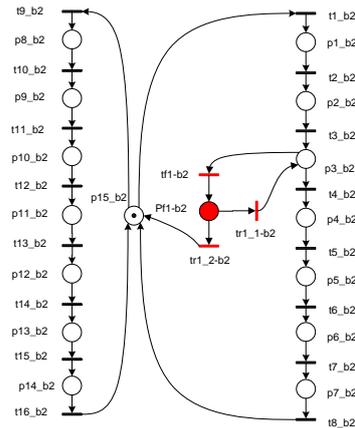


Figura 6.7: RdP do robot *B2* modelada no *Monitor*

especialmente eficaz. Bastaria, se fosse esta a única forma de ocorrência de falhas, uma lista de eventos de falha e, se ocorressem, assinalá-los. No entanto, havendo também que detectar falhas pela não ocorrência de determinados eventos, será necessário recorrer a outro tipo de algoritmo.

Em [4] apresenta-se um algoritmo para detecção de falhas, através da detecção de sequências de eventos incorrectas e em que não existem eventos directamente associados a falhas. Este método destina-se a *SEDs* temporizados, sob a forma de *Automata* ou RdPs temporizados, em que a ocorrência de eventos está bem definida e ordenada no tempo. A cada evento, associa-se uma *expectativa*, que não é mais do que o a sequência de eventos seguintes que garantem o funcionamento correcto do sistema. Quando a *expectativa*, o algoritmo de detecção declara a falha.

Ainda que este método não tenha sido desenvolvido para o tipo de RdP usado na modelação da *CFF*, é possível basear, no mesmo, a construção de um algoritmo que se adapte às características da *CFF* e do seu modelo RdP. Assim, obteve-se um algoritmo que não só detecta falhas directamente ligadas à ocorrência de eventos, como permite a detecção de sequências de eventos que indiciam a ocorrência de falhas.

A cada falha associa-se uma *expectativa* definida por:  $x_f = (e_i, t_i, e_f, t_{out})$  em que:  $e_i$  é o evento a seguir ao qual poderá ocorrer a falha;  $t_i$  é tempo que decorre depois de  $e_i$  ter acontecido;  $e_f$  é o evento que deverá ocorrer depois de  $e_i$ ;  $t_{out}$  é o tempo em que  $e_f$  terá que ocorrer para que não seja declarada a falha (*timeout*). Seja  $O_f$  o conjunto de todas as *expectativas*  $x_f$  e  $O_v$  o conjunto de todas as *expectativas* em verificação, isto é, *expectativas* em que já ocorreram os respectivos eventos  $e_i$  e se aguarda a ocorrência de  $e_f$ . Para falhas, cujo acontecimento está directamente ligado à ocorrência de eventos, apenas  $e_i$  é não nulo e representa o evento associado à falha. O diagrama da figura 6.8 ilustra o conceito de *expectativa*.

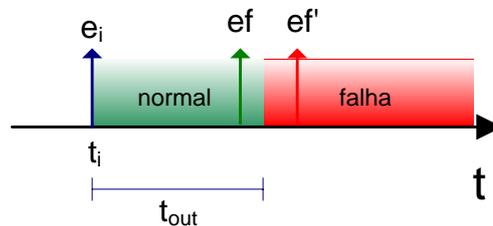


Figura 6.8: Diagrama temporal do método de detecção de falhas associadas à ocorrência de *timeouts*

Com base no conceito de *expectativas*, foi possível a criação de um algoritmo constituído por dois principais procedimentos, apresentados na figura 6.9<sup>1</sup>.

<sup>1</sup>a notação utilizada baseia-se na teoria dos conjuntos

---

<pre> <b>PROCEDIMENTO</b> Novo_Evento(<math>e_n</math>) {   <b>Se</b> existir <math>x_v \in O_v</math> tal que <math>e_n = x_v(e_f)</math> <b>então</b>   {     <math>O_v = O_v \setminus \{x_v\}</math>     <b>Declarar</b> Fim_Expectativa   }   <b>Se</b> existir <math>x_f \in O_f</math> tal que <math>e_n = x_f(e_i)</math> <b>então</b>   {     <math>x_n = x_f</math>     <math>O_v = O_v \cup \{x_n\}</math>     <b>Declarar</b> Nova_Expectativa   } } </pre>	<pre> <b>PROCEDIMENTO</b> Actualiza_Expectativas(<math>t_s</math>) {   <b>Para cada</b> <math>x_i \in O_v</math>   {     <math>x_i(t) = x_i(t) + t_s</math>     <b>Se</b> <math>x_i(t) &gt; x_i(t_{out})</math> <b>então</b>     {       <math>O_v = O_v \setminus \{x_i\}</math>       <b>Declarar</b> FALHA <math>x_i</math>     }   } } </pre>
---	---

---

Figura 6.9: Procedimentos do Detector de Falhas

O procedimento *Novo\_evento* trata cada um dos eventos que são recebidos, verificando se estão envolvidos na ocorrência de alguma falha. Se o evento recebido estiver referenciado na lista de *expectativas*  $O_f$ , como antecessor de uma possível falha, é colocada a respectiva *expectativa* na lista das *expectativas* em verificação  $O_v$ . Se, por outro lado, estiver referenciado como evento final de uma *expectativa* em verificação, esta será retirada da respectiva lista, sinal de que a falha não ocorreu.

O segundo procedimento, *Actualiza\_Expectativas*, é responsável pela actualização dos tempos associados às *expectativas* em verificação. Depois desta actualização, detecta eventuais *timeouts* ( $x_i(t_{out}) \leq 0$ ), declarando, nesse caso, a ocorrência da falha associada à respectiva *expectativa*.

Cada um dos procedimentos possui uma *thread* dedicada, sendo a devida sincronização efectuada por intermédio de eventos. Uma vez que partilham as mesmas estruturas de dados, é necessário assegurar a exclusão mútua no acesso às mesmas. Como é habitual, em problemas deste tipo, assegurou-se esta exclusão mútua com o recurso a *mutexs* (*mutual exclusion objects*), objectos de programação que proíbem o acesso simultâneo de várias *threads* a um mesmo recurso.

Apresentados os métodos de monitorização e detecção de falhas da CFF, demonstra-se, de seguida, a forma de conjugar os dois para, que se saiba se existe um plano de recuperação da CFF, isto é, se a falha ocorrida é uma das falhas para a qual foi modelada a sua recuperação.

### 6.3.3 Recuperação

Quando surge um evento de falha, é necessário recuperar adequadamente o sistema, para que este retome o funcionamento normal. Para tal, foi criado um ficheiro de falhas (*falhas.txt*), que contém as acções e comandos de recuperação de falhas a enviar ao PC de controlo. Desta forma, a seguir à detecção de uma falha, verifica-se, nesse ficheiro, se existe uma recuperação associada a essa falha e ao estado actual da CFF (através da marcação das RdP do *monitor*). No caso de não existir tal recuperação, é declarada *falha fatal* e ordena-se a reinicialização da CFF. No apêndice C, encontra-se a tabela C.2 que contém todas as acções e comandos passíveis de serem introduzidas no ficheiro *falhas.txt* e que serão enviadas para recuperação das falhas. Para modificar ou acrescentar a recuperação de uma determinada falha, bastará editar este ficheiro.

O processo de recuperação de uma falha consiste no envio de acções e comandos do *supervisor* para o PC de controlo, através de um *socket* de recuperação. As acções de recuperação enviadas para a CFF, baseiam-se na modelação efectuada no capítulo 5. A forma como está implementada a recuperação não permite o seguimento exacto da modelação efectuada. Esse seguimento só seria garantido com o desenvolvimento de um motor de RdP que emulasse o comportamento da RdP e associasse, ao disparo de transições, acções de comando.

No entanto, é possível garantir algum paralelismo entre as acções de recuperação enviadas para a CFF e o fluxo de marcas imposto à RdP da mesma pelas transições de falha e recuperação. Como é óbvio, outras acções,

como a actualização de variáveis associadas ao sistema de controlo, não foram tidas em conta na modelação, mas terão que ser efectuadas a fim de assegurar a correcta recuperação da CFF.

A recuperação segue os seguintes passos:

1. É indicado ao *monitor* o disparo das transições de falha, para que os componentes envolvidos entrem em estado de falha.
2. São enviadas para o *PC de controlo*, à imagem do que foi modelado, as acções que recuperam o sistema.
3. É indicado ao *monitor* o disparo das transições de recuperação, para actualização das redes de monitorização.

**Exemplo 6.1 (Detecção e Recuperação da falha  $F_{B2}$  )**

Apresenta-se, agora, um exemplo de recuperação implementado. Pretende-se, com isto, destacar o paralelismo entre as acções de recuperação e as extensões RdP que as modelam. Na figura 6.10 está representada a parte da RdP na qual interfere a detecção e recuperação da falha  $F_{B2}$ . A cada transição das extensões que modelam a detecção e recuperação da falha associaram-se números que representam a ordem pela qual são enviadas as respectivas acções de recuperação para a CFF. Demonstra-se apenas o caso em que o destino da peça seria uma das máquinas, neste caso a máquina M3. Na tabela 6.1 apresentam-se os eventos observados pelo *monitor*, numa sessão de fabrico em que o *supervisor* recuperou a CFF após detectar a ocorrência da falha  $F_{B2}$ . A tabela permite, também, fazer a correspondência entre disparos das transições da RdP da figura 6.10 e as acções do supervisor. A sessão de supervisão completa apresenta-se na tabela C.3. ■

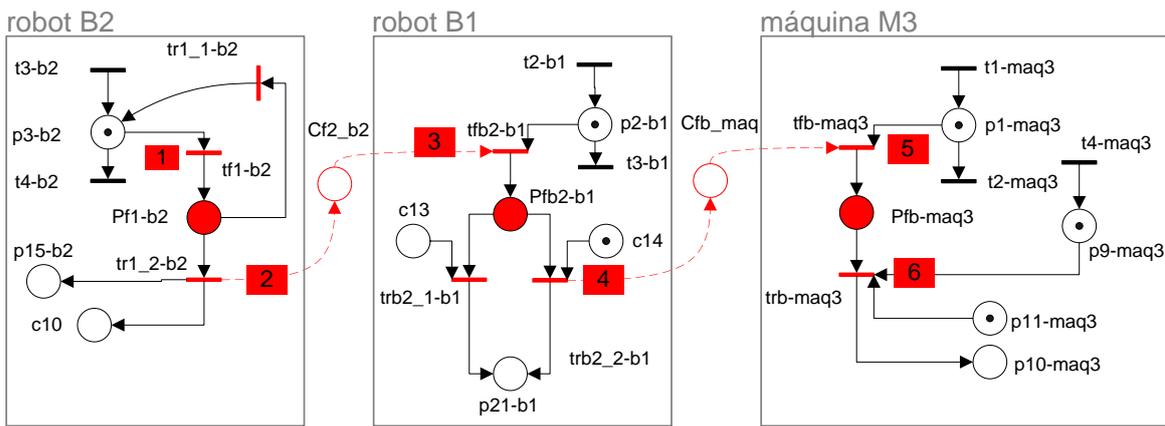


Figura 6.10: Sequência de recuperação de  $F_{B2}$

Tabela 6.1: Resultados da detecção e recuperação da falha  $F_{B2}$

Recuperação de B2	ordem	Recuperação de B1	ordem	Recuperação de M3	ordem
Ocorreu falha no braço 2 - 00:03	1	Reinicia Thread - 02:18	3 e 4	Reinicia Thread - 02:19	5 e 6
Termina Thread - 00:03					
Recuperação					
C2 Abortou Programas - 00:03					
C2 Controlo Activo - 00:04					
C2 Iniciou Programa de B2 - 00:04					
C2 Abortou Programas - 00:17					
C2 Home Position de B2 - 00:18					
C2 Iniciou Programas - 02:14					
C2 Thread Retomada - 02:18					
Peça Retira Peça de Produção - 02:18					
Recuperado					
Inicia Thread - 19: 2:18	2				

