

TESE DE DOUTORADO N° 193

**MÉTODOS DE CONTROLE DE FLUXO DE ENTRADA PARA SISTEMAS  
MODELADOS POR GRAFOS DE EVENTOS TEMPORIZADOS**

**José Ricardo da Silva Dias**

DATA DA DEFESA: 10/11/2014

**Universidade Federal de Minas Gerais**

**Escola de Engenharia**

**Programa de Pós-Graduação em Engenharia Elétrica**

**MÉTODOS DE CONTROLE DE FLUXO DE ENTRADA PARA  
SISTEMAS MODELADOS POR GRAFOS DE EVENTOS  
TEMPORIZADOS**

José Ricardo da Silva Dias

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica.

Orientador: Prof. Carlos Andrey Maia

Belo Horizonte – MG

Novembro de 2014

**"Métodos de Controle de Fluxo de Entrada para Sistemas Modelados por Grafos de Eventos Temporizados"**

**José Ricardo da Silva Dias**

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do grau de Doutor em Engenharia Elétrica.

Aprovada em 10 de novembro de 2014.

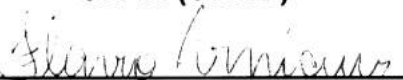
Por:



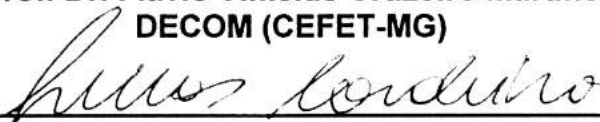
Prof. Dr. Carlos Andrey Maia  
DEE (UFMG) - Orientador



Prof. Dr. Vicente Ferreira de Lucena Jr.  
UFAM (UFMG)



Prof. Dr. Flávio Vinícius Cruzeiro Martins  
DECOM (CEFET-MG)



Prof. Dr. Lucas C. Cordeiro  
CETELI (UFAM)



Profa. Dr. Patrícia Nascimento Pena  
DELT (UFMG)



Prof. Dr. Guilherme Augusto Silva Pereira  
DEE (UFMG)

UNIVERSIDADE FEDERAL DE MINAS GERAIS

ESCOLA DE ENGENHARIA

PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

**Métodos de Controle de Fluxo de Entrada para  
Sistemas Modelados por Grafos de Eventos Temporizados**

José Ricardo da Silva Dias

Orientador: Prof. Dr. Carlos Andrey Maia

Co-Orientador: Prof. Dr.-Ing. Vicente Ferreira de Lucena Junior

Belo Horizonte, Novembro de 2014

# Métodos de Controle de Fluxo de Entrada para Sistemas Modelados por Grafos de Eventos Temporizados

Tese de Doutorado submetida à Banca Examinadora designada pelo Colegiado do Programa de Pós-Graduação em Engenharia Elétrica da Escola de Engenharia da Universidade Federal de Minas Gerais, como requisito para obtenção do Título de Doutor em Engenharia Elétrica, a ser avaliado por:

Banca Examinadora:

- Prof. Dr. Carlos Andrey Maia (DEE-UFMG)
- Prof. Dr. -Ing. Vicente Ferreira de Lucena Jr. (DEE-UFAM)
- Prof. Dr. Flávio Vinícius Cruzeiro Martins (DECOM-CEFET/MG)
- Prof. Dr. Lucas Carvalho Cordeiro (CETELI-UFAM)
- Prof<sup>a</sup>. Dr<sup>a</sup>. Patrícia Nascimento Pena (DELT-UFMG)
- Prof. Dr. Guilherme Augusto Silva Pereira (DEE-UFMG)

# Agradecimentos

Agradeço a Deus por todas as bênçãos concedidas, porque nEle vivemos, nos movemos, e existimos;

Por Ele encaminhar todas as coisas, por mover pessoas para me auxiliarem nos trabalhos e dificuldades;

À minha esposa Andrea Valeria Basualto González Dias e aos meus filhos André Ricardo, Samuel Elias e David Joshua, pela compreensão e amor;

Ao meu orientador, Professor Dr. Carlos Andrey Maia, pela disposição em desenvolver este trabalho, pela orientação e paciência;

Ao meu co-orientador, Professor Dr. -Ing. Vicente Ferreira de Lucena Junior, pela disposição em contribuir para o desenvolvimento deste trabalho;

Aos coordenadores, Professor Dr. José Pinheiro de Queiroz Neto e Professor Dr. Reinaldo Martinez Palhares, pelo auxílio nos momentos de dificuldades;

Aos professores do DINTER, Dr. Guilherme Augusto S. Pereira, Dr. Hani Camille Yehia e Dr. Eduardo Mazoni A. M. Mendes;

Aos novos amigos e irmãos: Áurea, Cíntia, Ronaldo, Juliana e família, pelo ambiente familiar nesta cidade;

Aos alunos do DINTER: Orlewilson Maia, João Guilherme e Marisol Plácido;

Aos amigos Enivaldo Correa, Leontino Neves e Adelson Vasconcelos pelo suporte técnico no início desta caminhada;

À UFMG, IFAM e CAPES pela concretização do projeto DINTER, pela oportunidade de desenvolvimento profissional;

À FAPEAM pelo apoio financeiro, como bolsista do programa RH-Interinstitucional.



## Resumo

Esta tese trata do controle de sistemas modelados por grafos de eventos temporizados (GET) e pela álgebra max-plus. Tais sistemas são sujeitos a fenômenos de sincronização e atraso de tempo. O foco principal para aplicação são os sistemas de manufatura, não se limitando, no entanto, a estes. Atualmente, os problemas de controle de sistemas modelados por GET são abordados de duas maneiras diferentes: (a) baseada em dígrafo da série de potências formal e, (b) baseada em dígrafo de datadores. A principal característica da primeira abordagem reside no fato de que esta baseia-se em técnicas de funções de transferência. A segunda abordagem, baseada em dígrafos de datadores, nos permite lidar diretamente com a realização do sistema. Os principais tipos de controle que utilizam as funções de transferências são os métodos por modelo de referência. O objetivo do controle por modelo de referência refere-se à imposição de um comportamento desejado para um determinado sistema, o qual, por sua vez, é ditado pelo uso de um controlador de realimentação. No contexto de GET, esse controlador minimiza a quantidade de fichas (matéria-prima) no GET e atrasa ao máximo possível a entrada delas no sistema. Para um determinado modelo de referência, o problema enfrentado consiste em encontrar o melhor controlador, caso ele exista. Esses problemas de controle para GET são usualmente propostos utilizando-se a política de gestão *just-in-time* (JIT), na qual, o objetivo é entregar a matéria-prima no momento exato da necessidade dos postos de trabalho, sem formar estoques desnecessários no interior da planta fabril, mas atendendo as especificações de demanda de produção. Nesta tese são propostos dois métodos de controle para sistemas modelados por GET, atuando no controle do fluxo de entrada matéria-prima do sistema. O primeiro método propõe o cálculo dos parâmetros de um controlador de realimentação que atrasa ao máximo a entrada de matéria-prima no sistema modelado por GET, e também calcula o número de fichas no sistema de realimentação, limitando ao máximo as fichas no sistema modelado. O segundo método de controle propõe a sincronização do disparo de várias transições internas em um sistema modelado por GET. Esse controle é obtido atrasando-se a entrada de matéria-prima no sistema e, na maioria dos casos, modificando-se a taxa de produção do sistema. Esse tipo de controle é importante, pois sincronizar transições permite definir o momento exato em que um evento ou atividade pode começar ou terminar numa sequência de atividades definidas. Os resultados obtidos reduzem a complexidade para o cálculo dos controladores para os dois tipos de controle apresentados, o que significa uma diminuição da carga computacional, proporcionando um efeito de estabilização do sistema, evitando o *overflow* dos estoques, usando como base a política de gestão JIT.

**Palavras-chave:** sistemas a eventos discretos, grafo de eventos temporizados, álgebra max-plus, controle de sistemas de manufatura.





# Abstract

This thesis deals with the control of systems modeled by timed events graphs (TEG) and the max-plus algebra, such systems are subject to synchronization phenomena and time delay. The main focus for application are manufacturing systems, however it is not limited to only these. Currently, the problems of control systems modeled by TEG are approached in two different ways: (a) based on dioid the formal power series, and (b) based on dioid of daters. The main feature of the first approach, lies in the fact that this is based on techniques of transfer functions. The second approach, based on dioid of daters, allows to deal directly with the fulfillment of the system. The main types of control that use transfer functions are the methods by reference control. The goal of the model reference control refers to the imposition of a desired behavior for a given system, which in turn is dictated by the use of a feedback controller. In the context of TEG, this minimizes the amount of controller token on TEG and delays as much as possible the entry of raw materials in the system. For a given model reference control, the problem faced consists of finding the best controller, in case it exists. These control issues to TEG are usually proposed in the just-in-time (JIT) management policy, in which the goal is to deliver the raw material at the exact moment of need of jobs, without forming unnecessary stocks within the manufacturing plant, but given the specifications of production demand. The goal of this thesis is to propose two control methods based on dioid of daters. The first method propose the calculation of parameters of a feedback controller that delays the entry of raw materials in the system modeled by TEG, also it calculates the number of tokens in the feedback system, limiting the most tokens in the modeled system. The second method, control method II, proposes the firing synchronization of several internal transitions in a system modeled by TEG. This control is achieved by delaying the entry of raw materials in the system and, in most cases, by modifying the production rate of the system. This type of control is important because synchronizing transitions allow to define the exact time that an event or activity can begin or end, in a sequence of defined activities. Where results are expected, the proposed methods reduce the complexity for the calculation of the controllers for these two types of control, which would lead to a reduction in the computational load, providing a stabilizing effect of the system, avoiding stocks overflow, using as parameter the JIT management policy.

**Keywords:** Discrete Event System, Timed Event Graph, Max-plus algebra, Control of Manufacturing Systems.



# Sumário

<b>1</b>	<b>Introdução</b>	<b>1</b>
1.1	Revisão da Literatura . . . . .	3
1.2	Objetivos . . . . .	7
1.3	Organização do Trabalho . . . . .	8
<b>2</b>	<b>Álgebra Max-plus</b>	<b>9</b>
2.1	Breve Histórico . . . . .	9
2.2	Álgebra Max-Plus . . . . .	10
2.2.1	Conceitos Básicos e Definições . . . . .	10
2.3	Exemplo Ilustrativo . . . . .	11
2.4	Propriedades Algébricas da Álgebra Max-Plus . . . . .	20
2.4.1	Vetores e Matrizes na Álgebra Max-Plus . . . . .	24
2.5	Teoria Espectral de Matrizes . . . . .	25
2.5.1	Grafos e Matrizes . . . . .	27
2.5.2	Definições e Teoremas Importantes da Teoria Espectral de Matrizes . . . . .	29
2.5.3	Solução de Equações Lineares . . . . .	31
2.6	Elementos da Teoria da Residuação . . . . .	34
2.6.1	Residuação do Produto para o Dioide Matricial . . . . .	35
2.7	Conclusão . . . . .	35
<b>3</b>	<b>Modelagem Matemática de Sistemas</b>	<b>37</b>
3.1	Sistemas a Eventos Discretos . . . . .	38
3.1.1	Evento . . . . .	38
3.1.2	Sistemas Dirigidos pelo Tempo e Sistemas Baseados em Eventos . . . . .	38
3.1.3	Propriedades Características de Sistemas a Eventos Discretos . . . . .	39
3.1.4	Exemplos de Sistemas a Eventos Discretos . . . . .	40
3.2	Redes de Petri . . . . .	43
3.2.1	Fundamentos de Redes de Petri . . . . .	44
3.2.2	Evolução Dinâmica das Redes de Petri . . . . .	46
3.2.3	Equações de Estado . . . . .	49
3.2.4	Modelos de Redes de Petri para Sistemas com Filas . . . . .	50
3.2.5	Redes de Petri Limitada . . . . .	51

3.3	Modelos Temporizados . . . . .	52
3.3.1	Redes de Petri Temporizadas . . . . .	52
3.4	Grafos de Eventos Temporizados . . . . .	53
3.4.1	Descrição de um GET por Datadores . . . . .	54
3.4.2	Representação das Equações Utilizando-se Datadores . . . . .	55
3.4.3	Inicialização . . . . .	57
3.4.4	Forma Matricial . . . . .	58
3.4.5	Forma Canônica - Eliminação da Parte Implícita . . . . .	58
3.4.6	Equações de Estado . . . . .	59
3.4.7	Exemplo . . . . .	60
3.5	Matrizes de Transferência com uso da Transformada- $\gamma$ . . . . .	61
3.5.1	Transformada- $\gamma$ . . . . .	61
3.5.2	Obtenção da Matriz de Transferência de um Sistema . . . . .	62
3.6	Dinâmica dos GET . . . . .	63
3.6.1	Noções de Estabilidade de GET . . . . .	64
3.7	Exemplos de Aplicações . . . . .	65
3.8	Conclusão . . . . .	67
<b>4</b>	<b>Métodos de Controle Desenvolvidos</b>	<b>69</b>
4.1	Pressupostos e Procedimentos de Modelagem . . . . .	71
4.2	Método de Controle I . . . . .	73
4.2.1	Síntese do Controlador . . . . .	76
4.2.2	Complexidade Computacional do Método I . . . . .	78
4.2.3	Exemplo Ilustrativo: Comparação entre os Resultados do Método I e do Modelo de Referência . . . . .	79
4.2.4	Aplicação do Método I para Sistemas com Várias Entradas e Várias Saídas . . . . .	83
4.2.5	Considerações sobre os Resultados do Método I . . . . .	90
4.3	Método de Controle II . . . . .	91
4.3.1	Notação Vetorial . . . . .	91
4.3.2	Método de Controle II: Descrição do Problema . . . . .	92
4.3.3	Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Duas Transições . . . . .	103
4.3.4	Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Três Transições . . . . .	106
4.3.5	Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Quatro Transições . . . . .	107
4.3.6	Outro Exemplo Ilustrativo para a Sincronização de Quatro Transições . . . . .	111
4.3.7	Considerações sobre o Método II . . . . .	112

<b>5</b>	<b>Conclusões</b>	<b>113</b>
5.1	Considerações Finais e Perspectivas . . . . .	113
<b>A</b>	<b>Controle por Modelo de Referência</b>	<b>115</b>
A.1	Síntese de Controladores . . . . .	116
A.1.1	Pré-compensação . . . . .	116
A.1.2	Realimentação de saída . . . . .	118
	<b>Referências Bibliográficas</b>	<b>121</b>



## Lista de Abreviaturas

GET	Grafos de Eventos Temporizados.
HST	High-Throughput Screening.
JIT	<i>Just-in-Time.</i>
MCP	Modelo de Controle Preditivo.
MIMO	Multiple-Input and Multiple-Output.
MISO	Multiple Input, Single Output.
SDVC	Sistemas Dinâmicos de Variáveis Contínuas.
SED	Sistemas a Eventos Discretos.