## 6.4 Dificuldades de implementação

Durante o estudo e desenvolvimento do software, que permitiu a implementação da componente prática do trabalho, surgiram variados problemas relacionados com avarias de hardware e debilidade do software de controlo, de entre os quais se destacam os seguintes:

- Engrenagem da pinça eléctrica desgastada
- Mau funcionamento no aperto do torno
- Desprogramação das memórias EPROM do controlador 2 devido a avaria da fonte de alimentação.
- Mau funcionamento da base linear da máquina ferramenta número um
- Ponte de díodos, do circuito de potência, queimada (máquina ferramenta número um)
- Monitor da máquina ferramenta número três avariado.
- Placa de controlo dos motores passo a passo da máquina ferramenta número três avariada
- Placa BocaBoard apresenta um mau funcionamento para metade das 16 portas, conseguindo-se pôr em funcionamento quatro delas, recorrendo a especificações diferentes das fornecidas pelo fabricante.
- Problemas na leitura das portas série, devido a deficiências nas rotinas de comunicação da aplicação de controlo original.

Estes imponderáveis, sempre presentes quando se tem o desejo de por a funcionar uma metodologia num sistema real, atrasaram, de facto, a realização de todos os objectivos iniciais do trabalho.

## CAPÍTULO 7

# CONCLUSÕES E DESENVOLVIMENTOS FUTUROS

Neste trabalho propôs-se uma arquitectura para a supervisão de sistemas de automação, com incidência na detecção e recuperação de falhas, com implementação na *CFF* do *LSCN*. Adicionalmente, integrou-se, nessa *CFF*, um sistema *CAD/CAM*, que lhe conferiu maior flexibilidade e independência.

A fim de possibilitar a implementação da arquitectura de supervisão apresentou-se um estudo exaustivo do sistema de eventos discretos [7] que modela a célula de fabricação flexível a supervisionar, bem como das propriedades estruturais e topológicas da mesma. Usou-se, para isso, a teoria das Redes de Petri (RdP) [13].

Recorrendo a uma inovadora forma de controlar RdP compostas por diversos componentes, a síntese automática de controladores, baseados no conceito de invariantes de lugar [11][9], foi possível obter uma RdP que modela o comportamento da *CFF*. Este método permitiu deixar de lado as tradicionais formas de impor restrições de controlo a *RdPs*, baseadas em métodos intuitivos e empíricos [14][1]. No capítulo 3 apresentou-se um conjunto de ferramentas autosuficientes para a modelação, controlo e análise de qualquer sistema de eventos discretos, desde que respeite os pressupostos que garantem optimalidade dos controladores obtidos a partir do referido método. Entre estes pressupostos, destaca-se a ausência de *self-loops* na rede a controlar. Mostraram-se, contudo, formas de os evitar.

A análise das propriedades da rede modelada, pelos métodos exaustivos (grafo de alcançabilidade), mostrou-se bastante complexa devido à forte componente combinatória, resultante da interacção dos diversos recursos. As soluções encontradas para este problema passaram pela utilização de métodos que reduziram a dimensão do grafo (*Stubborn Sets*) e, conseqüentemente, a complexidade da análise, reflectindo-se numa menor exigência das capacidades computacionais. Ainda assim, não foi possível garantir a ausência de bloqueios para lotes de fabrico maiores do que seis peças. Contudo, o facto de se respeitarem todas as condições da análise algébrica, leva a crer que a rede não possui bloqueios para qualquer dimensão de lote. Uma hipótese para desenvolvimento futuro, seria o uso da teoria apresentada em [2], que propõe uma análise modular das RdPs.

A arquitectura de supervisão proposta baseou-se na enumeração e classificação de uma série de falhas que podem ocorrer. A inclusão de sensores no sistema será fundamental para se detectar outro tipo de falhas. Sugere-se a inclusão de, entre outros, sensores de presença nas CNC's e tapetes, sistemas de visão que efectuem o seguimento de peças na célula, etc. Para as falhas escolhidas, foi possível apender à RdP pequenas extensões que modelam a detecção e recuperação das mesmas, sem que se alterem as suas propriedades, fundamentais, de vivacidade e limitação. A análise exaustiva, da rede resultante, revelou-se ainda mais complexa, devido ao acréscimo de componentes e ao aumento da componente combinatória dos mesmos.

A arquitectura proposta foi implementada e validada em tempo real na supervisão da *CFF*, mostrando-se que este é um passo sólido na utilização da metodologia proposta para a detecção e recuperação de falhas em sistemas de eventos discretos. Será possível, pela forma como se implementou a supervisão, alargar a supervisão para outro tipo de falhas. A detecção e recuperação de falhas é feita, em muitas abordagens, recorrendo ao uso de supervisores com capacidade de aprendizagem automática.

A inclusão de um posto de *CAD/CAM* revelou-se extremamente benéfico para a *CFF*, uma vez que lhe conferiu uma maior flexibilidade na sua capacidade de produção. Desde que a parametrização do software *CAD/CAM* esteja de acordo com os parâmetros da máquina e o pós-processador origine programas CNC que a máquina reconheça, todo o tipo de erros inerentes à má programação são, à partida, eliminados. Uma das formas de tornar esta *CFF* ainda mais automática e robusta, seria a adição de um sistema de visão, capaz de efectuar o controlo de qualidade das peças produzidas.

A *CFF* do *LSCN* possui, agora, maior flexibilidade no seu processo de fabrico. Esta flexibilidade reflecte-se na capacidade de adaptação a situações de avaria, capacidade de operar sobre diferentes produtos em simultâneo e, com a integração do sistema *CAD/CAM*, resposta rápida a alterações das características dos produtos.



# APÊNDICE A

## MODELAÇÃO DA *CFF*

Neste apêndice apresentam-se as RdPs resultantes da sincronização dos diversos componentes da *CFF*, apresentada no capítulo 4.