## Lista de Símbolos

$\oplus$	Operador de Maximização (max-plus).
$\otimes$	Operador de Soma (max-plus).
*	Estrela de Kleene.
$e$	Elemento 0.
$\varphi$	Tempos de Atrasos do Controlador.
$\gamma$	Operador de atraso em contagem.
$\mathbb{R}_{max}$	Conjunto $\mathbb{R} \cup \{\varepsilon\}$ , max, +.
$\overline{\mathbb{R}}_{max}$	Dioide completo $(\mathbb{R} \cup \{-\infty, +\infty\}, \text{max}, +)$ .
$\delta$	Operador de atraso no tempo.
$\frac{a}{b} = a \backslash b$	Residuação à esquerda.
$\frac{a}{b} = a / b$	Residuação à direita.
$ \rho $	Comprimento do Caminho.
$ \rho(u, y) $	Comprimento do Caminho entre os vértices $u$ e $y$ .
$\sigma$	Tempo para troca de passageiros entre os trens.
$\lambda$	Autovalor da Matriz $A$ .
$\frac{1}{\lambda}$	Taxa de Produção do Sistema.
$\tau$	Evento.
$\beta$	Taxa de Disparo do Controlador.
$m$	Número de Fichas do Controlador
$E$	Conjunto de Eventos.
$P$	Conjunto de Lugares.
$T$	Conjunto de Transições.
$S$	Semi-anel.
$l, p, n$	Variáveis.
$ST_1, ST_2$	Estações de uma rede ferroviária.
$x(k)$	Função de datadores associados às transições internas.
$k$	Número de disparos da transição.
$A, B$	matrizes do sistema modelado.
$[A]_i, [B]_i$	Vetores linha.
$u(k)$	Função de datadores associados às entradas.
$y(k)$	Função de datadores associados às saídas.
$\Delta_i$	Restrição de tempo de disparo associado a transição $i$ .



## Lista de Figuras

2.1	Rede correspondente a Equação (2.9). . . . .	12
2.2	Rede correspondente a Equação (2.9). . . . .	15
2.3	Rede com entrada e saída. . . . .	18
2.4	Ordem de atividades em um projeto. . . . .	19
2.5	Grafo de transição de uma matriz $5 \times 5$ . . . . .	27
2.6	Grafo de precedência associado à matriz anterior. . . . .	28
2.7	Composição paralela de duas matrizes de transição. . . . .	29
3.1	Comparação de caminhos de amostra para Sistemas Dinâmicos de Variáveis Contínuas (SDVC) (a) e Sistemas de Evento Discretos (b). . . . .	40
3.2	Um sistema de fila. . . . .	41
3.3	Sistema industrial de filas. . . . .	42
3.4	Uma interseção T controlada por semáforos. . . . .	43
3.5	Elementos de uma rede de Petri. . . . .	44
3.6	Rede de Petri com 5 lugares e 6 transições. . . . .	45
3.7	Dois marcações, $x_1$ e $x_2$ , ao gráfico de rede de Petri. . . . .	46
3.8	Sequências de disparos de transições em uma rede de Petri. . . . .	48
3.9	(a) Simples sistema de fila, (b) Modelo de rede de Petri para um sistema de fila simples com estado inicial $[0, 1, 0]$ . (c) Modelo de rede de Petri de um sistema de fila simples com estado inicial $[0, 1, 0]$ depois de disparada a sequência $\{a, s, a, a, c, s, a\}$ [CL99]. . . . .	50
3.10	Trecho de uma rede de Petri ilimitada. . . . .	51
3.11	Rede de Petri com lugar seguro, $P_2$ . . . . .	51
3.12	Modelo de uma rede de Petri dependente do tempo de um SED [CL99]. . . . .	52
3.13	Configurações para fluxo de fichas numa Rede de Petri. . . . .	53
3.14	Exemplo de um GET. . . . .	54
3.15	Simbologia utilizada em um GET com duas transições e um lugar. . . . .	56
3.16	Rede de Petri $P$ -temporizada (GET). . . . .	56
3.17	GET com fichas. . . . .	57
3.18	Um GET com duas entradas e uma saída. . . . .	60
3.19	Composição de Sistema. . . . .	61

3.20	Modelo de um sistema de manufatura com 8 postos de trabalho, 3 entradas de alimentação e 8 restrições entre as transições internas. . . . .	65
4.1	a) Máquina, e b) Estoque. . . . .	72
4.2	Exemplo de numeração das transições em um $GET_{MB}$ . . . . .	72
4.3	Exemplo de um sistema modelado por GET sem controle nas entradas de alimentação. . . . .	74
4.4	Sistema de realimentação para otimizar os estoques de entrada. . . . .	76
4.5	Sistema de manufatura proposto. . . . .	79
4.6	Sistema de realimentação obtido aplicando-se o método proposto. . . . .	81
4.7	Sistema de realimentação obtido aplicando-se o método por modelo de referência [CHBF01]. . . . .	81
4.8	Sistema com realimentação (Modelo de Referência). . . . .	84
4.9	Controlador obtido na aplicação do método proposto. . . . .	85
4.10	Sistema de manufatura MIMO. . . . .	87
4.11	Controle após a implementação do método proposto. . . . .	89
4.12	Sistema de Manufatura ilustrando uma simples sincronização. . . . .	93
4.13	Sistema de realimentação para sincronização de transições. . . . .	94
4.14	Sistema de Manufatura ilustrando uma sincronização de duas transições. . . . .	95
4.15	Atrasos aplicados no sistema para realizar a sincronização. . . . .	97
4.16	Sistema de Manufatura. . . . .	103
4.17	Resultado da aplicação do método de sincronização para duas transições. . . . .	105
4.18	Resultado para sincronização de três transições, $3 \otimes x_3(k) = 4 \otimes x_4(k) = x_5(k)$ . . . . .	107
4.19	Sistema de manufatura. . . . .	108
4.20	Resultado para a Sincronização de Quatros Transições Disparando no mesmo Instante de Tempo. . . . .	110
A.1	Controle por pré-compensação [MMLH05]. . . . .	117
A.2	Problema de estabilidade na pré-compensação [MMLH05]. . . . .	118
A.3	Estabilidade robusta da realimentação [MMLH05]. . . . .	118
A.4	Controle por realimentação [MMLH05]. . . . .	119

# Capítulo 1

## Introdução

Em um processo produtivo, mais de 50% do custo de um produto pode ser atribuído à movimentação e armazenamento de materiais [Her97]. A correta ocupação e utilização do espaço físico de uma empresa têm reflexos importantes na diminuição dos custos da instituição, tornando o sistema de manufatura mais produtivo e mais eficiente [Her97]. O controle de custos da manufatura é sinônimo de diferencial de produtividade, competitividade e crescimento patrimonial. Toda empresa, por mais que se empenhe no controle de seus custos, sempre incorrerá em capital para seu funcionamento [Bla98]. Manufatura é uma das funções que possuem influência determinante no desempenho da empresa, pois para que ela venha a ter sucesso no mercado, deve possuir um sistema de manufatura enxuto e eficiente [Gon01].

As modernas técnicas de pesquisa e novas tecnologias têm produzido sistemas com a finalidade de executar tarefas que, seja pela importância que adquirem em seu contexto, seja por sua complexidade e custo, justificam o esforço despendido na sua otimização e automação. Automação da manufatura, supervisão de tráfego, logística, sistemas operacionais, redes de comunicação de computadores, entre outros, são sistemas que não são tratados adequadamente pelos modelos matemáticos convencionais, baseados em equações diferenciais. Mas, a importância apresentada por estes sistemas faz com que seja altamente desejável encontrar soluções para problemas relacionados ao seu controle [Cur01].

Um estudo de controle de um sistema de manufatura pode ser feito utilizando-se simuladores. Essas ferramentas são indispensáveis na solução de muitos problemas, sendo usadas para descrever e analisar o comportamento de um sistema real, além de responder as perguntas sobre o sistema [Ban99]. Vários sistemas estudados no campo da ciência podem ser modelados e simulados por alguma ferramenta gráfica ou matemática, dentre as quais se incluem linguagens controláveis [RW89], Redes de Petri [Mur89], Redes de Petri Controladas [KH91], Cadeias de Markov [Çin75], Teoria das Filas [Kle75] e álgebra max-plus [BCOQ92].

A modelagem consiste de um modelo que represente o comportamento de algum sistema real ou imaginário, de acordo com condições iniciais estabelecidas. Modelar é o primeiro passo para a análise de um sistema de qualquer natureza. Quando o modelo é uma representação válida de um sistema, informações significativas podem ser retiradas sobre sua dinâmica ou seu desempenho. Então, o modelo pode ser definido como uma representação física, matemática, lógica ou

computacional qualquer de um sistema, processo, fenômeno ou entidade [Tri03].

A Álgebra Max-plus é uma ferramenta matemática utilizada, em conjunto com os GET, na modelagem de sistemas sujeitos a fenômenos de sincronização e de atraso. GET é um caso particular das redes de Petri temporizadas [MAH11]. Por meio do grafo de eventos é possível compreender o comportamento dinâmico do sistema em relação aos vários disparos das transições que podem ocorrer. Conhecer o comportamento temporal de cada transição, isto é, sua trajetória, possibilita definir o comportamento global do sistema, permitindo traçar o seu comportamento futuro em vista dos diversos objetivos traçados para o mesmo. Ainda, é possível determinar a quantidade de disparos de uma transição até um determinado instante de tempo, ou saber qual é o instante de tempo em que ocorrerá o  $n$ -ésimo disparo de um elemento. Esse é um fato que permite, utilizando-se as ferramentas apropriadas, determinar quanto o sistema pode produzir ou quanto ele necessita para poder corresponder às metas traçadas.

Em um GET, o mecanismo de temporização está relacionado com o tempo de permanência das fichas nos lugares da rede. Nele, todos os lugares possuem um único arco de entrada e um único arco de saída [Mur89], [BCOQ92]. O comportamento dinâmico do GET pode ser descrito por meio de equações de datadores, as quais são obtidas de forma recursiva, analisando-se o sistema modelado por GET. Também é possível tratar questões dinâmicas de um sistema de manufatura, tais como: estabilidade, restrições nos estados e no controle, comportamento cíclico e taxa de produção.

Os problemas de controle para GET são usualmente propostos utilizando-se a política de gestão *just-in-time* (JIT), na qual, o objetivo é entregar a matéria-prima no exato momento da necessidade dos postos de trabalho, sem a necessidade de se formar estoques desnecessários no interior da planta fabril, mas atendendo às especificações de demanda de produção. JIT é um sistema de administração de produção que estabelece os calendários de produção de todos os produtos em um sistema produtivo. Tudo o que produzido, comprado, armazenado e transportado deve ser feito dentro do calendário previamente aprovado, nenhuma outra atividade deve ser executada antes da hora exata. É aplicado em vários tipos de organização, tendo como principal objetivo a redução dos estoques e custos decorrentes do mesmo. O JIT é a base para muitas outras políticas de controle de produção. Aplicando-se corretamente esse sistema, a matéria-prima a ser utilizada chega ao local de utilização somente no momento exato em que for necessário. Os produtos somente são fabricados ou entregues a tempo de serem vendidos ou montados. Operacionalmente, basta dizer que JIT significa que cada processo deve ser suprido com os itens e quantidades certas, no tempo e lugar certo [Ghi95]. Nas fábricas onde o JIT está implementado, o estoque de matéria-prima é mínimo e suficiente para poucas horas de produção. Nesses sistemas produtivos a principais funções consistem em planejar as necessidades futuras de capacidade, planejar os materiais comprados, planejar níveis apropriados de estoques, programar atividades de produção, ser capaz de saber da situação correta, das pessoas, equipamentos e materiais em um sistema produtivo, ser capaz de reagir eficazmente, ser capaz de prometer prazos e outras funções. Essas atividades afetam os níveis de desempenho do sistema produtivo em termos de custo, qualidade, prazo, confiabilidade e flexibilidade e por consequência a competitividade da empresa no mercado onde atua [CG93]. De um modo geral, as

propostas de controle estão baseadas no controle em malha aberta, utilizando-se a pré-compensação, e em malha fechada, utilizando-se a realimentação [Mai03].