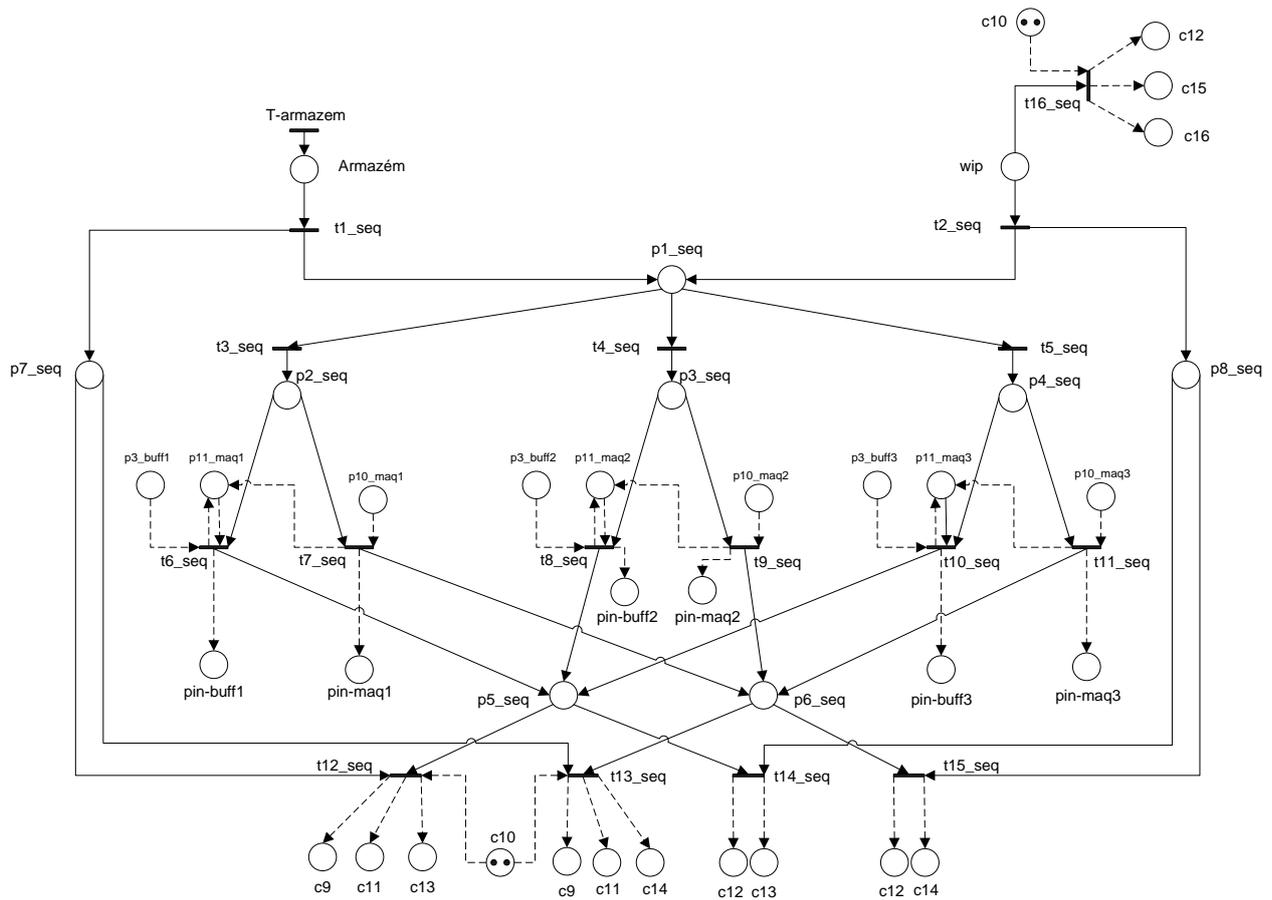


Figura A.1: RdP controlada do sequenciador

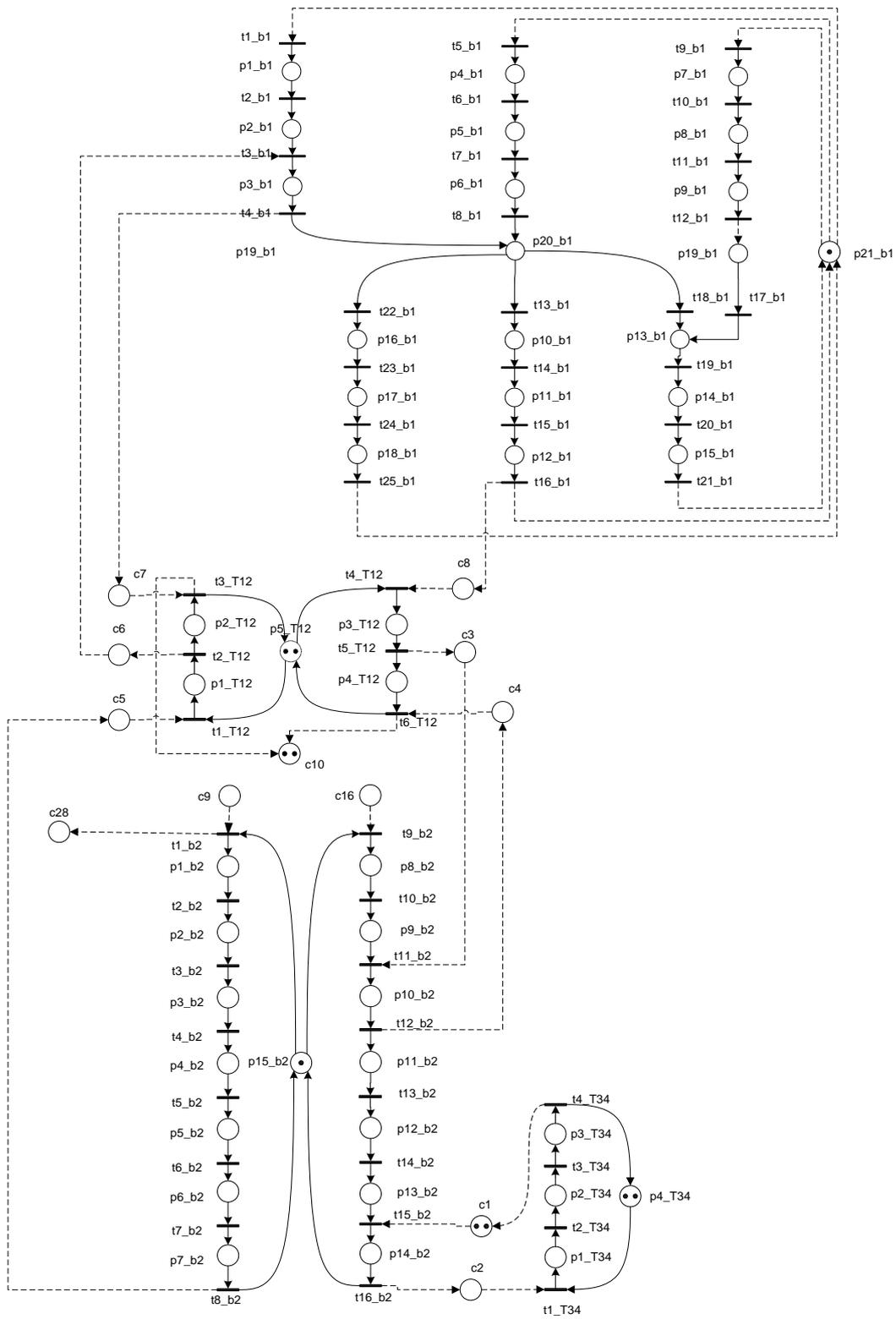


Figura A.2: RdP controlada do sistema braços-tapetes

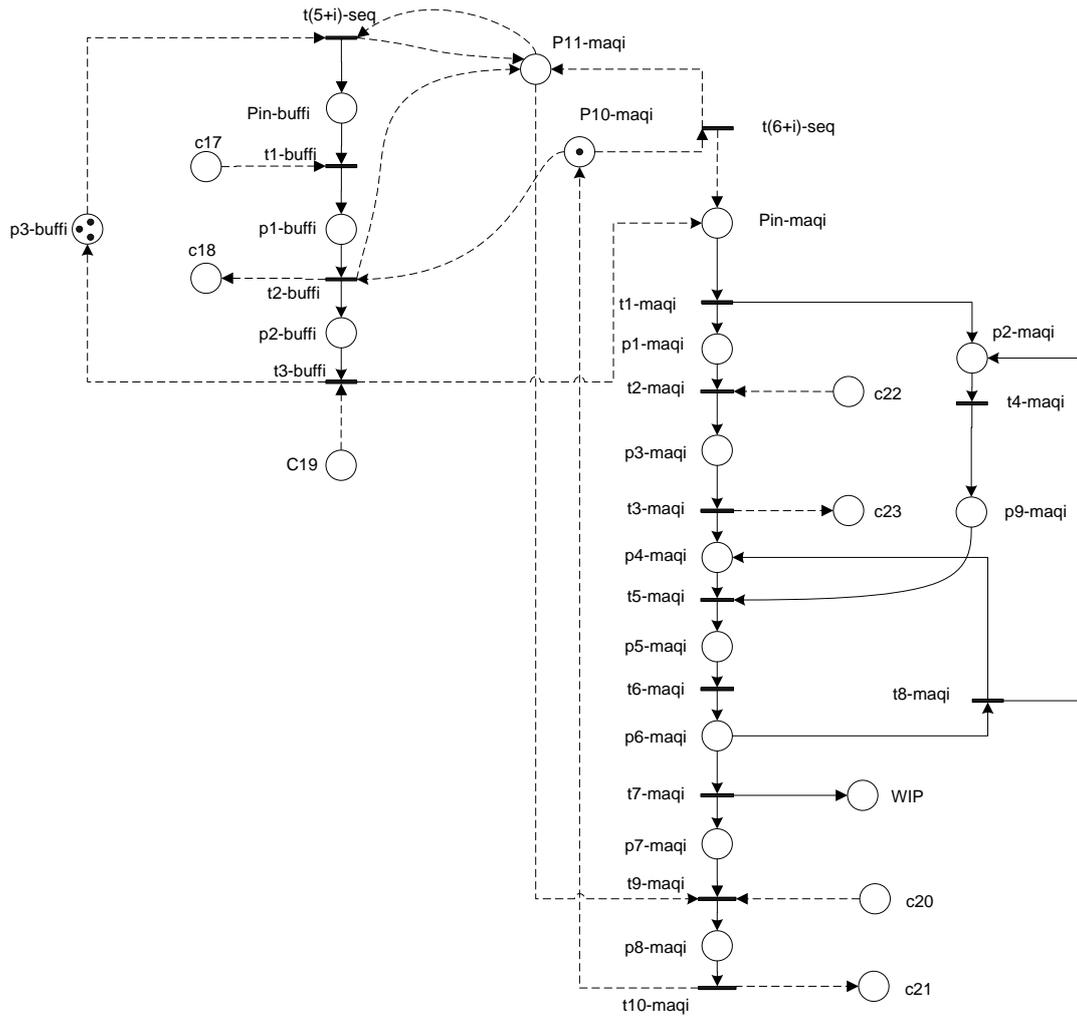


Figura A.3: RdP controlada das máquinas

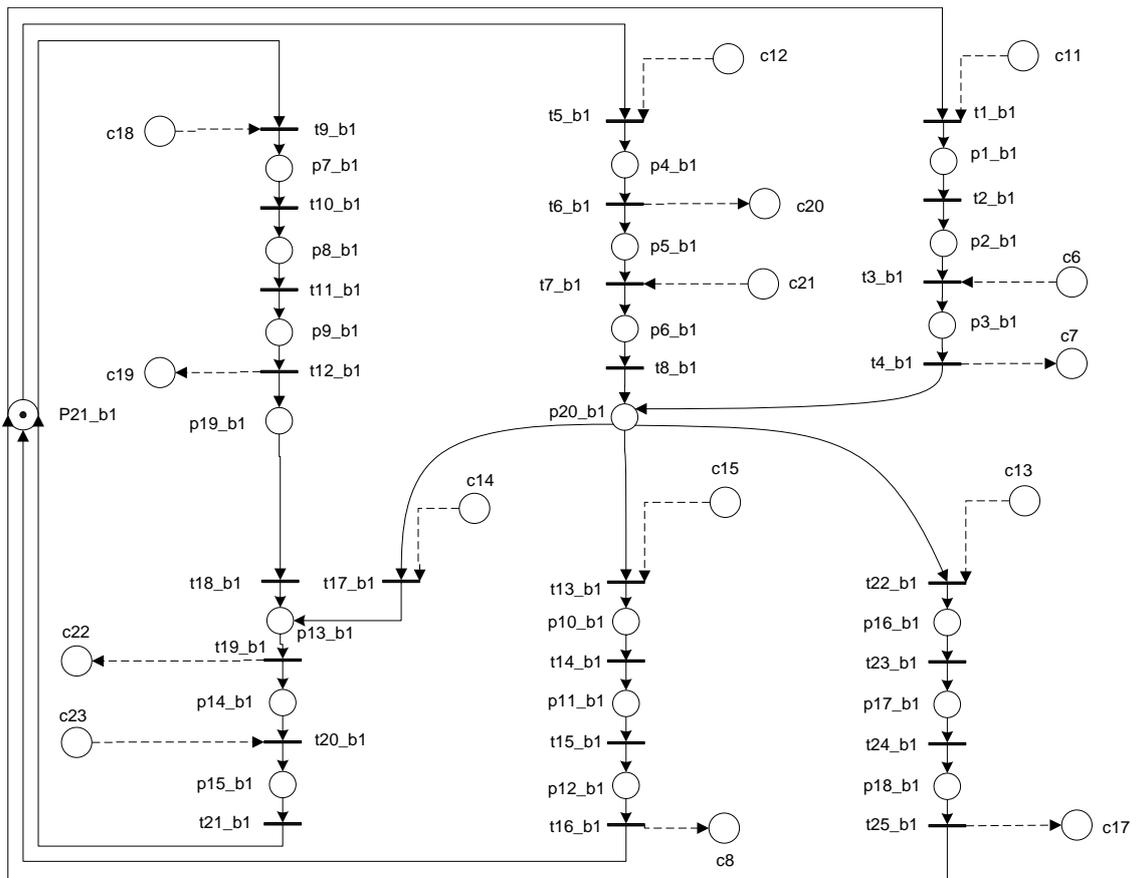


Figura A.4: RdP controlada do robot *B1*

## APÊNDICE B

# MODELAÇÃO DAS FALHAS

Para demonstrar o processo de modelação das falhas e a análise das propriedades da RdP com a adição das respectivas extensões, exemplifica-se o processo para o sistema modelado pela RdP da figura 3.4, que modela um sistema simples de transporte, constituído por uma posição de armazém, um braço robot e um tapete. A elevada complexidade da RdP da *CFP* torna imperativa a aplicação deste método de modelação a um sistema menos complexo, a fim de validar e motivar o mesmo. Uma vez que as falhas inerentes aos lugares  $p_1$ ,  $p_2$  e  $p_3$  são idênticas, no que diz respeito à ocorrência e recuperação, torna-se válida a transformação hierárquica da figura 3.8. Será, então, usada esta RdP, visto que torna significativamente mais simples a sua análise e não retira validade à mesma.

### Exemplo B.1 (Falha Local)

Como se observa na figura B.1, a ocorrência da falha está relacionada com a transição  $t_6$ , que retira do lugar  $p_1$  uma marca. Na rede original está previsto que, a seguir à marcação do lugar  $p_1$ , apenas a transição

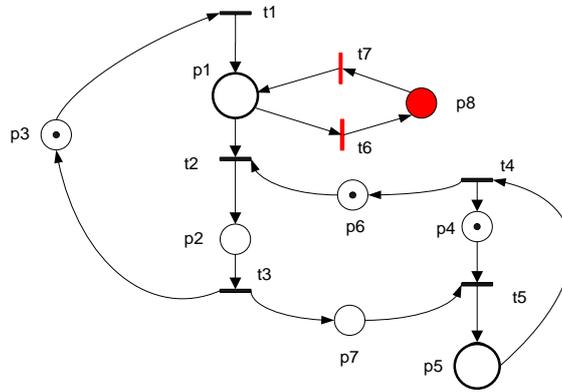


Figura B.1: Falha P-local

$t_2$  mude o estado da rede. Como é óbvio, se durante este estado ocorresse qualquer tipo de falha no braço, a rede manter-se-ia infinitamente no estado  $p_1$ , o que equivale ao bloqueio do sistema. Com a adição desta nova extensão, passa a ser possível reagir à ocorrência da falha. A transição  $t_6$  mudará o estado da rede e, depois de um processo de recuperação, a transição  $t_7$  colocará, novamente, a marca no lugar de onde havia sido retirada depois da falha.

O lugar de recuperação  $p_8$  não interage com nenhuma transição da RdP original. Por isso, a sua adição à RdP corresponderá a acrescentar uma linha de zeros à matriz de incidência original. Depois disto, falta acrescentar as colunas correspondentes às transições que se pretende acrescentar. Tem-se, então, que:

$$D_{t_6 t_7} = \begin{bmatrix} -1 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \quad D' = \begin{bmatrix} D & D_{t_6 t_7} \\ 0 & 0 \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

O que interessa provar, depois desta modelação, é que as propriedades interessantes da rede original são mantidas. Não é difícil prever que a adição deste extensão não alterará as propriedades da rede original. Trata-se de uma subrede sequencial que devolve uma marca ao lugar de onde a tirou, sem interações externas. Isto mesmo será provado na análise seguinte. Obteve-se a seguinte árvore de alcançabilidade: Todos os nós/marcações da árvore originados pelo disparo da transição  $t_6$  são estado de falha e de consequente recuperação. A **marcação**

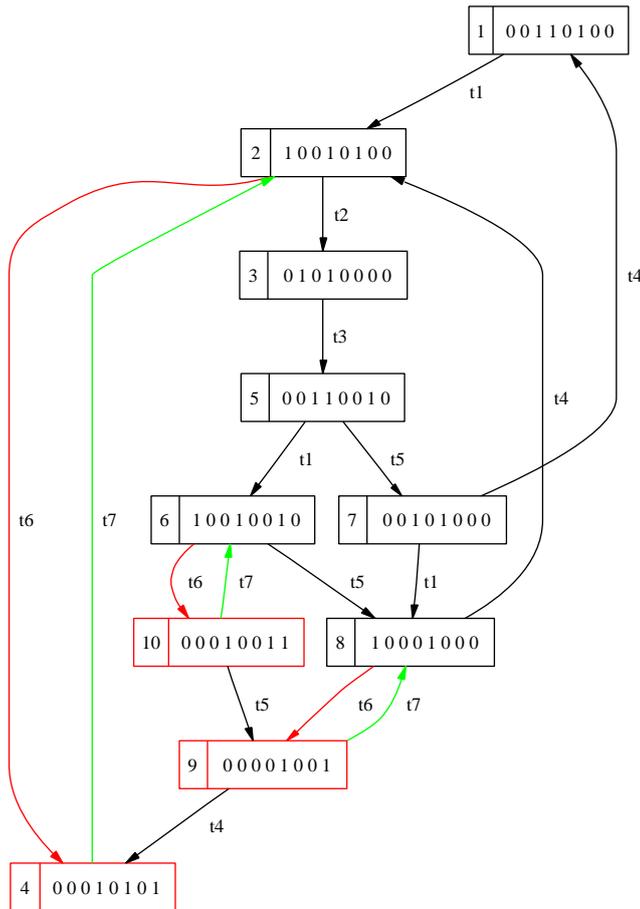


Figura B.2: Árvore de alcançabilidade da RdP da figura B.1

**inicial é atingível** a partir de todos eles, pelo que continua a ser garantido o funcionamento cíclico do sistema e a correspondente vivacidade do mesmo. Como seria de esperar, a RdP continua a ser **limitada e segura** e nenhuma das transições é morta.

A RdP possui agora mais um t-invariante:

$$y = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad x = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Facilmente se observa que este t-invariante corresponde à extensão adicionada e significa fundamentalmente que após, o disparo de  $t_6$  e  $t_7$ , a RdP volta ao mesmo estado. O t-invariante do funcionamento "normal" da RdP continua presente e prova que não há interferência da extensão adicionada. ■

### Exemplo B.2 (Falha Local a um módulo)

Na figura B.3 foi modelado um exemplo deste tipo de falhas. Neste caso, modelou-se um tipo de falha em que a recuperação do estado anterior à ocorrência da mesma é impossível. Neste exemplo o braço perde a peça durante o seu transporte e terá que voltar ao armazém para agarrar outra. Como o módulo do tapete não é afectado pela falha, diz-se que a falha é local ao módulo do braço. Por isso, o processo de recuperação apenas afecta o próprio braço e, embora seja impossível o regresso imediato ao estado em que ocorreu a falha, é possível levar o módulo à sua marcação inicial. Da mesma forma com que se obteve a matriz de incidência

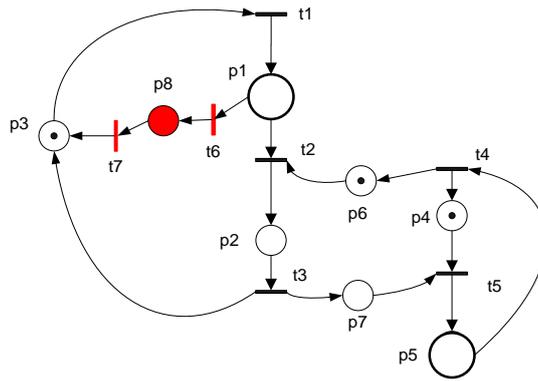


Figura B.3: Falha local a um módulo

do exemplo B.1, tem-se:

$$D_{t_6 t_7} = \begin{bmatrix} -1 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \quad D' = \begin{bmatrix} D & D_{t_6 t_7} \\ 0 & \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

A ocorrência de uma falha no braço dispara a transição  $t_6$ , esta retira a marca ao lugar  $p_1$  e, depois do processo de recuperação, é marcado o lugar  $p_3$ , conseguindo-se, assim, a desejada marcação inicial.

O grafo de alcançabilidade da figura B.4 apresenta o novo espaço de estados da RdP. É visível na árvore de alcançabilidade que a ocorrência da falha (por  $t_6$ ) leva ao reinício do ciclo de funcionamento do braço (através de  $t_7$ ). De notar, também, que a **marcação inicial**  $\mu_0$  é **atingível** a partir de todas as marcações de falha ( $\mu_8 = 1$ ) e que a rede mantém as propriedades de limitação e segurança. Todas as transições da rede se mantêm vivas.

A inclusão desta extensão introduziu na RdP um novo ciclo de funcionamento para o módulo do braço, comprovado pelo invariante  $y(1)$ .

$$y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad x = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

A RdP possui, agora, dois ciclos de funcionamento: o normal  $y(2)$  e um outro que é forçado aquando da ocorrência de uma falha. ■

### Exemplo B.3 (Falha global)

Recorrendo, mais uma vez, ao exemplo da figura 3.8, integrou-se no mesmo uma falha deste tipo. Trata-se de modelar uma falha no braço, quando este alimenta o tapete. A rede resultante está descrita na figura B.5. Neste caso, a recuperação da falha apresenta, de forma evidente, maior dificuldade de modelação. A falha ocorre numa marcação em que ambos os componentes *interagem*, pelo que a recuperação da mesma englobará, na impossibilidade de voltar ao mesmo estado, a reinicialização dos dois módulos. Ainda assim, a solução encontrada para este caso particular atinge a marcação inicial da RdP e garante as propriedades interessantes da RdP original.

A nova matriz de incidência será dada por:

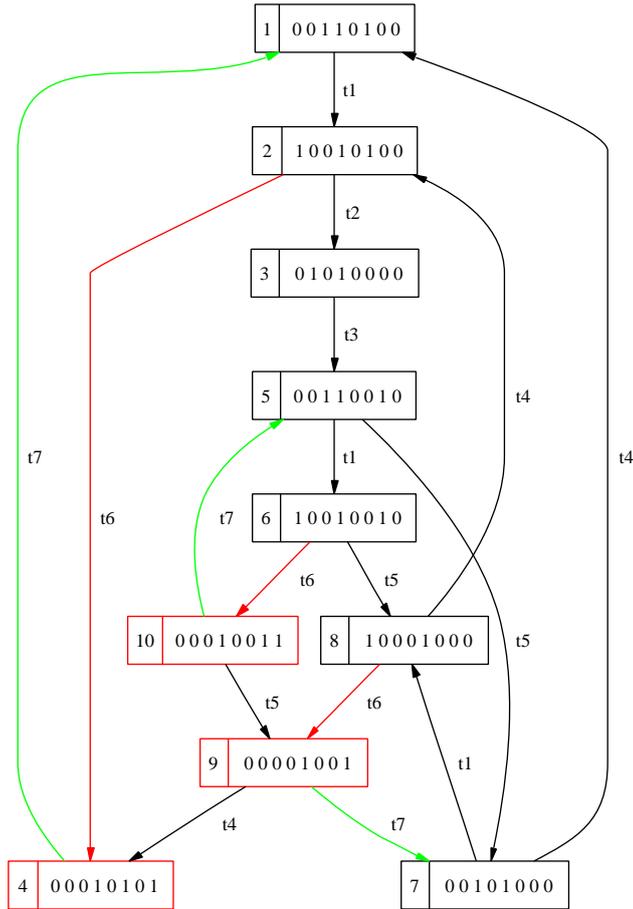


Figura B.4: Árvore de alcançabilidade da Rdp da figura B.3

$$D_{t_6 t_7} = \begin{bmatrix} 0 & 0 \\ -1 & 0 \\ 0 & 1 \\ 0 & 0 \\ 0 & 0 \\ 0 & 1 \\ 0 & 0 \\ 1 & -1 \end{bmatrix} \quad D' = \begin{bmatrix} D & D_{t_6 t_7} \\ 0 & \end{bmatrix} \quad \mu_0 = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Neste caso, a falha interfere no funcionamento do tapete. Uma vez que o tapete se encontra no início do seu ciclo, não será preciso reiniciá-lo, mas terá que ser colocada uma marca no lugar  $p_6$ , que indica que o tapete está pronto. Desta rede, resulta a árvore de alcançabilidade da figura B.6. Como se observa na árvore, existe apenas uma marcação de falha (nó 5). Observa-se, claramente, que há, depois da ocorrência da falha ( $t_6$ ), o retorno à marcação inicial da rede ( $t_7$ ), estando, por isso, provada a sua atingibilidade. Observa-se, mais uma vez, que não existem transições mortas e que a rede continua a ser limitada e segura.