A motivação para o desenvolvimento deste trabalho surge da necessidade do desenvolvimento de métodos matemáticos apropriados para encontrar soluções para o problema do controle para sistemas modelados por Grafos de Eventos Temporizados (GET), Sistemas a Eventos Discretos (SED). O primeiro controle visa o controle do fluxo de entrada, e o segundo, além do controle do fluxo de entrada, permite a sincronização de atividades dentro do processo produtivo. Esses controles permitem o uso racional dos espaços e evitam perdas no processo produtivo, pois a matéria prima só é entregue ao posto de trabalho no momento exato que o posto necessita, evitando estoque desnecessário dentro do sistema. Embora esses sistemas não sejam tratados adequadamente pelos modelos matemáticos convencionais, eles são apropriadamente modelados pelos GET. Isso deve-se ao fato de que, no uso do GET esses sistemas podem ser descritos dinamicamente por equações lineares recursivas utilizando-se as operações de maximização e de soma da álgebra max-plus.

Neste trabalho foram modelados, analisados e simulados sistemas que são denominados como “Sistemas de Manufatura”. O objetivo é contribuir com métodos que permitem o controle do fluxo de entrada de sistemas modelados por grafos de eventos temporizados. A ferramenta matemática utilizada é a álgebra max-plus.

Uma breve revisão da literatura sobre os últimos trabalhos publicados que estão relacionados com o tema são apresentados a seguir.

## 1.1 Revisão da Literatura

Existem na literatura alguns trabalhos relacionados ao controle de sistemas que podem ser modelados por grafo de eventos em conjunto com a álgebra max-plus. Dentre esses, alguns tratam da modelagem de sistemas de tráfego [vEO98] [FGS04] [LC04] [Gov07] [Gar07] [Kat07] [BM91], que são sistemas sujeitos a restrições de sincronização. Outros enfocam na modelagem de sistemas de manufatura, e a análise pode ser estendida a todos os sistemas que podem ser modelados como sistemas a eventos discretos, que são caracterizados por fenômenos de sincronização e atraso de tempo.

Como já foi dito, os problemas de controle para GET são usualmente propostos com o objetivo de minimizar os estoques do sistema atendendo as especificações de demanda. Essas propostas de controle estão baseadas no controle em malha aberta, utilizando-se a pré-compensação, e em malha fechada, utilizando-se a realimentação [Mai03].

Moody [MYLA94] enfoca o problema de restrições de funcionamento que são representadas na forma de desigualdades lineares. Apresenta um controlador que força o sistema a respeitar as restrições impostas a ele. Esse controlador é também uma rede de Petri e é obtido transformando o conjunto de restrições em invariantes de lugar do sistema controlado. Caso o sistema possua transições não-controláveis, pode não ser possível forçar diretamente a obediência de uma dada restrição, que é então dita não-admissível.

Boimond e Ferrier [BF96] propõem uma estrutura de controle interno, utilizado na teoria de controle convencional, para projetar um controlador que trabalha como o modelo inverso do sistema.



O controlador proposto trabalha com saídas pré-definidas, modificando a entrada de referência para cada demanda proposta. O controle é dividido em dois blocos, um chamado de Previsão e outro de Inversão. O modelo inverso é obtido por residuação. Um ponto importante deste controle é a robustez da estrutura de controle interno.

Conttenceau et al. [CHBF99] tratam de um sistema de realimentação que atrasa a entrada do sistema ao máximo, mantendo a mesma relação de transferência entre a entrada e a saída. É calculado o maior controle de realimentação possível sem diminuir o desempenho do sistema, com a finalidade de atrasar, tanto quanto possível, a entrada de fichas no sistema. Estes sistemas funcionam com realimentação de informações da saída para a entrada, utilizando como referência um sistema de controle Kanban, onde observa-se que a maior realimentação é obtida por residuação.

Menguy [MBHF00] aborda o controle *just-in-time* para sistemas com entradas controladas ou não-controladas. A solução de controle proposta prevê o controle da mudança da demanda de produção para entradas controladas, modificando as produções previstas, e prevê o controle de matérias-primas para entradas não-controladas. Esse controle é feito no fornecimento de matéria-prima, tal solução é encontrada utilizando-se teoria da residuação.

Schutter et al. [SvdB00] adotam uma estratégia de controle baseada na abordagem por modelo de referência. Um problema de controle é proposto e o que se deseja obter é o “melhor” controle possível de maneira a garantir que a saída do sistema siga uma determinada trajetória de referência. Esse problema de controle está relacionado com a estratégia JIT de planejamento de produção. Um sistema de manufatura é tomado como exemplo, e uma trajetória, como sendo uma referência para o sistema, é dada. O objetivo do controlador é retardar ao máximo a entrada de matérias-primas no processo, de modo a evitar a formação de estoques no interior da planta, mas atendendo à demanda pré-especificada pela taxa de produção de referência. A solução desse problema foi genericamente denominada de controle ótimo, no sentido de que o “melhor” controle possível é aquele que atrasa ao máximo o disparo dos eventos de entrada do sistema. Uma solução foi encontrada utilizando-se o Modelo de Controle Preditivo (MCP), uma das principais vantagens deste, é que pode acomodar restrições sobre as entradas e saídas do sistema, normalmente usando modelos lineares em tempo discreto. Neste trabalho, o MCP é aplicado para uma classe de sistemas a eventos discretos que podem ser modelados como lineares pela álgebra max-plus. Em geral, o problema de otimização é não-linear e não-convexo. O MCP usa o horizonte de controle finito e permite a inclusão de restrições adicionais, podendo lidar com mudanças estruturais e mudanças nos parâmetros do sistema, incluindo sensores e atuadores de falhas. No MCP, o índice de desempenho, ou critério de custo, ou função objetivo, é a soma direta entre o erro de referência e o esforço de controle. O MCP usa o horizonte de controle para reduzir o número de variáveis de otimização, isso resulta numa diminuição da carga computacional. O modelo é atualizado constantemente por estimativas geradas, e novas medições são feitas passo-a-passo por substituições sucessivas das equações de controle.

Conttenceau et al. [CHBF01] tratam da síntese de controle de realimentação de GET. Esses controladores são possíveis no âmbito da gestão da produção e podem representar uma classe de

sistemas de produção que apresentam apenas fenômenos de sincronização e atrasos, permitindo modificar a dinâmica de um sistema (linha de produção ou oficina de fabricação) de acordo com um modelo de referência dado, retardando ao máximo a liberação de peças para o sistema, de acordo com a política de produção *just-in-time*.

Em Maia et al. [MHMC03] diversas estratégias de controle baseadas na abordagem por modelo de referência são apresentadas. Um problema de controle é proposto e o que se deseja obter é o “melhor” controle possível de maneira a garantir que a saída do sistema siga uma determinada trajetória de referência. O autor propõe uma estrutura de controle baseada em uma pré-compensação e na realimentação. Apresenta ainda um novo método de cálculo utilizando o modelo de referência. A estrutura é baseada na pré-compensação bem como na realimentação, e garante uma solução ótima para a estratégia JIT de planejamento de produção.

Qi Zhu et al. [ZSX04] visam desenvolver um método de agendamento de tarefas para uma célula de manufatura flexível por meio da fusão das redes de Petri temporizadas e da álgebra max-plus. A modelagem envolve parâmetros de decisões e situações de falhas das máquinas, bem como enfatiza a necessidade do agendamento das tarefas para obter-se a melhor eficiência produtiva. Essa eficiência é obtida em função das sequências de atividades propostas, além de evitar problemas de decisão. Os autores enfocam a vantagem do uso das redes de Petri que podem representar de forma concisa as restrições do sistema, com regras de disparo das transições, tornando a rede de Petri uma ferramenta bastante útil na formulação dos problemas de escalonamento com a flexibilidade de roteamento e recursos compartilhados. Também apresenta uma simples célula de manufatura, uma célula de fabricação flexível que consiste de dois robôs, dois tipos diferentes de máquinas CNC (fresa e torno) e um transportador. O sistema pode processar dois tipos de matéria-prima (blocos e cilindros). Várias são as operações entre as composições: (a) pegar a matéria-prima e colocar no transportador; (b) colocar a peça no tipo de máquina específica; (c) processar a peça; (d) armazenar a peça. Três experiências são propostas: (1) com agendamento fixo, (2) tempo real e, (3) com adaptação em tempo real. Na experiência com tempo real é introduzida uma situação de falha, e somente após a falha ser removida é que o sistema volta a funcionar normalmente. Dependendo do tempo de falha e em que máquina ela ocorre, ela poderá ou não afetar o tempo final de processo. Na situação de adaptação em tempo real é introduzida a situação de conflito, enfrentada pelo robô 2. As situações de decisão são analisadas para verificar qual seria a melhor escolha a ser feita comparando-se os tempos finais de cada tarefa.

Brunsch et al. [BRH12] tratam da modelagem e controle de sistemas HST (*high-throughput screening*), calculando um modelo de controlador para lidar com desvios inesperados da operação cíclica pré-determinada durante a execução do sistema. Enfatizam que a única possibilidade de controle é retardar os eventos de início de atividades. É ilustrado como um controlador de realimentação pode ser calculado, usando-se a teoria da residuação. A estratégia de controle é fornecer uma solução ótima no contexto just-in-time, ou seja, atrasar o início das atividades, tanto quanto possível, sem diminuir o rendimento do sistema.

Kim e Lee [KL12] propuseram uma maneira para projetar controles robustos que impedem uma

*cluster tool* de violar tempos de restrições. Fazem o uso de grafos de eventos temporizados e da álgebra max-plus para descrever um *cluster tool system*. Um *cluster tool* pode ser definido como um ambiente fechado que consiste de processamento em módulos, bloqueios de carga e um robô para transporte de peças. Existem vários tipos de *cluster tools* de acordo com o layout dos Processos Modulares e o número de braços do robô.

Amari et al. [ADLM12] tratam da verificação de algumas condições temporais. A verificação de restrições de tempo é formulada em termos de um problema de controle, assumindo que algumas entradas do processo podem ser controladas. O comportamento da planta é modelado com equações max-plus, e as restrições temporais são representadas por desigualdades lineares. O autor propõe um método para a síntese da lei de controle que permite conhecer um determinado conjunto de restrições de tempo. A lei de controle resultante em si é finalmente definida como um conjunto de equações de diferenças lineares em max-plus, envolvendo um número finito de atrasos. Esta lei de controle é causal e pode ser implementada on-line, para um conjunto conhecido de estados do sistema. Tal equação, corresponde a um sistema de realimentação, que é representado por um GET. No entanto, não tem por objetivo calcular os atrasos máximos que as entradas de alimentação podem ter sem atrasar a saída do sistema. O resultado apresentado não possui a máxima produtividade, pois, o sistema de realimentação não garante suficiente número de fichas para manter um fluxo produtivo.

Dias et al. [DML13] propõem um método para calcular os parâmetros de um controlador para controlar o fluxo de entrada de sistemas modelados por grafos de eventos temporizados em sistema com múltiplas entradas e múltiplas saídas, esse é um dos trabalhos que apresentam resultados prévios das propostas desta tese.

David-Henriet et al. [DHHRC14] e [DHHRC13] lidam com um sistema max-plus com sincronização parcial, esta sincronização parcial é um sistema a eventos discretos persistente dividido em dois subsistemas, um subsistema principal e um subsistema secundário, de modo que o subsistema secundário tem de adaptar o seu comportamento ao subsistema principal, mas o subsistema principal não é afetado pelo subsistema secundário. Um modelo formal baseado em equações max-plus recursivas é apresentado para sistemas com sincronização parcial. Além disso, a saída do sistema possui prioridade mais elevada. O desempenho do subsistema principal nunca é degradado para melhorar o desempenho do subsistema secundário. Um sistema a eventos discretos é dito ser persistente se a ocorrência de um evento nunca desativa o outro evento, é o caso em que duas transições diferentes são ativadas pelo mesmo conjunto de condições, quando uma transição dispara a outra ainda permanece habilitada para o disparo. Tais sistemas podem aparecer depois de solucionados os conflitos entre os eventos por uma programação predefinida nas áreas de processos de manufatura ou de redes de transporte. Nesses trabalhos os resultados mostram apenas pouca sincronização entre as saídas e entradas dos dois sistemas de interesse, subsistema principal e subsistema secundário, já que o objetivo não é uma sincronização total. Também não é apresentada qualquer forma de sincronização, dentro de um mesmo sistema entre suas transições internas. Na proposta desta tese, o Método II oferece uma forma de sincronização total entre as transições internas de um sistema

principal.

Cottenceau et al. [CLH14] tratam do GET ponderados. Desenvolvem a teoria max-plus que permite descrever o comportamento para os sistemas que variam no tempo. Uma classe de GET *First-In First-Out* (o primeiro que entra é o primeiro que sai) é modelada por GET ponderados, para os quais existe um modelo de entrada e saída.