A Rdp possui os seguintes invariantes:

$$y = \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 \end{bmatrix} \quad x = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix}$$

De notar o t-invariante  $y(1)$ , que representa o ciclo de funcionamento da Rdp em caso de falha. Também,

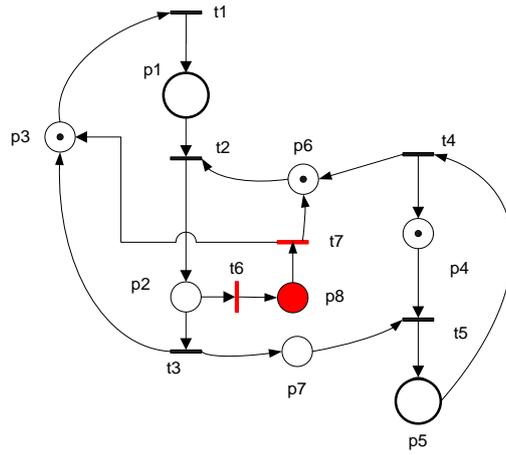


Figura B.5: Falha Global

através da análise dos invariantes, não se detecta qualquer alteração das propriedades da RdP.

Na aplicação à RdP original, este tipo de falhas revestir-se-á de grande complexidade, dado o número de módulos envolvidos. ■

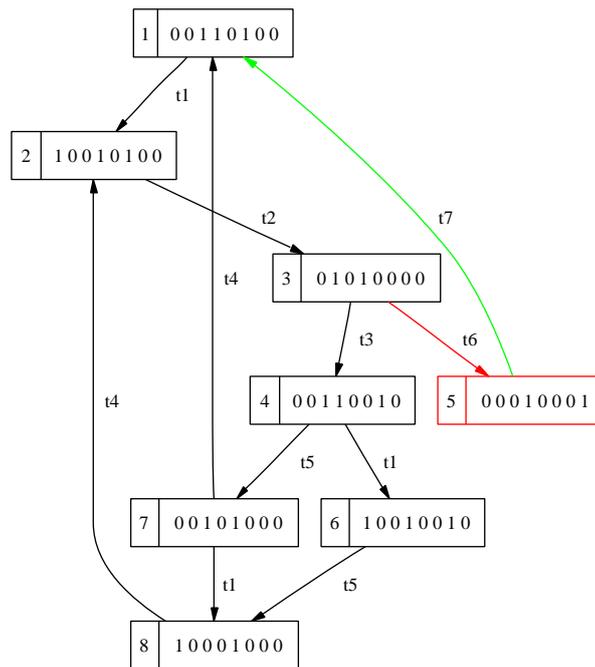
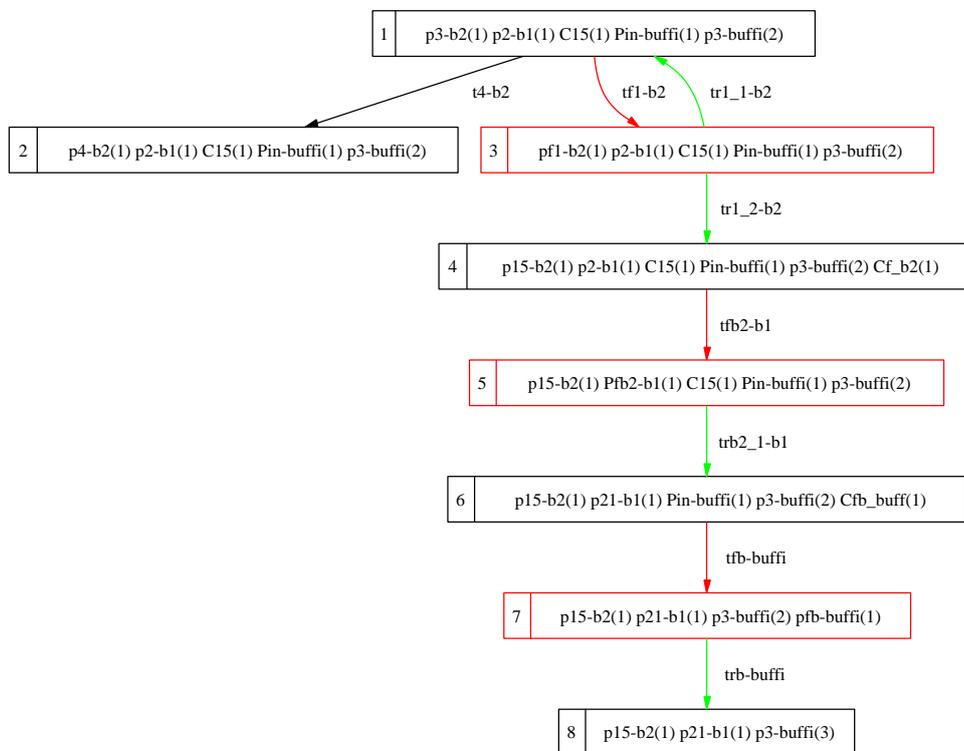


Figura B.6: Árvore de alcançabilidade da RdP da figura B.5

Figura B.7: Grafo de alcançabilidade que descreve a ocorrência e recuperação de  $F_{B2}$  quando o destino da matéria prima seria um *buffer*

# APÊNDICE C

## IMPLEMENTAÇÃO PRÁTICA

Tabela C.1: Eventos da CFF

Peca - Entidade 00	
Eventos	Código
Peca Completa	00
Peca Largada	01
Braço 1 Ocupado	02
Braço 2 Ocupado	03

Tapetes - Entidades 01 a 04	
Eventos	Código
Finaliza Execução	01
Chegou	02
Pronto	03
Finaliza Percurso	04
Inicia Percurso	05
Carregado	06
Peca Terminada	07

Braços - Entidades 05 e 06	
Eventos	Código
Fim de Percurso	01
Finaliza Execução	02

Célula - Entidade 07	
Eventos	Código
Armazém 1 Pronto	00
Armazém 2 Pronto	01
Termina Execução da célula	02
Fim de Execução	03

Maquinas - Entidades 08 a 10	
Eventos	Código
Fim de Execução da máquina	00
Programa Carregado	01
Acabou Maquinação	02
Aperto Aberto	03
Máquina Livre	04
Aperto Fechado	05
Máquina Pronta	06
Máquina Carregada	07
Máquina Disponível	08
Peça Agarrada	09

Controlador 1 - Entidade 11	
Eventos	Código
Tapete 1 Pronto	00
Tapete 2 Pronto	01
Braço 1 Pronto	02
Máquina 1 Pronta	03
Máquina 2 Pronta	04
Máquina 3 Pronta	05
Homing do Braço 1 Completo	06
Impacto do Braço 1	07

Controlador 2 - Entidade 12	
Eventos	Código
Tapete 3 Pronto	00
Tapete 4 Pronto	01
Braço 2 Pronto	02
Homing do Braço 2 Completo	03
Impacto do Braço 2	04

Tabela C.2: Acções do Supervisor

Tapetes - Entidades 01 a 04			Braços - Entidades 05 e 06		
Comando	Código	parâmetro	Comando	Código	parâmetro
ActualizaDestino	00	Destino	AbreOuFechaCom	00	0/1 (fecha/abre)
ActualizaEstado	01	Livre/Ocupado	ActualizaDestino	01	Destino
Avanca	02		ActualizaEstado	02	Livre/Ocupado
Recua	03		Agarra	03	
Termina	04		Larga	04	
Thread Suspensa	05		Termina	05	
Thread Retomada	06		Vai Para Origem	06	
			Thread Suspensa	07	
			Thread Retomada	08	
			Vai Para Destino	09	
			Reinicia Braço	10	
			Reinicia Thread	11	

Maquinas - Entidades 08 a 10			Controlador - Entidade 11 e 12		
Comandos	Código	parâmetro	Comandos	Código	parâmetro
AbreAperto	00		Abortar Programas	00	
ActualizaEstado	01	Destino	Home Position do Braço	01	
Carrega	02		Home Position da Base	02	
CarregaCodG	03		Controlo Activo	03	
FechaAperto	04		Iniciar Programa do Braço	04	
IniciaPrograma	05		Iniciar Programas	05	
PreparaCarregamento	06				
PreparaExecucao	07				
Termina	08				
TestaFimPrograma	09				
Reinicia Máquina	10				
Thread Retomada	11				
Thread Suspensa	12				
Inicia Thread	13				

Baud Rate	9600
Data bits	8
Stop bits	1
Paridade	Não
Modo	XON/XOFF

Tabela C.4: Especificações das portas RS232

Tabela C.3: Sessão de Supervisão onde se recupera a falha  $F_{B2}$ 

```

----Supervisão Iniciada às - 18:58:49

Célula      Armazém 1 Pronto -      18:59:55
Peça        Braço 2 Ocupado -      18:59:55
Braço 2     Vai para o Armazém 1 - 18:59:55
Maquina 3   Escolheu Peça -        18:59:55
Braço 1     Vai para o Tapete 1 - 18:59:55
Maquina 3   Carrega Programa -    18:59:57
Célula      Armazém 2 Pronto -    18:59:57
Maquina 3   Programa Carregado -   18:59:57
Braço 2     Chegou ao Armazém 1 - 18:59:59
Braço 2     Agarra -                18:59:59

----Ocorreu falha no braço 2 -      19: 0: 3

Braço 2 Termina Thread -          19: 0: 3
Controlador 2 Thread Suspensa -    19: 0: 3
Controlador 2 Abortou Programas - 19: 0: 3
Controlador 2 Controlo Activo -    19: 0: 4
Controlador 2 Iniciou Programa do Braço - 19: 0: 4
Braço 1     Chegou ao Tapete 1 -      19: 0: 5
Controlador 1 Braço 1 Pronto -     19: 0: 5
Controlador 2 Abortou Programas - 19: 0:17
Controlador 2 Home Position do Braço - 19: 0:18
Controlador 2 Iniciou Programas - 19: 2:14
Controlador 2 Thread Retomada -    19: 2:18
Peça        Retira Peça de Produção - 19: 2:18
Braço 2     Inicia Thread -          19: 2:18
Braço 1     Reinicia Thread -         19: 2:18
Peça        Braço 2 Ocupado -         19: 2:18
Braço 2     Vai para o Armazém 2 -    19: 2:18
Maquina 3   Reinicia Thread -         19: 2:19

----Falha Recuperada -              19: 2:20

Braço 2     Chegou ao Armazém 2 -    19: 2:22
Braço 2     Agarra -                19: 2:22
Braço 2     Agarrou -               19: 2:31
Braço 2     Vai para o Tapete 1 -    19: 2:31
Braço 2     Chegou ao Tapete 1 -    19: 2:36
Braço 2     Larga -                 19: 2:36
Braço 2     Largou -                19: 2:42
Tapete 1    Avança -                   19: 2:42
Tapete 1    Chegou -                  19: 2:46
Braço 1     Agarra -                19: 2:46
Braço 1     Agarrou -               19: 2:51
Braço 1     Vai para o Buffer 3 -     19: 2:51
Braço 1     Chegou ao Buffer 3 -     19: 2:57
Braço 1     Larga -                 19: 2:57
Braço 1     Largou -                19: 3: 9

```



## APÊNDICE D

# FABRICO ASSISTIDO POR COMPUTADOR CAD/CAM

Um dos objectivos deste trabalho era o de conferir, à *CFE*, a integração vertical suficiente para que esta abrangesse as principais fases do processo de produção. Para isso, foi essencial a adição de um módulo CAD/CAM, constituído pela ferramenta *MasterCam 9 (Mcam9)* e por uma aplicação (*Posto de Produção*) que comunica remotamente com a *CFE*, fazendo a ligação entre o sistema CAD/CAM e a produção.

O CAD (*Computer Aided Drafting & Design*) é parte integrante do subsistema da engenharia de concepção. Consiste num sistema de projecto assistido por computador e tem como principais objectivos: definir os requisitos do produto, determinar a estratégia de concepção e formalizar a concepção por meio de documentação diversa (esquemáticos, listas de detalhes, etc). O software CAM (*Computer Aided Manufacturing*) utiliza as especificações geradas no CAD (geometria da peça) e programa as trajectórias das ferramentas e as operações de manufactura. Este *software* está, normalmente, equipado com um pós-processador que gera automaticamente o programa CNC a ser usado pela máquina CNC.

Além das vantagens anteriormente apresentadas, o CAD/CAM permite reduzir o tempo que decorre desde o início da especificação até ao início da produção, baixa os custos de especificação e desenvolvimento, para além de conseguir uma maior precisão e qualidade nas especificações e nos desenhos. Os produtos são cada vez mais complexos e, cada vez, mais sujeitos a modificações. Os sistemas de CAD/CAM facilitam a manutenção e actualização da documentação e podem obviar à necessidade da existência de protótipos de teste e simulação. Este tipo de sistemas possibilitam, às empresas, ultrapassar esforços de concepção e especificação cada vez mais prolongados e complexos, face a produtos cujo ciclo de vida tem vindo a ser encurtado. Na produção de uma peça mecânica, a integração dos sistemas CAD/CAM torna possível, a partir do esquema ou desenho da peça:

- Obter o modelo bi ou tri-dimensional da peça,
- proceder a estudos de engenharia sobre o modelo e, eventualmente, re-especificar o modelo,
- estimar custos de produção e analisar soluções alternativas,
- planear a sequência de operações do processo de fabrico e gerar os programas para as máquinas de controlo numérico que irão executar a peça,
- transferir estes programas para as máquinas.

Na figura D.1 está representado um diagrama das principais fases de um projecto CAD/CAM. Nas secções seguintes deste capítulo serão descritas, com mais pormenor, cada uma das referidas fases.

### D.1 CAD

#### Ferramenta CAD e Limitações na produção

O *Mcam9* reúne, num só bloco, os processos de modelação e desenho, simulação e pós-processamento. A esta vantagem juntam-se ainda outras:

- Compatibilidade com as principais ferramentas de desenho vectorial existentes.
- Actualidade, sendo, por isso, usado nas principais empresas de produção de moldes nacionais.
- Adaptação às principais máquinas CNC existentes no mercado, devido a um pós-processamento adaptável.
- Comunicação directa com a máquina via *RS232*.

Como referido no capítulo 2, as máquinas da *CFE* possuem algumas características que limitam o processo de modelação. Entre elas destacam-se as seguintes:

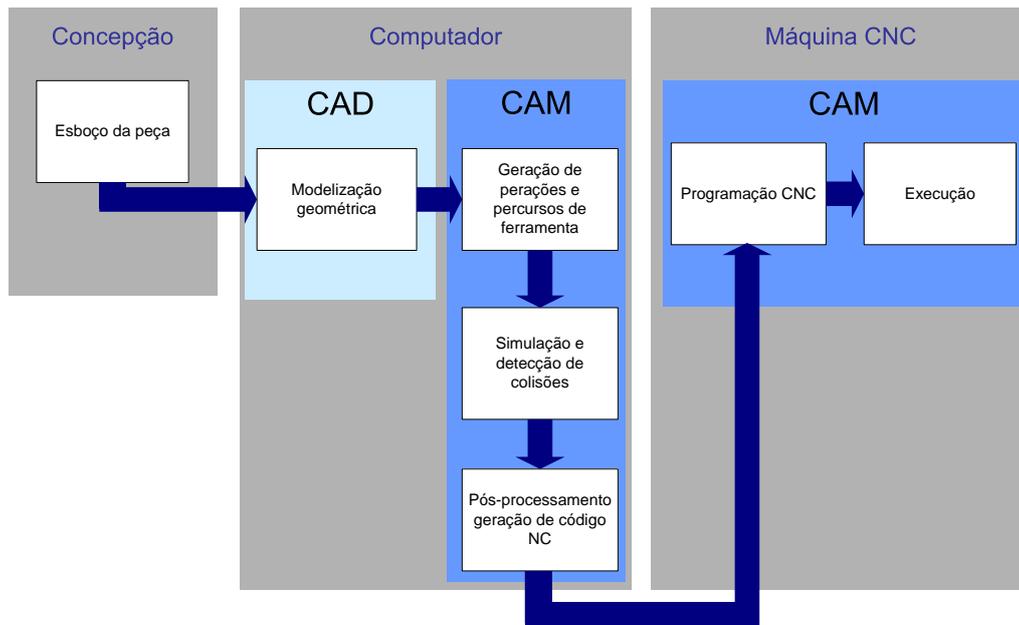


Figura D.1: Diagrama das fases de um projecto CAD/CAM/CNC

- Reduzida capacidade de armazenamento, que levam à impossibilidade de criação de programas CNC com mais de 222 blocos (linhas). A solução para esta limitação passa pela divisão automática do programa principal em subprogramas.
- Tamanho da matéria bruta, limitado pelos efectores terminais dos braços que efectuam o transporte e pelo dispositivo de fixação das máquinas.
- Ferramentas de maquinaria disponíveis.

Na criação do modelo exemplificativo, foram tidas em conta estas limitações. A elaboração deste modelo não pretende um levantamento exaustivo de todas potencialidades do programa mas, em vez disso, pretende ilustrar a facilidade com se podem ultrapassar todas as fases de concepção e criação do modelo CAD da peça, até à obtenção do programa CNC final. Optou-se, também, por apenas utilizar uma das máquinas fresadoras, para não tornar demasiada exaustiva a demonstração. O processo para um modelo a ser maquinado pelo torno é essencialmente o mesmo, salvo algumas diferenças inerentes ao próprio processo de fabrico.

### Modelo CAD da peça

Relativamente à peça a criar, tal como referido anteriormente, existiam algumas limitações inerentes às características das próprias máquinas. Optou-se por criar uma peça que poderia ser usada, por exemplo, como um molde para criação de uma peça de plástico. Este exemplo permite demonstrar algumas, das muitas, capacidades deste tipo de máquinas.

Na figura D.2 está o modelo CAD da peça que se pretende obter no fim do processo de produção. Estes modelos podem ser criados numa qualquer ferramenta CAD e depois exportados para o *MCam9*.

## D.2 CAM

### Bruto de Maquinagem

Antes de proceder ao desenho da peça, há que definir qual o material a usar e quais as dimensões do bruto de maquinagem que originará a peça.

O material usado no exemplo da demonstração foi o poliuretano, já usado em [5]. Na figura D.3 está representada a dimensão e a forma do bruto de maquinagem utilizado.

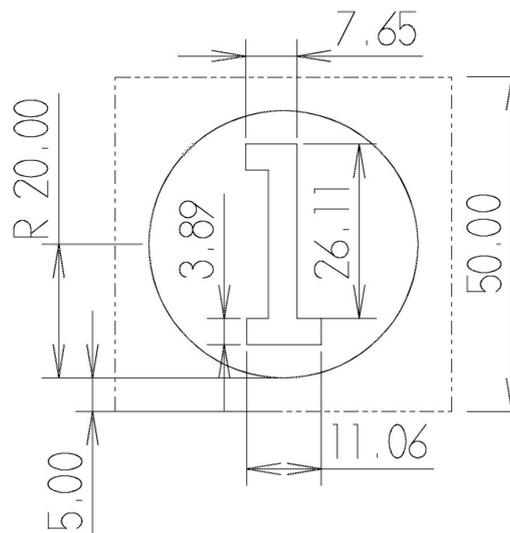


Figura D.2: Vista de cima do modelo CAD da peça com as dimensões em milímetros

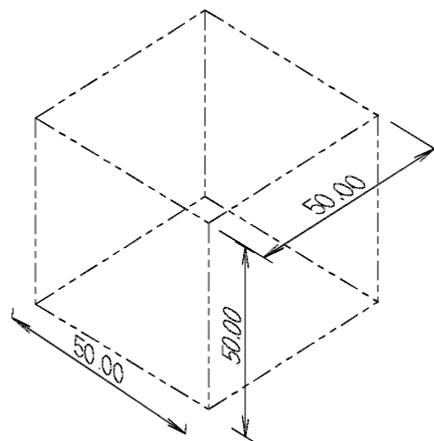


Figura D.3: Peça Bruta com as dimensões em milímetros

### Dispositivos de Fixação

A fresadora dispõe, na sua base, de uma prensa de aperto que fixa, sob pressão, a matéria bruta a ser maquinada. Como só uma das faces do cubo é que é maquinada a uma profundidade de 5mm, não houve a preocupação de a ferramenta colidir com a sua própria base de fixação.

### Ferramentas e Operações de Maquinação

Para a realização deste molde, usou-se apenas um aperto e uma operação. De notar que a peça em bruto já tem todas as faces faceadas com as dimensões finais, pelo que não há necessidade de proceder a esse tipo de operações. Como as peças são obtidas através do corte de um serrote mecânico, é de esperar que existam erros nas suas dimensões exteriores, no entanto, tais erros não têm qualquer influência no molde idealizado. Para a operação de fresagem, usou-se uma fresa com as dimensões que se apresentam na figura D.4.