Seleim e ElMaraghy [SE14] apresentam um método para geração fácil e rápida das equações max-plus para linhas de fluxo de sistemas produtivos de qualquer tamanho ou estrutura. As equações geradas podem modelar as linhas com tamanho de estoque finito ou infinito. Para a modelagem da linha de produção, os estoques são inicialmente assumidos como sendo infinito em todas as estações. As equações do modelo das linhas de produção são geradas como sendo um conjunto de estações em série, que se fundem após a identificação das diferentes fases, usando uma matriz de adjacência para a linha de fluxo. Nas equações geradas, a dinâmica do sistema é capturada nas duas matrizes que são função do tempo de processamento das diferentes estações da linha. O método é intuitivo e o código de software é de fácil compreensão, portanto, pode facilitar a rápida análise de diferentes configurações de linhas de fluxo de produção e avaliação de cenários hipotéticos. O método baseia-se na observação de características diferentes que afetam exclusivamente as equações finais. Estas características podem ser adicionadas sequencialmente, de modo a formar o sistema de equações finais. A correção de todas as equações geradas é verificada comparando-se os resultados com modelos de simulação de cada um dos sistemas apresentados. A primeira hipótese é realista para sistemas automatizados, bem como para os sistemas semi-automatizados de manuseamento de materiais paletizados, onde a variação do tempo de processo é muito menor do que o tempo de processamento e, portanto, podem ser negligenciadas. A segunda suposição também é realista, quando se estuda o funcionamento normal de um sistema de curto prazo. O objetivo é compreender e otimizar o comportamento do sistema, ao invés de se estudar uma operação de longo prazo para verificar a capacidade de planejamento do mesmo, onde as falhas das máquinas teriam um efeito significativo.

## 1.2 Objetivos

O objetivo desta tese é propor dois métodos de controle para sistemas modelados por grafos de eventos temporizados representados por equações de datadores, para:

- Solucionar o problema do controle do fluxo de entrada de sistemas modelados por grafos de eventos temporizados; e,
- Sincronizar o disparo de transições para sistemas modelados por grafos de eventos temporizados.

O Primeiro método de controle propõe o cálculo dos parâmetros de um controlador de realimentação, que atrasa ao máximo a entrada de matéria-prima no sistema modelado por GET, calculando, também, o número mínimo de fichas no sistema de realimentação, limitando ao máximo as fichas no sistema modelado. Os parâmetros desse controlador proporcionam a definição de uma

lei de controle, que é o resultado dos parâmetros do controlador adicionados na modelagem matemática do sistema.

Um segundo método de controle propõe a sincronização do disparo de várias transições internas em um sistema modelado por GET. Esse controle é obtido atrasando-se a entrada de matéria-prima no sistema e, na maioria dos casos, modificando-se a taxa de produção do sistema. São propostas duas equações gerais para o cálculo dos parâmetros de um controlador que sincroniza o disparo dessas várias transições. Esses parâmetros são os tempos de atraso e a taxa de alimentação do sistema. O cálculo do número de fichas de realimentação é obtido da mesma forma que no primeiro método. Esse tipo de controle é importante, pois sincronizar transições permite definir o momento exato que um evento, ou atividade, pode começar ou terminar numa sequência de atividades definidas. Os resultados desse método são apresentados na sincronização de duas, três e quatro transições.

Os resultados apresentados nesta tese reduzem a complexidade para o cálculo dos controladores para esses dois tipos de controle. Isso resulta numa diminuição da carga computacional, e em geral proporcionam um efeito de estabilização do sistema, evitando o *overflow* dos estoques, usando como base a política de gestão JIT.

### 1.3 Organização do Trabalho

No Capítulo 2 é tratado o assunto da álgebra max-plus com as principais definições e operações nessa álgebra. A relação entre grafos e matrizes é estudada juntamente com a teoria espectral de matrizes. Também, neste capítulo são vistas as soluções de equações na álgebra max-plus, bem como a apresentação de uma breve descrição sobre residuação.

No Capítulo 3 é feita uma revisão dos principais conceitos e dos fundamentos da teoria de sistemas a eventos discretos, com a introdução dos conceitos de evento, sistemas dirigidos pelo tempo e sistemas baseados em eventos, tratados em comparação aos sistemas dinâmicos de variáveis contínuas; são apresentados os fundamentos de redes de Petri, as equações de estado e as dinâmicas da rede de Petri para uma classe especial de redes. É introduzida a estrutura de temporização nas redes de Petri, gerando as redes de Petri temporizadas, e são apresentados alguns exemplos de aplicações. Também, são tratados os grafos de eventos temporizados, é iniciado o estudo sobre grafos de eventos, grafos associados com matrizes, e a representação de sistemas por meio de datadores e por matrizes de transferência.

No Capítulo 4 são desenvolvidos os métodos de controle do fluxo de entrada para GET utilizando-se um controlador. O primeiro método calcula o parâmetros do controlador para atrasar o máximo possível a entrada de matéria-prima no sistema produtivo. O segundo método calcula os parâmetros do controlador para realizar a sincronização dos disparos de transições internas do sistema. Também são apresentados os resultados da aplicação dos métodos para vários sistemas.

Finalmente, no Capítulo 5 é feita a conclusão desta tese. As vantagens do emprego das ferramentas, o grau de importância dos resultados obtidos e os possíveis desdobramentos em termos de pesquisas futuras são abordados.

## Capítulo 2

# Álgebra Max-plus

A álgebra max-plus é uma ferramenta matemática com muitas aplicações práticas em vários sistemas existentes, por exemplo: sincronização de sistemas de transporte em linhas ferroviárias, controle de alimentação de processos produtivos e controle de semáforos. Essa ferramenta é aplicada na modelagem de sistemas sujeitos a fenômenos de sincronização e de atraso de tempo.

Várias propriedades de SED podem ser expressas em termos de teoria de grafos utilizando a álgebra max-plus. Verifica-se também a existência de uma relação poderosa entre grafos e matrizes nessa álgebra, sobre a qual muitos estudos tem sido publicados, assim sendo, destacar-se-ão algumas dessas relações ao longo deste capítulo.

Ao mesmo tempo, à álgebra max-plus será atribuído um tratamento formal, onde seus conceitos básicos e propriedades algébricas serão abordados, fazendo-se a descrição de matrizes e vetores nela aplicados.

A teoria espectral de matrizes é um dos focos de estudo, no qual serão tratados os assuntos sobre ciclo médio, autovalores e autovetores.

Este capítulo também propõe-se a abordar a solução de equações na álgebra max-plus, bem como a teoria da residuação.

Para uma melhor estruturação e esclarecimento do tema, o capítulo é distribuído da seguinte forma: na Seção 2.2 é tratado o assunto da álgebra max-plus com as principais definições de operações desta álgebra; na Seção 2.5 a teoria espectral de matrizes é tratada juntamente com o estudo de grafos associados com matrizes, bem como a solução da equação implícita  $x = ax \oplus b$ . Na Seção 2.6 é apresentada uma breve descrição sobre a teoria da residuação. Na Seção 2.7 é feita a conclusão do capítulo.

### 2.1 Breve Histórico

A álgebra max-plus é um dos muitos semi-anéis idempotentes que são considerados em vários campos da matemática. Ela tem encontrado aplicações em muitas áreas, tais como análise combinatória, otimização, física-matemática e geometria algébrica [HS07]. Também é usada em teoria de controle, agendamento de tarefas em máquina, processos de eventos discretos, fabricação de sistemas, redes de telecomunicações, sistemas de processamento paralelo e controle de semáforos [Gau97] [CG79]. Muitas equações que são usadas para descrever o comportamento destas aplicações são não-lineares na álgebra convencional, mas tornam-se lineares na álgebra max-plus, sendo essa

uma das razões para sua aplicação em diversas áreas.

Muitos dos teoremas e técnicas usadas na clássica álgebra linear tem análogos no semi-anel max-plus. Cuninghame-Green e vários outros pesquisadores tem dedicado muito tempo criando grande parte da teoria da álgebra max-plus que existe. Uma considerável quantidade desses resultados podem ser encontrados em [CG79], tais como: estudos e conceitos para solucionar sistemas de equações lineares, problemas de autovalor e independência linear no sentido max-plus. A seguir serão introduzidos o conceito básico de álgebra max-plus e as definições das operações nessa álgebra, mais adiante, objetivando ilustrar a utilidade dessa álgebra, será desenvolvido um exemplo ilustrativo de dessa álgebra com aplicação em uma simples rede ferroviária entre duas cidades.

## 2.2 Álgebra Max-Plus

As definições, propriedades e os teoremas da álgebra max-plus foram extraídos de Baccelli et al. [BCOQ92], Cohen [Coh01] e Heidergott et al. [HOvdW06].

### 2.2.1 Conceitos Básicos e Definições

Também conhecida como álgebra de dioides, refere-se ao fato de que esta álgebra é baseada em duas operações, nomeadas formalmente de adição e multiplicação e denotadas por  $\oplus$  e  $\otimes$ . No entanto, o seu significado real (em termos de álgebra normal) é diferente e, para começar o desenvolvimento dessa álgebra, algumas definições básicas são importantes:

Define-se  $\varepsilon = -\infty$  e  $e = 0$ , também será denotado  $\mathbb{R}_{max}$  o conjunto  $\mathbb{R} \cup \{\varepsilon\}$ , onde  $\mathbb{R}$  é o conjunto de números reais. Para dois elementos  $a$  e  $b \in \mathbb{R}_{max}$ , serão definidas as operações  $\oplus$  e  $\otimes$ , de forma que

$$a \oplus b = \max(a, b) \quad \text{e} \quad a \otimes b = a + b. \quad (2.1)$$

Claramente,

$$\max(a, -\infty) = \max(-\infty, a) = a, \text{ e}$$

$$a + (-\infty) = -\infty + a = -\infty,$$

para qualquer  $a \in \mathbb{R}_{max}$ , de modo que

$$a \oplus \varepsilon = \varepsilon \oplus a = a \quad \text{e} \quad a \otimes \varepsilon = \varepsilon \otimes a = \varepsilon, \quad (2.2)$$

para qualquer  $a \in \mathbb{R}_{max}$ . As seguintes definições são ilustradas com exemplos numéricos, como segue:

$$5 \oplus 3 = \max(5, 3) = 5,$$

$$5 \oplus \varepsilon = \max(5, -\infty) = 5,$$

$$5 \otimes \varepsilon = 5 + (-\infty) = -\infty = \varepsilon,$$

$$e \oplus 3 = \max(0, 3) = 3,$$

$$5 \otimes 3 = 5 + 3 = 8.$$

O conjunto  $\mathbb{R}_{max}$  juntamente com as operações  $\oplus$  e  $\otimes$  são chamados de álgebra max-plus, e denotados por

$$\mathcal{R}_{max} = (\mathbb{R}_{max}, \oplus, \otimes, \varepsilon, e).$$

Assim, como na álgebra convencional, a operação  $\otimes$  tem prioridade sobre a operação  $\oplus$ . Por exemplo,

$$5 \otimes -9 \oplus 7 \otimes 1$$

tem de ser compreendido como

$$(5 \otimes -9) \oplus (7 \otimes 1).$$

Nota-se que  $(5 \otimes -9) \oplus (7 \otimes 1) = 8$ , enquanto que  $5 \otimes (-9 \oplus 7) \otimes 1 = 13$ . As operações  $\oplus$  e  $\otimes$  possuem propriedades algébricas interessantes. Por exemplo, para  $a, b$  e  $c \in \mathbb{R}_{max}$ , é correto afirmar que

$$\begin{aligned} a \otimes (b \oplus c) &= a + \max(b, c), \\ &= \max(a + b, a + c), \\ &= (a \otimes b) \oplus (a \otimes c). \end{aligned}$$

### 2.3 Exemplo Ilustrativo

Considere uma rede ferroviária entre duas cidades, cada cidade com uma estação, como indicado na Figura 2.1. Essas estações são denominados  $ST_1$  e  $ST_2$ , respectivamente, e estão ligados por duas linhas ferroviárias. Um trilho vai de  $ST_1$  para  $ST_2$ , e o tempo de viagem para um trem ao longo dessa linha é assumido como sendo de 3 unidades de tempo. A outra linha vai de  $ST_2$  para  $ST_1$ , e o tempo de viagem ao longo dessa outra linha dura 5 unidades de tempo. Em conjunto, essas duas linhas ferroviárias formam um circuito. Os trens vindos de  $ST_1$  para  $ST_2$  devem retornar para  $ST_1$  em outra faixa, e os trens que partem de  $ST_2$ , depois de terem visitado  $ST_1$ , voltam para  $ST_2$ . Além dessas duas linhas, outras duas linhas existem, na forma de circuitos, ligando os subúrbios das cidades com a sua respectiva estação. Uma ida e volta ao longo dessas linhas dura 2 unidades de tempo para o trem no trajeto entre a cidade e a estação  $ST_1$ , e 3 unidades de tempo no trajeto da cidade para a estação  $ST_2$ . Evidentemente, existem estações locais nestes subúrbios, mas elas não desempenham nenhum papel importante para equacionar o problema, então não serão indicadas. O objetivo é projetar um calendário com os seguintes critérios:

- a) Que os tempos de viagem dos trens ao longo de cada uma das linhas seja fixo;
- b) A frequência de saída dos trens deva ser tão alta quanto possível;
- c) A frequência dos trens deva ser a mesma ao longo de todas as quatro linhas, produzindo um calendário com horários regulares de partida;
- d) O trem que chegar a uma estação precisa esperar pelo outro, a fim de permitir a troca de passageiros nos trens;
- e) Os trens em uma estação devem partir assim que for permitido.