De notar que a ferramenta escolhida foi aquela que, por um lado, permitisse maquinar na peça bruta o desenho escolhido e, por outro, devido ao seu diâmetro, produzisse o mesmo resultado final num menor tempo de maquinação devido às restrições referidas anteriormente.

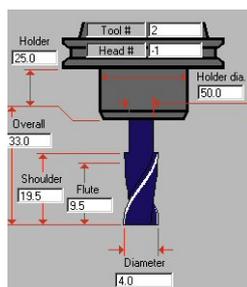


Figura D.4: Esquema do Suporte de fixação e da ferramenta (dimensões em milímetros)

A operação de maquinação consiste na criação de um baixo-relevo (*pocket*) em torno do algarismo 1. Depois de escolhidas as superfícies limite do *pocket*, o *MCam9* gera, automaticamente, todo o caminho que a fresa efectua (*toolpath*), quer a entrar no material, a movimentar-se, ou a sair do mesmo, sempre de acordo com a tamanho da ferramenta.

Nas figuras D.5 e D.6 encontra-se o caminho que a ferramenta percorre durante a maquinação da peça.

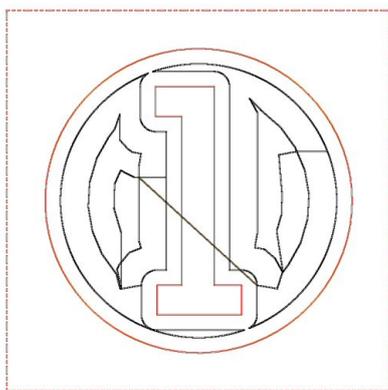


Figura D.5: Caminho que a ferramenta irá efectuar no material - caminho feito pela ferramenta a preto

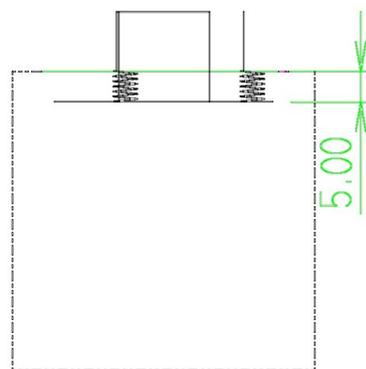


Figura D.6: Vista lateral da operação a efectuar pela ferramenta

### Simulação e Pós-processamento

O *MCam9* possui um simulador que, ao reproduzir fielmente os comandos do código CNC, permite pré-visualizar as acções que a fresadora irá desenrolar até ao fabrico completo do molde. A figura D.7 é demonstrativa desse mesmo simulador na fase final da maquinação da peça bruta. Os tempos de maquinação e a correcção das operações de maquinação poderão, então, ser testados sem se recorrer à produção real de peças, suprimindo todos os custos que daí adviriam.

As operações geradas são, depois, convertidas em código CNC pelo pós-processador do programa, apresentado em anexo. O pós-processador consiste na operação de "tradução" da linguagem gerada pelo *MCam9* para a linguagem CNC da máquina. Isto deve-se à variação dos formatos dos ficheiros CNC de máquina para máquina. Foi criado um pós-processador adaptado às máquinas da *CFE*, visto que o *MCam9* não as contempla nesta versão.

### D.3 Tecnologia CNC

Gerados os ficheiros em código CNC, a produção das peças fica, então, entregue às máquinas CNC. O Controlo Numérico (NC) define-se como um método de automação, no qual as acções de uma máquina são controladas por um programa contendo dados codificados alfanumericamente. As máquinas CNC incorporam,

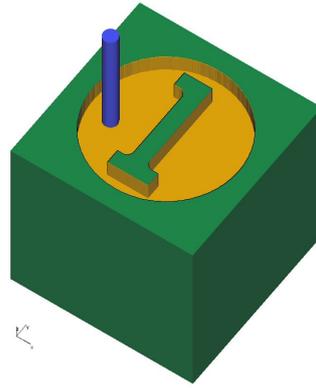


Figura D.7: Simulador do Programa *MCam9*

também, um computador que controla o programa a executar, lendo-o da memória. Este tipo de máquinas surgiu por volta de 1950 e revolucionou os processos de manufactura a nível mundial. É possível enumerar as seguintes vantagens do uso deste tipo de máquinas:

1. Redução nos tempos de fabricação.
2. Maior repetibilidade na sequência das operações, fazendo com que os tempos padrões previstos sejam mais rígidos, obtendo-se, desta forma, uma maior exactidão de custos.
3. Redução nos tempos de preparação (*set-up*) tornando viável a produção de pequenos lotes.
4. Redução de itens acabados no stock, por se possibilitar produção de pequenos lotes.
5. Redução nos tempos e na frequência, com que as inspecções de qualidade são efectuadas.
6. Qualidade do produto final independente da fadiga do operador.

O sistema de controlo numérico baseia-se no diagrama da figura D.8.

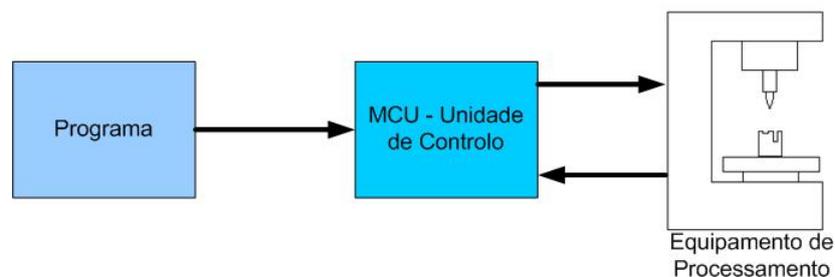


Figura D.8: Sistema de Controlo Numérico

### Programa

O programa CNC foi gerado com base em coordenadas relativas. A posição inicial das bases de *M1* ou *M3* corresponde à posição onde *B1* deposita a peça, que servirá como referência para o resto do processo de maquinação.

O programa contém os dados acerca da peça e informações adicionais, como a velocidade de corte, a velocidade de avanço e o *timing* de mudança de ferramenta. Qualquer erro ao nível do programa NC (coordenada ou velocidade fora de alcance e outros) gera um erro fatal na máquina. Estes erros só poderão ser recuperados com a intervenção humana, através da reinicialização da máquina e da colocação da base e da fresa nas posições iniciais, uma vez que o controlo de movimento não possui informação acerca das posições actuais do mesmo, características do controlo por motores passo-a-passo. De salientar, também, que sempre que a máquina é

reiniciada a origem das coordenadas absolutas passa a ser a posição actual, o que também limita a automação da recuperação da falha.

### **MCU - Unidade de Controlo**

Nesta unidade, está localizado o *hardware* de controlo de movimento, constituído pelos sistemas de processamento, de interface para leitura do programa, controlo dos motores e algoritmos de geração de trajectória. O MCU possui um interface digital que permite comandar e observar diversos parâmetros da máquina, tais como: estado do aperto da máquina, comando de emergência activado, interrupção do fabrico, etc. Este interface será muito importante para monitorizar o comportamento da máquina e recuperar eventuais falhas.

### **Equipamento de Processamento**

Do equipamento de processamento, fazem parte os elementos motrizes da máquina que efectuam as tarefas. Nas máquinas da *CFE*, os motores são todos do tipo passo-a-passo e não existem, sequer, sensores de fim de curso a fim de limitar o movimento dos mesmos.

De lamentar as limitações ao nível da recuperação de falhas impostas pelas máquinas. Estas limitações tornam, quase sempre, indispensável a intervenção humana. Ainda assim, o interface digital permite uma razoável interacção entre as máquinas e o processo de controlo da *CFE*.

## APÊNDICE E

# BRAÇOS ROBOT SCORBOT - ER VII

### E.0.1 Robot

O robot possui uma área de trabalho limitada pela máxima excursão que cada junta pode efectuar. A figura E.1 ilustra os limites desta.

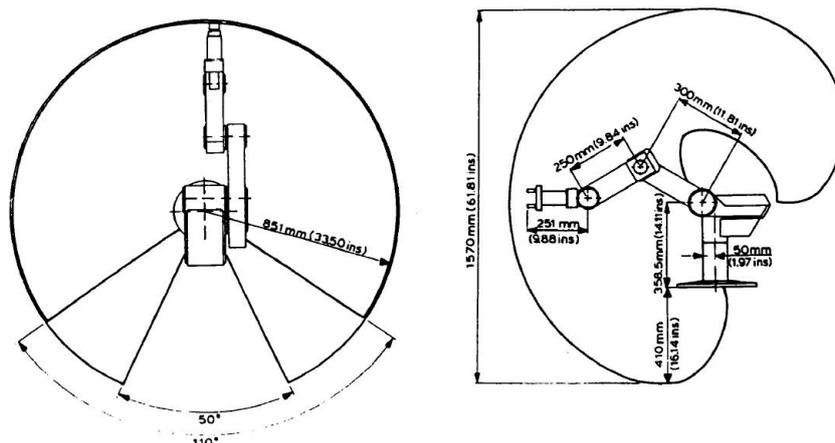


Figura E.1: Limites da área de trabalho

A posição do robot pode ser definida por 5 ângulos, correspondentes à posição relativa de cada junta. No entanto, é mais comum definir a posição do efector terminal em coordenadas cartesianas. Neste caso, como o braço tem 5 juntas, é necessário recorrer aos ângulos de *pitch* e *roll* para definir correctamente a posição e orientação do efector terminal. Surgem, desta forma, dois sistemas de coordenadas equivalentes para descrever a configuração do robot. Na tabela E.1 estão representados os dois sistemas.

Tabela E.1: Sistema de coordenadas do robot

Coordenadas Polares	Coordenadas Cartesianas
$\theta_1$	x
$\theta_2$	y
$\theta_3$	z
$\theta_4$	$\theta_4$
$\theta_5$	$\theta_5$

Na programação do robot pode-se utilizar, indiferentemente, qualquer um dos dois sistemas de coordenadas, visto serem equivalentes. Porém, cabe ao respectivo controlador efectuar as necessárias transformações entre eles (cinemática directa e inversa).

Na tabela E.2 estão as principais características do braço robot Scorbot ER-VII.

### E.0.2 Controlador

Na figura E.2 está o esquema do controlador, com a respectiva legenda na tabela E.3.

Na tabela E.4 apresentam-se as características mais importantes do controlador.