No modelo apresentado, é definido um número total de quatro trens, um trem em cada um dos circuitos externos e dois trens nos circuitos internos. Os instantes de partida dos dois trens na



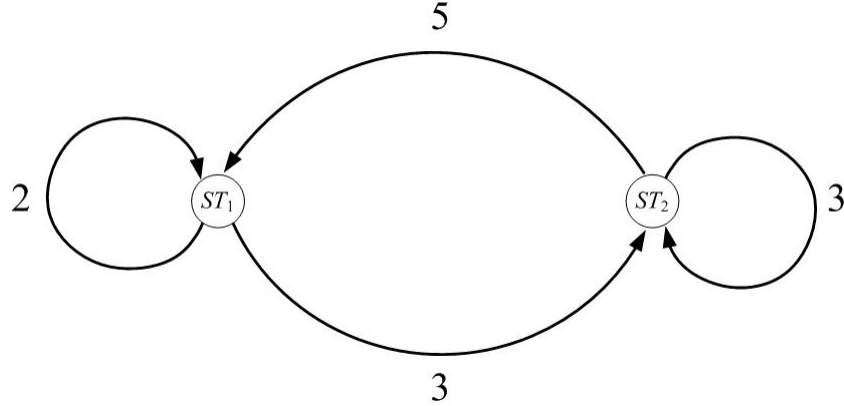


Figura 2.1: Rede correspondente a Equação (2.9).

estação  $ST_1$ , um em direção a  $ST_2$  e outro para os subúrbios, serão indicados por  $x_1$ . Estes dois trens partem ao mesmo tempo, tão logo seja possível, após as mudanças de passageiros entre os trens. Da mesma forma,  $x_2$  são os instantes de partida dos dois trens em  $ST_2$ . Juntos, os horários de partida são escritos como um vetor  $x \in \mathbb{R}$ . Os primeiros horários de partida durante o dia, no início da manhã, serão dados por  $x(0)$ . Depois, os trens deixam as estações nos instantes de tempo indicados pelos dois elementos do vetor  $x(1)$  e assim por diante. Os horários de partida do  $k$ -th trem são indicados por  $x(k-1)$ , tais partidas são chamadas de *eventos* no modelo. Por causa das normas estabelecidas, segue-se que

$$x_1(k+1) \geq x_1(k) + a_{11} + \sigma, \quad (2.3)$$

$$x_2(k+1) \geq x_2(k) + a_{12} + \sigma. \quad (2.4)$$

Os valores de  $a_{ij}$  denotam o tempo de percurso a partir da estação  $ST_j$  para a estação indicada por  $ST_i$ , e  $\sigma$  denota o tempo destinado para os passageiros efetuarem a mudança de um trem para o outro. Sem perda de generalidade,  $\sigma$ , que é o tempo de transferência dos passageiros entre os trens, pode ser adicionado ao tempo total do percurso de cada trem, então,  $a_{11} = 2$  e  $a_{12} = 5$ , da mesma forma, os instantes de partida em  $ST_2$ , assim tem-se

$$x_1(k+1) \geq \max(x_1(k) + 2, x_2(k) + 5),$$

$$x_2(k+1) \geq \max(x_1(k) + 3, x_2(k) + 3).$$

Visto que a frequência das partidas deve ser tão alta quanto possível, e os trens partem o mais rápido possível, as desigualdades nas duas últimas expressões irá, de fato, ter que ser igualdades, o que leva a

$$x_1(k+1) = \max(x_1(k) + 2, x_2(k) + 5),$$

$$x_2(k+1) = \max(x_1(k) + 3, x_2(k) + 3).$$

Assim, se os instantes iniciais de partida,  $x(0)$ , são dados, todos os instantes de partida futuros podem ser determinados de uma única forma. Se, por exemplo,  $x_1(0) = x_2(0) = 0$ , então, a sequência  $x(k)$ , para  $k = 0, 1, 2, 3, 4, \dots$ , será

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 5 \\ 3 \end{pmatrix}, \begin{pmatrix} 8 \\ 8 \end{pmatrix}, \begin{pmatrix} 13 \\ 11 \end{pmatrix}, \begin{pmatrix} 16 \\ 16 \end{pmatrix}, \dots$$

Uma nova sequência é obtida quando um dos tempos iniciais de partida é modificado para  $x_1(0) = 1$  e  $x_2(0) = 0$  (ou seja, os primeiros trens em  $ST_2$ , um em cada sentido, deixam a estação no instante de tempo 0, mas os primeiros trens em  $ST_1$  deixam a estação no tempo 1),

$$\begin{pmatrix} 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 5 \\ 4 \end{pmatrix}, \begin{pmatrix} 9 \\ 8 \end{pmatrix}, \begin{pmatrix} 13 \\ 12 \end{pmatrix}, \begin{pmatrix} 17 \\ 16 \end{pmatrix}, \dots$$

Comparando as duas sequências anteriores, com o tempo entre partidas sendo definido como a duração de tempo entre duas partidas subsequentes ao longo da mesma pista, a segunda sequência tem o mesmo tempo médio entre partidas, que é igual a 4. Ao passo que a primeira sequência tem apenas em média (ou média dos tempos entre partidas) os tempos iguais a 3 e 5. Se estas sequências de partidas forem em horários reais, então a maioria das pessoas vai preferir o segundo calendário, uma vez que ele apresenta uma regularidade.

Uma pergunta que pode surgir, refere-se à possibilidade de obter-se um calendário “mais rápido” (ou seja, um calendário com um menor tempo médio entre partidas) escolhendo-se horários iniciais de partida adequados. A resposta é não. A razão é que o tempo de duração de um trem no circuito interno é igual a 8, e existem dois trens neste circuito. Assim, o tempo médio entre partidas nunca pode ser inferior a  $8/2 = 4$ .

A idéia desenvolvida aqui é que existe uma álgebra que modela um sistema a eventos discretos (SED) que não possui concorrência de recursos e pode naturalmente ser trabalhada como um modelo de sistema linear. O exemplo do trem mostra que a operação de maximização é essencial para obter o fenômeno de sincronização, operando nos tempos de chegada para calcular os horários de partida.

O trem somente terá sua partida liberada quando todos os recursos estiverem disponíveis, sugerindo uma adição de recursos. Neste sentido, convencionalmente “max” será denotado por  $\oplus$ . Neste mesmo exemplo, indica-se a necessidade de conhecer o instante de partida do trem, o tempo de trajeto e outros dados de interesse. A fim de capturar tais informações será convencionalmente a operação de “adição” da álgebra convencional em multiplicação nesta nova álgebra, denotada por  $\otimes$ . As operações  $\oplus$  e  $\otimes$  desempenham operações fundamentais para compreensão da dinâmica dos sistemas modelados por regras de sincronização.

Para tratar da modelagem de sistemas na álgebra, ilustrada anteriormente, uma analogia com as equações de estado, partindo-se da equação mais conhecida em modelagem de sistemas, é feita,

$$x(t+1) = Ax(t), \quad t = 0, 1, 2, \dots \quad (2.5)$$

O vetor  $x \in \mathbb{R}^n$  representa o estado de um modelo básico e este estado evolui com o tempo de acordo com esta equação;  $x(t)$  denota o estado no tempo  $t$ . A matriz  $A$  de tamanho  $n \times n$  representa os parâmetros do sistema.

Se uma condição inicial

$$x(0) = x_0 \quad (2.6)$$

é dada, então a evolução da equação é dada por (2.5). Esta equação é uma equação vetorial. Quando escrita na forma escalar obtêm-se

$$x_i(t+1) = \sum_{j=1}^n A_{ij}x_j(t), \quad i = 1, \dots, n; \quad t = 0, 1, \dots \quad (2.7)$$

O símbolo  $x_i$  denota o  $i$ -ésimo termo do vetor  $x$ ; os elementos  $A_{ij}$  são as entradas da matriz quadrada  $A$ . Se  $A_{ij}$  para  $i = 1, \dots, n, j = 1, \dots, n$  e  $x_j(t)$  para  $j = 1, \dots, n$ , são dados, então  $x_j(t+1)$ , para  $j = 1, \dots, n$ , pode ser calculado de acordo (2.7).

As únicas operações usadas em (2.7) são multiplicação ( $A_{ij} \times x_j(t)$ ) e adição ( $\sum$ ). Suponha que as duas operações de (2.7) são alteradas de forma que a adição torne-se uma “maximização” e multiplicação uma “adição”. Essa modificação é feita com o objetivo de modelar os fenômenos de sincronização e atraso de tempo. Então a Equação (2.7) pode ser reescrita como

$$\begin{aligned} x_i(k+1) &= \max(A_{i1} + x_1(k), A_{i2} + x_2(k), \dots, A_{in} + x_n(k)) \\ &= \max_j(A_{ij} + x_j(k)), \quad i = 1, \dots, n. \end{aligned} \quad (2.8)$$

Se a condição inicial (2.6) também é válida para (2.8), então a evolução temporal de (2.8) é novamente completamente determinada. É claro que as evoluções de (2.7) e (2.8) serão diferentes. A equação (2.8), tal como está, é uma equação não-linear. Por exemplo, dados  $n = 2$  e  $A$  uma matriz  $2 \times 2$ . Supondo

$$A = \begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix}, \quad (2.9)$$

e que a condição inicial seja

$$x_0 = \begin{pmatrix} 1 \\ 0 \end{pmatrix}. \quad (2.10)$$

A evolução de (2.5) é dada por

$$x(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x(1) = \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \quad x(2) = \begin{pmatrix} 23 \\ 14 \end{pmatrix}, \quad x(3) = \begin{pmatrix} 167 \\ 102 \end{pmatrix}, \dots$$

e a evolução de (2.8) é

$$x(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x(1) = \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \quad x(2) = \begin{pmatrix} 11 \\ 9 \end{pmatrix}, \quad x(3) = \begin{pmatrix} 16 \\ 13 \end{pmatrix}, \dots \quad (2.11)$$

O argumento  $t$  em  $x(t)$  é sempre pensado como referindo-se ao tempo; no tempo  $t$  o estado é  $x(t)$ . Com relação à equação (2.8) será apresentado um significado diferente para este argumento. Com o objetivo de salientar o significado diferente, o argumento  $t$  foi substituído por  $k$ . Para entender a mudança da variável, considere uma rede que consiste de uma série de nós e arcos conectando esses nós. A rede correspondente à equação (2.8) possui  $n$  nós, um para cada componente  $x_i$ . Os valores de  $A_{ij}$  correspondem aos arcos que partem do nó  $j$  para o nó  $i$ . Em termos de teoria dos grafos, tal rede é chamada de “grafo direcionado” ou “grafo dirigido”, porque os arcos individuais entre os nós são setas unidirecionais. Portanto, os arcos correspondentes a  $A_{ij}$  e  $A_{ji}$ , ambos existindo, são considerados diferentes.

Os nós da rede podem significar atividades (operações) onde cada nó tem seu próprio tipo de atividade. Estas, por sua vez, possuem um tempo finito, chamado de tempo de atividade ou tempo de realização da operação. Esses tempos de atividades, no entanto, podem ser diferentes para diferentes nós. Supõe-se, a partir de então, que uma atividade em um determinado nó só pode começar quando todas as atividades anteriores (na seqüência de setas indicando o início da atividade) tenham sido executadas, e os resultados destas, enviados ao longo dos arcos para o nó atual. Assim, o arco correspondente a  $A_{ij}$  pode ser interpretado como um canal de saída para o nó  $j$  e, simultaneamente, como um canal de entrada para o nó  $i$ . Supondo então que o nó  $i$  inicia a sua atividade logo que todos os nós precedentes, ou predecessores, enviam os seus resultados (podendo significar: mensagens, ingredientes ou produtos, entre outros) para o nó  $i$ , então (2.8) descreve quando essas atividades ocorrem. A interpretação das quantidades utilizadas são:

1 -  $x_i(k)$  é o instante de tempo em que o nó  $i$  torna-se ativo pela  $k$ -ésima vez;

2 -  $A_{ij}$  é a soma do tempo de atividade do nó  $j$  mais o tempo de viagem do nó  $j$  para o nó  $i$  (ou pode ser considerado como tempo de transporte ou tempo de comunicação, dependendo do caso).

O fato de se escrever  $A_{ij}$  em vez de  $A_{ji}$  para uma quantidade que vai do nó  $j$  para o nó  $i$  tem a ver com as equações matriciais que serão escritas da forma clássica com vetores coluna, como será visto mais adiante.

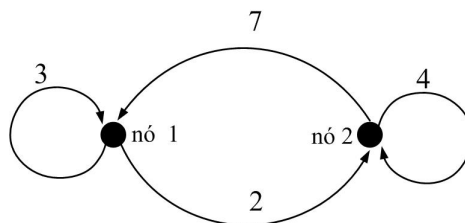


Figura 2.2: Rede correspondente a Equação (2.9).

Na Figura 2.2 é ilustrada uma rede que possui dois nós e quatro arcos. A interpretação dinâmica dessa rede é a seguinte: quando o nó 1 inicia sua atividade, a próxima atividade não pode ser iniciada antes que tenham passado 3 unidades de tempo. Da mesma forma, o tempo entre duas atividades subseqüentes do nó 2 não poderá ocorrer antes de quatro unidades de tempo. O nó 1 envia o resultado de suas atividades para o nó 2, e uma vez iniciada a atividade no nó 1, ela leva 2

unidades de tempo para que o resultado desta atividade chegue ao nó 2. Da mesma forma, após o início de uma atividade do nó 2, o resultado dessa atividade demorará 7 unidades de tempo para chegar ao nó 1. Passa-se agora a supor que uma atividade refere-se a alguma produção. O tempo de produção do nó 1 pode, por exemplo, ser 1 unidade de tempo; depois disso, o nó 1 precisa de 2 unidades de tempo para a recuperação (que pode representar lubrificação, limpeza, manutenção ou outra atividade de preparação da máquina.) e o tempo de viagem do resultado (produto final) do nó 1 para o nó 2, é 1 unidade de tempo. Assim, o valor de  $A_{11} = 3$  é composto da soma do tempo de produção que é igual a 1 mais o tempo de recuperação que é igual a 2. O valor de  $A_{21} = 2$  é composto do tempo de produção que é igual a 1 mais o tempo de viagem que é igual a 1. Da mesma forma, se o tempo de produção do nó 2 é 4, então este nó não precisa de nenhum tempo de recuperação (então  $A_{22} = 4$ ), e o tempo de viagem do nó 2 para o nó 1 é de 3 (então  $A_{12} = 7 = 4 + 3$ ).

Verificando-se a seqüência (2.11) novamente, observa-se que a interpretação dos vetores  $x(k)$  é diferente da inicial. O argumento  $k$  não significa um instante de tempo, mas um contador que diz quantas vezes os vários nós foram ativados. No instante de tempo 14, o nó 1 foi ativado por duas vezes (mais precisamente, o nó 1 começou duas atividades, respectivamente, nos instantes 7 e 11). No mesmo instante de tempo igual a 14, o nó 2 foi ativado três vezes (começando as atividades nos instantes 4, 9 e 13). A contagem das atividades é tal que coincide com o argumento do vetor  $x$ . A condição inicial é considerada como a atividade tempo zero. Na Figura 2.2, pode-se ver que existe sempre um arco que sai de um nó para outro, o que em muitas redes, referindo-se as situações mais práticas, este não será o caso. Se não houver um arco do nó  $j$  para o nó  $i$ , então o nó  $i$  não precisará de nenhum resultado do nó  $j$ . Portanto, o nó  $j$  não tem influência direta sobre o comportamento do nó  $i$ . Nesta situação, considera-se a entrada  $A_{ij}$  como sendo igual a  $-\infty$ . Na Equação (2.8) o termo  $-\infty + x_j(k)$  não influencia  $x_i(k+1)$ , pois  $x_j(k)$  é finito. O número  $-\infty$  irá ocorrer com frequência e será representado por  $\varepsilon$ .

A Equação (2.8) será escrita como

$$x_i(k+1) = \bigoplus_j A_{ij} \otimes x_j(k), \quad i = 1, \dots, n,$$

ou em notação vetorial

$$x(k+1) = A \otimes x(k). \quad (2.12)$$

O símbolo  $\bigoplus_j c(j)$  refere-se ao elemento máximo de  $c(j)$  com respeito a todos os elementos de índice  $j$ , e  $\otimes$  refere-se à adição. Considerando dois escalares  $a$  e  $b$ , uma operação  $a \oplus b$  representará o máximo entre esses dois escalares. Se a condição inicial de (2.12) é  $x(0) = x_0$ , então

$$x(1) = A \otimes x(0),$$

$$x(2) = A \otimes x(1) = A \otimes (A \otimes x(0)) = (A \otimes A) \otimes x_0 = A^2 \otimes x_0,$$

$$x(3) = A \otimes x(2) = A \otimes (A^2 \otimes x(0)) = (A \otimes A^2) \otimes x_0 = A^3 \otimes x_0.$$

em geral

$$x(k) = \underbrace{(A \otimes A \otimes \dots \otimes A)}_{k \text{ vezes}} \otimes x_0 = A^k \otimes x_0.$$

Assim, as matrizes  $A^2, A^3, \dots$ , podem ser calculadas. Considerando a matriz  $A$  de (2.9), tem-se que

$$A^2 = \begin{pmatrix} \max(3+3, 7+2) & \max(3+7, 7+4) \\ \max(2+3, 4+2) & \max(2+7, 4+4) \end{pmatrix}, \quad = \begin{pmatrix} 9 & 11 \\ 6 & 9 \end{pmatrix}.$$

Em geral

$$(A^2)_{ij} = \bigoplus_l A_{il} \otimes A_{lj} = \max_l (A_{il} + A_{lj}). \quad (2.13)$$

Uma extensão de (2.12) é

$$x(k+1) = (A \otimes x(k)) \oplus (B \otimes u(k)), \quad (2.14)$$

$$y(k) = C \otimes x(k).$$

O símbolo  $\oplus$  na fórmula anterior refere-se a maximização componente a componente. O vetor  $u$  de tamanho  $l$  é chamado de vetor de entrada do sistema e o vetor  $y$  de tamanho  $p$  representa a saída do sistema. Os componentes de  $u$  referem-se a nós, que não possuem nós predecessores. Da mesma forma, os componentes de  $y$  referem-se a nós que não possuem nós sucessores. Os componentes de  $x$  agora referem-se aos nós internos, isto é, para nós que possuem tanto sucessores quanto predecessores. As matrizes  $B = \{B_{ij}\}$  e  $C = \{C_{ij}\}$  possuem tamanhos  $n \times l$  e  $p \times n$ , respectivamente. A maneira tradicional para escrever (2.14) é

$$\begin{aligned} x(k+1) &= \max(A_{i1} + x_1(k), \dots, A_{in} + x_n(k), \\ &\quad B_{i1} + u_1(k), \dots, B_{il} + u_l(k)), \quad i = 1, \dots, n, \\ y(k) &= \max(C_{i1} + x_1(k), \dots, C_{in} + x_n(k)), \quad i = 1, \dots, p. \end{aligned}$$

(2.14) podendo também ser escrita como

$$x(k+1) = A \otimes x(k) \oplus B \otimes u(k), \quad (2.15)$$

$$y(k) = C \otimes x(k).$$

entendendo-se que a multiplicação tem prioridade sobre a adição, a Equação (2.14) pode ser escrita

da seguinte forma

$$x(k+1) = Ax(k) \oplus Bu(k), \quad (2.16)$$

$$y(k) = Cx(k).$$

Considerando agora, como um exemplo, um sistema com entradas e saídas dado por:

$$x(k+1) = \begin{pmatrix} 3 & 7 \\ 2 & 4 \end{pmatrix} x(k) \oplus \begin{pmatrix} \varepsilon \\ 1 \end{pmatrix} u(k), \quad (2.17)$$

$$y(k) = \begin{pmatrix} 3 & \varepsilon \end{pmatrix} x(k).$$

a rede correspondente para esse sistema é o da Figura 2.3. Observa-se que  $B_{11} = \varepsilon (= -\infty)$ , justamente pelo fato da entrada estar somente no nó número 2. Considerando as condições iniciais

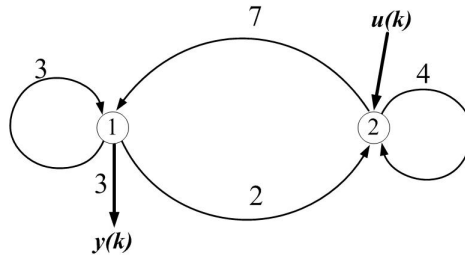


Figura 2.3: Rede com entrada e saída.

as mesmas de (2.10), e que

$$u(0) = 1, \quad u(1) = 7, \quad u(2) = 13, \quad u(3) = 19, \dots$$

então são obtidos os tempos de início de cada atividade em relação ao conjunto de valores das entradas

$$x(0) = \begin{pmatrix} 1 \\ 0 \end{pmatrix}, \quad x(1) = \begin{pmatrix} 7 \\ 4 \end{pmatrix}, \quad x(2) = \begin{pmatrix} 11 \\ 9 \end{pmatrix}, \quad x(3) = \begin{pmatrix} 16 \\ 13 \end{pmatrix}, \dots$$

e o tempo de saída do sistema

$$y(0) = 4, \quad y(1) = 10, \quad y(2) = 14, \quad y(3) = 19, \dots$$

Com esses resultados iniciais, é possível fazer análises mais detalhadas dos sistemas modelados por essas novas equações, e muitos outros sistemas podem ser formulados, os quais levam a equações do tipo (2.12) e (2.14). Para trabalhar e resolver esses problemas, essa teoria será abordada e desenvolvida nos próximos capítulos.

A seguir é ilustrado um exemplo de um grafo dirigido e suas respectivas matrizes modeladas

por meio das equações (2.12) e (2.14).

**Exemplo 2.3.1** (Agendamento). Considere um projeto que consiste em várias tarefas. Algumas dessas tarefas não podem ser iniciadas antes que outras tenham sido concluídas. A dependência destas tarefas pode ser dada em um grafo dirigido em que cada nó coincide com uma tarefa (ou, equivalentemente, com uma atividade). Por exemplo, considere o gráfico da Figura 2.4, há seis nós, numerados de 1 a 6. O nó 1 representa a atividade inicial e nó 6 representa a atividade final. Supõe-se que as atividades, exceto a final, levam certo tempo para serem executadas. Além disso, pode haver tempos de viagem ou transporte. O fato de nó 6 não apresentar um tempo de atividade, não pode ser considerada uma restrição. Se o nó 6 representasse uma atividade com tempo diferente de zero, então, um nó 7 fictício poderia ser adicionado ao nó 6, assim sendo o nó 7 representaria a atividade final. Os arcos entre os nós na Figura 2.4 indicam as restrições de precedência. Por exemplo, o nó 4 não pode começar antes que os nós 2 e 5 tenham terminado suas atividades. O número  $A_{ij}$  associado ao arco do nó  $j$  para o nó  $i$  denota o tempo mínimo decorrido entre o início de uma atividade no nó  $j$  e o início de uma atividade no nó  $i$ .

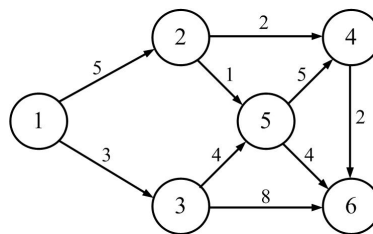


Figura 2.4: Ordem de atividades em um projeto.

Por meio do princípio da programação dinâmica é possível calcular o caminho crítico no grafo. O termo “crítico” se refere a “mais lento”. A duração total do projeto global não pode ser menor do que a soma de todos os números  $A_{ij}$  ao longo do caminho crítico.

Outra forma de encontrar o momento em que a atividade do nó  $i$  pode começar, o qual será denotado  $x_i$ , é a seguinte: Supondo-se que o nó 1 pode começar uma atividade a partir de uma entrada  $u$  injetada neste nó. Essa entrada  $u$  é uma variável que deve ser adicionada externamente ao sistema. Então  $x_1 = u$ . Para os outros  $x_i$  pode ser feito

$$x_i = \max_{j=1,\dots,6}(A_{ij} + x_j). \quad (2.18)$$

Se não existe um arco que liga o nó  $i$  para o nó  $j$ , então  $A_{ji}$  assume o valor  $\varepsilon$ . Então, pode-se escrever o sistema com as seguintes matrizes:

$$x = Ax \oplus Bu, \quad (2.19)$$



sendo

$$A = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon & \varepsilon & 5 & \varepsilon \\ \varepsilon & 1 & 4 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 8 & 2 & 4 & \varepsilon \end{pmatrix}, \quad B = \begin{pmatrix} e \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \\ \varepsilon \end{pmatrix}.$$

No contexto do exemplo  $e = 0$ , ou seja,  $B_{11} = 0$ .

## 2.4 Propriedades Algébricas da Álgebra Max-Plus

*Propriedade 2.4.1 (Associativa).*

$$(a \oplus b) \oplus c = a \oplus (b \oplus c), \quad \forall a, b, c \in \mathbb{R}_{max};$$

e

$$(a \otimes b) \otimes c = a \otimes (b \otimes c), \quad \forall a, b, c \in \mathbb{R}_{max}.$$

*Propriedade 2.4.2 (Comutativa).*

$$a \oplus b = b \oplus a, \quad \forall a, b \in \mathbb{R}_{max};$$

e

$$a \otimes b = b \otimes a, \quad \forall a, b \in \mathbb{R}_{max}.$$

*Propriedade 2.4.3 (Distributividade da  $\oplus$  em relação a  $\otimes$ ).*

$$\begin{aligned} a \otimes (b \oplus c) &= (a \otimes b) \oplus (a \otimes c), \\ (b \oplus c) \otimes a &= (b \otimes a) \oplus (c \otimes a), \end{aligned} \quad \forall a, b, c \in \mathbb{R}_{max}.$$

A multiplicação é distributiva com respeito à soma, tanto a direita quanto a esquerda (nota-se que uma afirmação não implica na outra porque a multiplicação não é necessariamente comutativa).