Tabela E.2: Características do robot

Item	Especificação
Estrutura mecânica	Robot articulado na vertical com cinco eixos
Área de trabalho:	
Junta 1	250°; 310° por programação do utilizador
Junta 2	170°
Junta 3	225°
Junta 4	180°
Junta 5	360°
Efecto terminal	efector terminal eléctrico tipo pinça
Actuadores	Servo motores eléctricos DC
Realimentação	Encoders ópticos em todos os eixos
Carga de trabalho	2 kg ( <i>efector terminal</i> incluído)
Repetibilidade	0.2mm
Velocidade máxima	1000mm/sec
Peso	30 kg

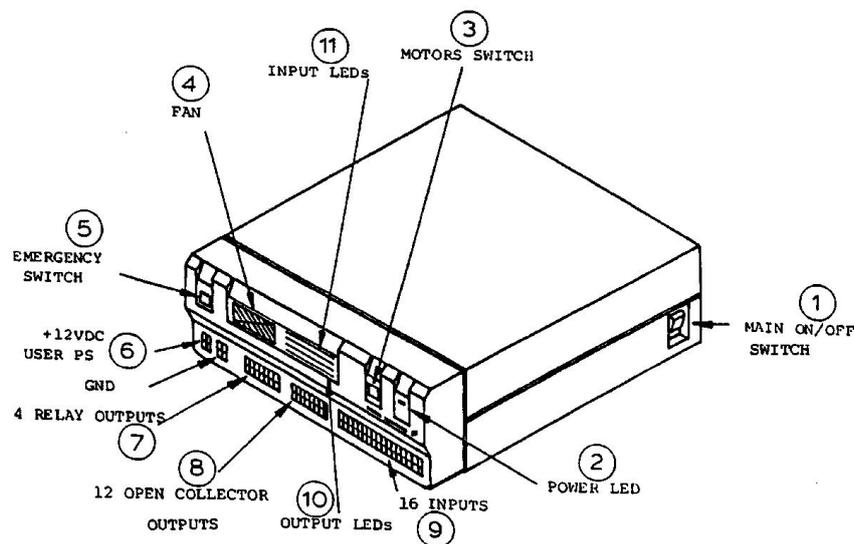


Figura E.2: Esquema do Controlador do Braço

## Sistema de Controlo

A referência de posição e velocidade é calculada em ciclos de controlo de 10 ms, por um controlador do tipo PID. Para esse cálculo o controlador serve-se da posição actual da junta, fornecida pelo encoder óptico que a ela está acoplado.

O processador calcula a posição e velocidade desejadas e devolve o valor (digital) obtido na gama  $[-5000; 5000]mV$ . O conversor analógico converte o valor digital num analógico dentro da gama  $[-5; 5]V$ , que será aplicado a um gerador analógico de PWM a  $20kHz$ . O Duty-Cycle da onda dita a velocidade do eixo e faz com que o drive de potência do motor comute  $\pm 24VDC$ , a  $20kHz$ . Como o motor não consegue reagir a uma frequência tão alta será afectado apenas pelo valor médio da onda. Na figura E.3 apresenta-se a malha do ciclo de controlo atrás descrito.

Tabela E.3: Legenda da figura E.2

1	Interruptor principal
2	LED indicador ligado/desligado
3	Interruptor para corte de energia aos motores
4	Ventoinha
5	Botão de paragem de emergência
6	Fonte de alimentação disponível de 12V DC
7	Saídas de relés
8	Saídas digitais
9	Entradas digitais
10	LEDs indicadores do estado das saídas digitais
11	LEDs indicadores do estado das entradas digitais

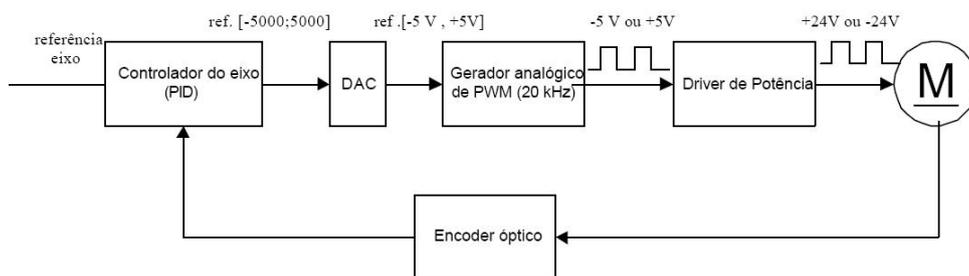


Figura E.3: Esquema do Controlador de cada eixo do Braço

Tabela E.4: Características do controlador do robot

Item	Especificação	Observações
Tipo de controlador por eixo	PID	
Número de eixos	8 (máximo 11)	
Controlo dos eixos	PWN , 20kHz	<i>Pulse Width Modulation</i>
Ciclo de controlo	10ms	
Controlo de trajectória	ponto-a-ponto, interpolação linear e circular	
Perfil de controlo de velocidade	Parabolóide, trapezoidal, malha aberta	Aceleração e desaceleração definíveis separadamente para cada grupo de controlo
Velocidade	Definível por velocidade ou tempo entre dois pontos	Programável em percentagem de um valor máximo
CPU	Motorola 68010	
EPROM	384 Kbytes	
RAM de trabalho	64 Kbytes	
RAM do utilizador	128 Kbytes	Não volátil (NVRAM)
Número de linhas de programa e posições	12800 linhas de programa ou 6375 posições	Cada posição requer o dobro do espaço de uma linha de programa
Sistema operativo	Proprietário multi-tarefa podendo executar até 20 programas independentes	Possibilidade de edição de um programa com outros em execução
Saídas	16 Saídas digitais	4 saídas de relés (NA e NO) e 12 saídas em colector aberto
Comunicação	Série RS232C	Ficha D25, Controlo XON/XOFF
Métodos de programação	1. ACL 2. Scorbace Level 5	
Sistemas de coordenadas	Coordenadas polares ou cartesianas	
Protecções	Hardware: Limite de corrente por eixo Fusível em cada eixo Software: Protecção de impacto Protecção térmica Limites para cada eixo	
Alimentação	100/110/230 VAC, 50/60 Hz 500W máximo	
Fonte de alimentação para utilizador	12 VDC, 2A	Regulada
Alimentação dos motores	24 VDC, 18A	
Peso	19Kg	

## APÊNDICE F

# MÁQUINAS DA *CFE*

As fresadoras são do modelo Emco Model F1-CNC Vertical/Horizontal Milling Machine Model F1-CNC - Sistema dos três eixos controlados por Microprocessador CNC.

### Principais Características das Fresadoras:

- (3 x 2 Axes)
- Memória para o programa limitada a 222 blocos
- Programação no Sistema Inglês (Inch) / Sistema Métrico
- Interpolação Linear
- Interpolação Circular
- Interface RS232
- Monitor de 12"

As portas RS232 das máquinas CNC comunicam segundo as especificações da tabela F.1.

Tabela F.1: Especificações das portas RS232

Baud Rate	300
Data bits	7
Stop bits	2
Paridade	Não
Modo	XON/XOFF

Tabela F.2: Interface X22 - Interface de verificação e comando

Função	Estados da máquina	Comando
Base de fixação	Aberta / Fechada	Abrir / Fechar
Refrigeração	On / Off	-
Carregamento do programa	-	Receber / Enviar
Modo de Operação	Manual / CNC	Manual / CNC
Estado do Programa	Carregado / Não Carregado	-
Paragem da Máquina	-	On / Off
Emergência	On / Off	-

Tabela F.3: Especificações técnicas da fresadora EMCO-F1

<b>Specification</b>	<b>Description</b>
<b>Working Range:</b>	
- X-Axis (Longitudinal)	7.87"
- Y-Axis (Transverse)	3.94"
- Z-Axis (Vertical)	7.87"
- Milling Head Rotation - Vertical/Horizontal	90 degrees
<b>Milling Table</b>	
- Table Surface	16.54" x 4.92"
- T-Slots - Number	2
- Slot Width, Spacing	0.43" , 3.54"
<b>Distance - Cutting Spindle / Milling Table</b>	
- Vertical	1" - 8.86"
- Horizontal	3.15" - 11.02"
<b>Cutting Spindle</b>	
- Tool Mounting	#30 V-Flange
- Tool Clamping	Quick Change Clamping Device
- Throat Clearance	5.51"
<b>Cutting Spindle Drive</b>	
- DC Motor	0.6 HP
- Spindle Speeds, Stepless	300 - 2,000 RPM
<b>Feed Drive</b>	
- Feeds in Manual Operation, Stepless	0.2 - 15.75 IPM
- Feeds in CNC Operation	0.08 - 19.65 IPM
- Single Step	0.0001"
- Rapid Traverse Speeds	23.62 IPM
<b>Dimensions and Weight</b>	
- Machine Weight, Approx	265 lbs.
- Control Unit Weight	88 lbs.
- Dimensions of Machine	33" W x 29.5" D x 28.4" H
- Dimensions of Control System	28" W x 17.7" D x 21.3" H

# Bibliografia

- [1] Andrea Bobbio. System Modelling with Petri Nets, pages pp 102–143. Kluwer p.c., 1990.
- [2] S. Christensen and L. Petrucci. Modular analysis of petri nets. Computer Journal, 43(3):224–242, 2000.
- [3] Jörg Desel. Place/transition-nets i. In Introductory Tutorial - Petri Nets, Junho 2000. 21st International Conference on Application and Theory of Petri Nets.
- [4] D.N. Pandalai e L.E. Holloway. Template languages for fault monitoring of timed discrete event processes. IEEE Transactions on Automatic Control, 45(5), May 1999.
- [5] Gonçalo Gonçalves e Pedro Henrique Delgado. Estudo e desenvolvimento de um gestor multitarefa para uma célula de fabricação flexível. Trabalho Final de Curso, Instituto Superior Técnico, 1997.
- [6] Gonçalo Gonçalves e Pedro Henrique Delgado. Manual Técnico da Célula de Fabricação Flexível. Instituto Superior Técnico, 1997.
- [7] Christos G. Cassandras e Stéphane Lafortune. Introduction to Discrete Event Systems. The Kluwer International Series On Discrete Event Dynamic System. Yu-Chi Ho, first edition, 1999.
- [8] Eshed Robotec. Advanced Control Language - Reference Guide, third edition, 1982.
- [9] Marian V. Iordache and Panos J. Antsaklis. Synthesis of supervisors enforcing firing vector constraints in petri nets. Technical report, ISIS Group at the University of Notre Dame, Notre Dame, IN 46556, USA, February 2002.
- [10] M.F.M. Lima. Desenvolvimento de células de fabricação flexível: Simulação e controlo, Instituto Superior Técnico, 1996.
- [11] Jonh O. Moody and Panos J. Antsaklis. Supervisory Control Of Discrete Event Systems Using Petri Nets. Kluwer Academic Publishers, 1998.
- [12] Hernâni Raul Vergueiro Monteiro Cidade Mourão. Estudo e realização de células de fabrico flexivel: Desenvolvimento de uma aplicação. Tese de Mestrado, Instituto Superior Técnico, Junho 1993.
- [13] Tadao Murata. Petri nets: Properties, analysis and applications. Proceedings Of The IEEE, VOL. 77(NO. 4):pp. 541–580, April 1989.
- [14] James L. Peterson. Petri Net Theory. Prentice-Hall, Inc., Englewood Cliffs, N.J. 07632, 1981.
- [15] Stephen Roch and Peter H. Starke. Integrated Net Analyser 2.2. Universitat zu Berlin - Institut fur Informatik.
- [16] Marcos Renato Da Silveira. Une contribution à la supervision des systèmes flexibles de production des systems flexibles de production manufacturière: L'Évaluation de la faisabilité de la reprise, Ecole Doctorale Systemes, 2000.