*Propriedade 2.4.4 (Existência do Elemento nulo).*

$$\exists \varepsilon \in \mathbb{R}_{max} : a \oplus \varepsilon = \varepsilon \oplus a = a, \quad \forall a \in \mathbb{R}_{max};$$

*Propriedade 2.4.5 (Existência do elemento unitário).*

$$\exists e \in \mathbb{R}_{max} : a \otimes e = e \otimes a = a, \quad \forall a \in \mathbb{R}_{max};$$

*Propriedade 2.4.6 (O Elemento nulo é absorvente para  $\otimes$ ).*

$$\varepsilon \otimes a = a \otimes \varepsilon = \varepsilon, \quad \forall a \in \mathbb{R}_{max};$$

**Exemplo 2.4.1.** Seja  $\mathbb{R}_{max} = \mathbb{R} \cup \{-\infty\}$  o mesmo que  $\mathbb{Q}_{max} = \mathbb{Q} \cup \{-\infty\}$  ou  $\mathbb{Z}_{max} = \mathbb{Z} \cup \{-\infty\}$  é definido que:

$$a \oplus b = \max\{a, b\}, e \quad a \otimes b = a + b, \quad \forall a, b \in \mathbb{R}_{max}.$$

Então  $\mathbb{R}_{max}$ , com estas duas operações, verifica todos os axiomas anteriores (nota-se que é necessário agregar ao conjunto  $\mathbb{R}$  o elemento  $-\infty$ , pois caso contrário, a soma não teria elemento nulo). Neste caso temos que  $\varepsilon = -\infty$  e  $e = 0$ . Observa-se também que a soma é idempotente.

*Propriedade 2.4.7 (Idempotência).*

$$a \oplus a = a, \quad \forall a \in \mathbb{R}_{max}.$$

As potências são introduzidas na álgebra max-plus de uma forma natural, utilizando-se a propriedade associativa. De forma a denotar-se uma conjunto de números naturais incluindo o zero de  $\mathbb{N}$  e define-se  $a \in \mathbb{R}_{max}$

$$a^{\otimes n} = \underbrace{a \otimes a \otimes \dots \otimes a}_{n \text{ vezes}}$$

para todo  $n \in \mathbb{N}$  com  $n \neq 0$ , e para  $n = 0$ , define-se  $a^{\otimes 0} = e (= 0)$ . Observa-se que  $a^{\otimes n}$ , para qualquer  $n \in \mathbb{N}$ , é escrita na álgebra convencional como

$$a^{\otimes n} = \underbrace{a \otimes a \otimes \dots \otimes a}_{n \text{ vezes}} = n \times a.$$

Por exemplo,

$$5^{\otimes 3} = 3 \times 5 = 15.$$

Seguindo esta mesma linha, são introduzidas as potências negativas de um número real como

$$8^{\otimes -2} = -2 \times 8 = -16 = 16^{\otimes -1},$$

por exemplo. Na mesma linha, raízes em max-plus podem ser introduzidas como

$$a^{\otimes \alpha} = \alpha \times a,$$

para  $\alpha \in \mathbb{R}$ , tem-se que, por exemplo,

$$8^{\otimes \frac{1}{2}} = \frac{1}{2} \times 8 = 4$$

e

$$12^{\otimes -\frac{1}{4}} = -\frac{1}{4} \times 12 = -3 = 3^{\otimes -1}.$$

Continuando com o ponto de vista algébrico, mostra-se que a álgebra max-plus é um exemplo de uma estrutura algébrica, chamada de semi-anel, que será introduzida a seguir.

**Definição 2.4.1** (Semi-anel [BCOQ92]). Um semi-anel é um conjunto não vazio  $R$  dotado de duas operações binárias  $\oplus_R$  e  $\otimes_R$  tal que

- $\oplus_R$  é associativa e comutativa com elemento nulo  $\varepsilon_R$ ;
- $\otimes_R$  é associativa, distributiva sobre  $\oplus_R$ , e tem elemento unitário  $e_R$ ;
- $\varepsilon_R$  é absorvente para  $\otimes_R$ .

Sendo denotado por  $\mathcal{R} = (R, \oplus_R, \otimes_R, \varepsilon_R, e_R)$ . Se  $\otimes_R$  é comutativo, então  $R$  é chamado comutativo, e se  $\oplus_R$  é idempotente, então  $R$  é chamado idempotente, ou seja,  $a \oplus a = a$ ,  $\forall a \in R$ , “ $\varepsilon_R$ ” é o elemento nulo, e “ $e_R$ ” é o elemento unitário.

A partir desse pressuposto pode-se concluir que a álgebra max-plus é um exemplo de um semi-anel comutativo e idempotente. Importante se faz, no entanto, ressaltar que, apesar da existência de outros semi-aneis, este trabalho não pretende tratar a este respeito.

*Observação 2.4.1.* Muitas vezes o símbolo  $\otimes$  será omitido como acontece habitualmente na álgebra convencional. Por exemplo:

$$\underbrace{a \otimes a \otimes \dots \otimes a}_{k \text{ vezes}} \quad (2.20)$$

será representado por  $a^k$  e  $a^0$  será igual a  $e$ .

**Teorema 2.4.1** (Relação de Ordem [BCOQ92]). *Em um semi-anel  $R$  verifica-se a seguinte equivalência:*

$$a = a \oplus b \iff \exists c : a = b \oplus c, \quad \forall a, b, c \in S.$$

*Estas afirmações equivalentes definem uma relação de ordem (parcial)  $\succeq$  em  $R$  que está definido por:*

$$a \succeq b \iff a = a \oplus b.$$

Esta relação de ordem é compatível com a soma, ou seja:

$$a \succeq b \Rightarrow a \oplus c \succeq b \oplus c, \quad \forall c \in \mathbb{R},$$

com a multiplicação, isto é

$$a \succeq b \Rightarrow ac \succeq bc, \quad \forall c \in \mathbb{R}.$$

Além disso, todo par de elementos  $a$  e  $b$  de  $R$  possui uma menor cota superior (chamada simplesmente de cota superior) na qual é dada por  $a \oplus b$  e  $\varepsilon$  é o menor elemento de  $R$ .

*Demonstração.* Se  $a = a \oplus b$ , então tomando  $c = a$ , resulta que  $b \oplus c = b \oplus a = a$ . Inversamente, se  $a = b \oplus c$ , então somando  $b$  a ambos os lados desta igualdade obtem-se:  $a \oplus b = (b \oplus c) \oplus b = (b \oplus b) \oplus c = b \oplus c = a$ .  $\square$

Para mostrar que  $\succeq$  é uma relação de ordem é necessário provar que ela é reflexiva, anti-simétrica e transitiva.

1. Reflexiva ( $a \succeq a, \forall a \in R$ ):

$$a \oplus a = a \Rightarrow a \succeq a, \quad \forall a \in R.$$

2. Anti-simétrica ( $a \succeq b, b \succeq a \Rightarrow a = b$ ):

$$\left. \begin{array}{l} a \succeq b \Rightarrow a = a \oplus b \\ b \succeq a \Rightarrow b = b \oplus a \end{array} \right\} \Rightarrow a = a \oplus b = b \oplus a = b,$$

3. Transitiva ( $a \succeq b, b \succeq c \Rightarrow a \succeq c$ ):

$$\begin{aligned} \left\{ \begin{array}{l} a \succeq b \\ b \succeq c \end{array} \right\} &\Rightarrow \left\{ \begin{array}{l} a = a \oplus b \\ b = b \oplus c \end{array} \right\} \Rightarrow \left\{ \begin{array}{l} a \oplus c = a \oplus b \oplus c \\ a = a \oplus b \\ b = b \oplus c \end{array} \right\} \Rightarrow \\ &\Rightarrow \left\{ \begin{array}{l} a \oplus c = a \oplus b \\ a = a \oplus b \end{array} \right\} \Rightarrow \{a = a \oplus c\} \Rightarrow \{a \succeq c\}. \end{aligned}$$

Como  $a \succeq b \Rightarrow a = a \oplus b$ , resulta que  $a \oplus c = a \oplus b \oplus c = a \oplus b \oplus c \oplus c = (a \oplus c) \oplus (b \oplus c)$ , e portanto  $a \oplus c \succeq b \oplus c$ .

Como  $a \succeq b \Rightarrow a = a \oplus b$ , resulta que  $ca = c(a \oplus b) = ca \oplus cb$  então  $ca \succeq cb$ .

Claramente  $a \oplus b \succeq a$  e  $a \oplus b \succeq b$  pois

$$(a \oplus b) \oplus a = (a \oplus a) \oplus b = a \oplus b,$$

e analogamente,

$$(a \oplus b) \oplus b = a \oplus (b \oplus b) = a \oplus b.$$

Além disso, se  $c \succeq a$  e  $c \succeq b$ , resulta que  $c = c \oplus a$  e  $c = c \oplus b$ , então

$$(c = c \oplus c) = c \oplus a \oplus c \oplus b = c \oplus a \oplus b,$$

no qual implica que  $c \succeq a \oplus b$ . Portanto,  $a \oplus b$  é a menor das cotas superiores de  $a$  e  $b$ .

Finalmente, como  $a \oplus \varepsilon = a, \forall a \in S$ , resulta que  $a \succeq \varepsilon, \forall a \in S$ , ou seja,  $\varepsilon$  é o menor elemento de  $R$  (como  $\varepsilon$  é o elemento neutro da soma, da última propriedade pode-se concluir que todos os elementos de  $R$  são maiores que zero e portanto positivos).

Note que a relação de ordem  $\succeq$  induzida por  $\oplus$  é uma ordem total, ou seja:

$$a \succeq b \text{ ou } b \succeq a, \quad \forall a, b \in S \quad (2.21)$$

se e somente se

$$a \oplus b = a \text{ ou } b \oplus a = b, \quad \forall a, b \in S \quad (2.22)$$

*Observação 2.4.2.* Uma consequência importante deste fato é que sendo a soma  $\oplus$  idempotente, então o elemento  $a$  não possui oposto, assim, não existe um  $b$  tal que  $a \oplus b = \varepsilon$ , salvo no caso em que  $a$  é  $\varepsilon$ , uma vez que

$$a \oplus b = \varepsilon \Rightarrow \varepsilon \succeq a; \quad (2.23)$$

mas como  $a \succeq \varepsilon$ , resulta que  $a = \varepsilon$ .

Também a soma  $\oplus$  não é cancelativa, ou seja,  $a \oplus b = a \oplus c$  não implica em geral que  $b = c$ . Por exemplo, em  $\mathbb{R}_{max}$ .

$$5 \oplus 3 = \max\{5, 3\} = 5 = \max\{5, 1\} = 5 \oplus 1.$$

### 2.4.1 Vetores e Matrizes na Álgebra Max-Plus

Os elementos de  $x \in \mathbb{R}_{max}^n$  são uma estrutura denominada de vetores max-plus. Esses conjuntos de vetores especificam uma operação interna e uma operação externa, definidas sobre um semi-anel idempotente (também chamados de dioides), sobre  $\mathbb{R}_{max}$ . E seu elemento zero é definido como  $(\varepsilon, \dots, \varepsilon)'$ .

A estrutura dos vetores possui uma operação interna de adição, denotada por  $\otimes$ , que é chamada de álgebra idempotente se  $\otimes$  é associativa, possuindo um elemento unitário  $e$ , e é distributiva com respeito a  $\oplus$ .

$(\mathbb{R}_{max})^{n \times n}$  é o conjunto de matrizes  $n \times n$ , com coeficientes em  $\mathbb{R}_{max}$ , dotado com duas operações internas, dessa forma:

Seja  $R$  um dioide “escalar” arbitrário. Considere o conjunto  $R^{n \times n}$  formado por todas as matrizes quadradas  $n \times n$  com elementos em  $R$ . Então, define-se, sobre este conjunto, a soma e a multiplicação convencionais a partir das operações de soma  $\oplus$  e de multiplicação  $\otimes$  de  $R$ . Uma matriz  $A$  pode ser escrita como,

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nm} \end{pmatrix}.$$

O elemento  $a_{ij}$  pode, também, ser denotado por  $[A]_{ij}$ ,  $i \in \underline{n}$ ,  $j \in \underline{m}$ .

A soma das matrizes  $A, B \in \mathbb{R}_{max}^{n \times m}$ , denotada por  $A \oplus B$ , é definida como:

$$[A \oplus B]_{ij} = a_{ij} \oplus b_{ij} = \max(a_{ij}, b_{ij}). \quad (2.24)$$

Para matrizes  $A \in \mathbb{R}_{max}^{n \times l}$  e  $B \in \mathbb{R}_{max}^{l \times m}$ , o produto matricial  $A \otimes B$  é definido como

$$[A \otimes B]_{ik} = \bigoplus_{j=1}^l a_{ij} \otimes b_{jk} = \max_{j \in \underline{l}} \{a_{ij} + b_{jk}\} \quad (2.25)$$

$i = 1, \dots, n, j = 1, \dots, n, \forall A, B \in S^{n \times n}$ .

Então o conjunto  $R^{n \times n}$ , com estas operações adquire uma estrutura de dioide. Por sua vez, a matriz identidade (elemento neutro da multiplicação) é a matriz que tem a sua diagonal principal formada por elementos de valor igual a  $e$  (elemento neutro da multiplicação de  $R$ ), sendo que fora da diagonal é formada por elementos de valor igual a  $\varepsilon$  (elemento neutro da soma de  $R$ ), tal matriz identidade será representada por  $e$ . Verifica-se então que o elemento neutro da soma em  $R^{n \times n}$  é a matriz com todos os componentes iguais a  $\varepsilon$ , a qual também será representada por  $\varepsilon$ .

Do ponto de vista prático, em particular para os sistemas de equações lineares, o trabalho é feito com matrizes que não são quadradas, e especialmente com vetores fila e coluna.

**Exemplo 2.4.2.** Sejam  $A$  e  $B$  duas matrizes tais que:

$$A = \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \quad B = \begin{bmatrix} 5 & e \\ 3 & e \end{bmatrix}$$

As composições série e paralelo das matrizes acima são dadas por:

Paralela:

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij} = \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \oplus \begin{bmatrix} 5 & e \\ 3 & e \end{bmatrix} = \begin{bmatrix} (2 \oplus 5) & (\varepsilon \oplus e) \\ (3 \oplus 3) & (5 \oplus e) \end{bmatrix} = \begin{bmatrix} 5 & e \\ 3 & 5 \end{bmatrix}.$$

Série:

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj} = \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \otimes \begin{bmatrix} 5 & e \\ 3 & e \end{bmatrix} =$$

$$(A \otimes B)_{ij} = \begin{bmatrix} (2 \otimes 5) \oplus (\varepsilon \otimes 3) & (2 \otimes e) \oplus (\varepsilon \otimes e) \\ (3 \otimes 5) \oplus (5 \otimes 3) & (3 \otimes e) \oplus (5 \otimes e) \end{bmatrix} = \begin{bmatrix} 7 & 2 \\ 8 & 5 \end{bmatrix}.$$

Potência:

$$A^{\otimes 4} = A \otimes A \otimes A \otimes A = \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \otimes \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \otimes \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} \otimes \begin{bmatrix} 2 & \varepsilon \\ 3 & 5 \end{bmatrix} = \begin{bmatrix} 8 & \varepsilon \\ 18 & 20 \end{bmatrix}.$$

## 2.5 Teoria Espectral de Matrizes

Um grafo dirigido  $G$  é definido por um conjunto de nós interconectados por arcos orientados, e é denominado ponderado se os arcos que ligam os nós  $i$  e  $j$  possuem pesos associados a eles.

Esse arco é denotado por  $(i, j)$  e o peso  $w(i, j)$  desse arco é igual ao termo  $A_{ij}$  de uma matriz  $A$  associada ao grafo.

Mais formalmente, um grafo dirigido  $G$  é definido nos seguintes termos:

**Definição 2.5.1** (Grafo Dirigido [BCOQ92]). O grafo dirigido é um par  $(V, \mathcal{E})$ , sendo  $V$  um conjunto de elementos chamados de nós, de  $G$ , e  $\mathcal{E} \subset V \times V$ , um conjunto cujos elementos ordenados (não necessariamente diferentes) de pares de nós, chamados de arcos de  $G$ .

A possibilidade de vários arcos entre dois nós existe, e é denominado de multigrafo. Neste trabalho, no entanto, serão tratados grafos dirigidos, no qual existe, no máximo, um arco (ou seja, zero ou um) entre quaisquer dois nós, a saber, a distinção entre grafos e grafos dirigidos. A diferença entre os dois reside no fato de que, em um grafo os elementos de  $\mathcal{E}$  não são ordenados, enquanto que em um grafo dirigido tais elementos, por sua vez, apresentam-se ordenados. Ao invés de serem denotados como nós e arcos, podem ao mesmo tempo, ser chamados de vértices e arestas, respectivamente.

Define-se o número de nós por  $n$ , e o número individual de nós por  $1, 2, \dots, n$ . Se  $(i, j) \in \mathcal{E}$ , então  $i$  é chamado de nó inicial, ou a origem do arco  $(i, j)$ , e  $j$  é o nó final, ou o destino do arco  $(i, j)$ . Gráficamente, os nós são representados por pontos, e o arco  $(i, j)$  é representado por uma “flecha” de  $i$  para  $j$ .

**Definição 2.5.2** (Antecessor, Sucessor [BCOQ92]). Se em um grafo  $(i, j) \in \mathcal{E}$ , então  $i$  é chamado de um antecessor de  $j$  e  $j$  é chamado de sucessor de  $i$ . O conjunto de todos os antecessores de  $j$  é indicado por  $\pi(j)$  e o conjunto de todos os sucessores de  $i$  é indicado por  $\rho(i)$ . Um antecessor também é chamado de um nó a montante e um sucessor também é chamado de nó a jusante.

**Definição 2.5.3** (Fonte, Reservatório [BCOQ92]). Se  $\pi(i) = \emptyset$ , então o nó  $i$  é chamado de uma fonte, se  $\rho(i) = \emptyset$  então  $i$  é chamado de reservatório. Dependendo da aplicação, uma fonte, respectivamente reservatório, também é chamado de nó de entrada, respectivamente, nó de saída do grafo.

**Definição 2.5.4** (Caminho, Circuito, Laço, Comprimento [BCOQ92]). Um caminho  $(\rho)$  é uma sequência de nós  $(i_1, i_2, \dots, i_p)$ ,  $p > 1$ , tal que  $i_j \in \pi(i_{j+1})$ ,  $j = 1, \dots, p - 1$ . Nó  $i_1$  é o nó inicial e  $i_p$  é o nó final deste caminho. Equivalentemente, também se diz que um caminho é uma sequência de arcos que conecta uma sequência de nós. Um caminho elementar é um caminho no qual nenhum nó aparece mais de uma vez. Quando nó inicial e nó final coincidem, então se chama de circuito. Um circuito  $(i_1, i_2, \dots, i_p = i_1)$  é um circuito elementar se o caminho  $(i_1, i_2, \dots, i_{p-1})$  é elementar. Um laço é um circuito  $(i, i)$ , ou seja, um circuito composto por um único nó, que é inicial e final. Essa definição assume que  $i \in \pi(i)$ , ou seja, existe um arco de  $i$  para  $i$ . O comprimento de um caminho ou um circuito é igual à soma dos comprimentos dos arcos dos quais ele é composto, o comprimento de cada um dos arcos iguais é igual a 1, salvo indicação em contrário. Com essa convenção, o comprimento de um laço é 1. O comprimento de um caminho  $\rho$  é denotado  $|\rho|_l$ , sendo  $l$  referente a caminho. O conjunto de todos os caminhos e circuitos em um grafo é denotado  $R$ . Um dígrafo é dito ser acíclico se não contém  $R$  circuitos.

**Definição 2.5.5** (Descendente, Ascendente [BCOQ92]). O conjunto de descendentes  $\varrho^+(i)$  do nó  $i$  é constituído por todos os nós  $j$  tal que existe um caminho de  $i$  para  $j$ . Da mesma forma o conjunto de ascendentes  $\pi^+(i)$  do nó  $i$  é o conjunto de todos os nós  $j$  tais que existe um caminho de  $j$  para  $i$ . Um deles tem, por exemplo,  $\pi^+(i) = \pi(i) \cup \pi(\pi(i)) \cup \dots$  o mapeamento  $i \mapsto \pi^*(i) = \{i\} \cup \pi^+(i)$  é o fechamento transitivo de  $\pi$ ;  $i \mapsto \varrho^*(i) = \{i\} \cup \varrho^+(i)$  é o fechamento transitivo de  $\varrho$ .

**Definição 2.5.6** (Cadeia, Grafo Conectado [BCOQ92]). Um grafo é chamado conexo se para todos os pares de nós  $i$  e  $j$  existe uma cadeia unindo  $i$  e  $j$ . Uma cadeia é uma sequência de nós  $(i_1, i_2, \dots, i_p)$  de tal modo que entre cada par de nós sucessivos qualquer arco  $(i_j, i_{j+1})$  ou arco  $(i_{j+1}, i_j)$  existe. Se as orientações dos arcos são ignoradas na definição de um caminho, obtém-se uma cadeia.

**Definição 2.5.7** (Grafo Fortemente Conexo [BCOQ92]). Um grafo  $G(A)$  é dito ser fortemente conexo ou conectado se para quaisquer dois diferentes nós,  $i$  e  $j \in G(A)$ , existe um caminho de  $i$  para  $j$ . Equivalentemente,  $i \in \varrho^*(j)$  para todo  $i, j \in V$ , com  $i \neq j$ .

Faz-se notório então ressaltar que de acordo com esta definição, um nó isolado, com ou sem laço, é um grafo fortemente conexo.

### 2.5.1 Grafos e Matrizes

Existem dois tipos de grafos dirigidos (Figuras 2.5 e 2.6), com pesos associados a seus arcos, que podem ser associados com uma matriz com componentes em um dióide.

**Definição 2.5.8** (Grafo de Transição Associado a uma Matriz [BCOQ92]). O grafo de transição associado a uma matriz  $A$  de dimensão  $n \times p$  é um grafo bipartido com  $n + p$  vértices, no qual existe um arco que vai do vértice  $j \in \{1, \dots, p\}$  até o vértice  $p + i$  com  $i \in \{1, \dots, n\}$  se  $A_{ij} \neq \varepsilon$ . Para este caso, o peso associado ao arco é  $A_{ij}$ .

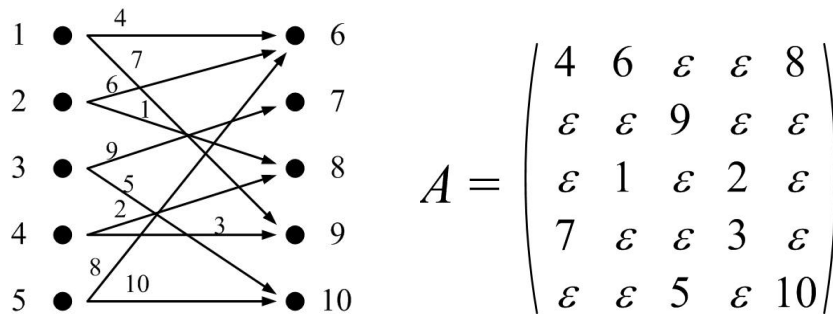


Figura 2.5: Grafo de transição de uma matriz  $5 \times 5$ .

**Definição 2.5.9** (Grafo de Precedência Associado a uma Matriz [BCOQ92]). O grafo de precedência associado a uma matriz  $A$  de dimensão  $n \times n$ , é um grafo dirigido com  $n$  vértices, no qual existe um arco que vai do vértice  $j \in \{1, \dots, n\}$  ao vértice  $i \in \{1, \dots, n\}$  se  $A_{ij} \neq \varepsilon$ . Para este caso o peso associado ao arco é  $A_{ij}$ .



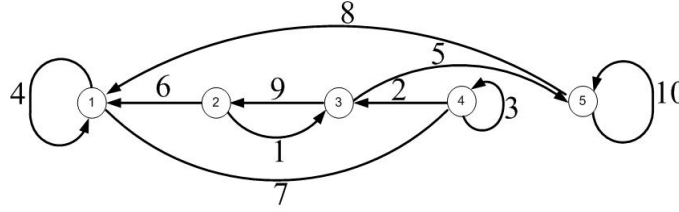


Figura 2.6: Grafo de precedência associado à matriz anterior.

Observa-se que o grafo de transição de uma matriz quadrada  $n \times n$ , a qual possui  $2n$  vértices, pode transformar-se em um grafo de precedência da referida matriz (com  $n$  vértices) da seguinte maneira: basta unir o vértice de  $i$  com o vértice  $n+i$  do grafo de transição para obter o vértice  $i$  do grafo de precedência, onde  $i = 1, \dots, n$ . Isso pode ser verificado comparando-se os grafos anteriores.

A soma de duas matrizes  $A$  e  $B$  de dimensões iguais, corresponde à composição paralela dos respectivos grafos de transição e de precedência, ou seja, existe um arco entre dois vértices se existe um arco entre os ditos vértices em ao menos um dos grafos associados às matrizes, e seu peso associado é a soma ( $\oplus$ ) dos pesos associados dos arcos nos dois grafos.

A composição paralela  $\oplus$  de matrizes é definida para matrizes que possuem o mesmo tamanho pela seguinte regra: Se  $A = A_{ij}$  e  $B = B_{ij}$  tem o mesmo tamanho, então

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij}.$$

O produto  $C = A \otimes B$  de  $A$  por  $B$  ( $A$  de dimensão  $m \times n$  e  $B$  de dimensão  $n \times p$ ) corresponde à composição em série dos respectivos grafos de transição como representada na Figura 2.7. Neste caso, o peso  $C_{ij}$  do arco que vai do vértice  $j$  ao vértice  $i$  no grafo de transição associado à matriz  $C$ , é obtido “somando-se” os pesos de todos os caminhos paralelos que vão do vértice  $j$  até o vértice  $i$  no grafo que se obtém ao concatenar os grafos de transição das matrizes  $A$  e  $B$ .

Nota-se que se o dioide considerado é completamente ordenado, então isto corresponde a ficar com o caminho de máximo peso que vai de  $j$  para  $i$ .

A composição série  $\otimes$  de matrizes  $A$  e  $B$  é definida somente quando o número de colunas de  $A$  é igual ao número de linhas de  $B$  (ou seja,  $A$  é  $m \times n$  e  $B$  é  $n \times p$ ), sendo

$$(A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}.$$

As relações entre grafos e matrizes propiciou uma introdução a símbolos matemáticos que necessitam de maior rigor no tratamento e solução de sistemas de equações matriciais, muitos resultados na álgebra max-plus podem ser interpretados graficamente. Essas relações proporcionaram o desenvolvimento de definições e teoremas importantes para a solução de equações complexas existentes nessa nova álgebra.

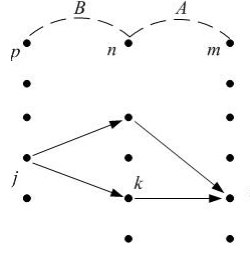


Figura 2.7: Composição paralela de duas matrizes de transição.

### 2.5.2 Definições e Teoremas Importantes da Teoria Espectral de Matrizes

**Definição 2.5.10** (Ciclo Médio [BCOQ92] [HOvdW06]). O peso médio de um caminho é definido como a soma dos pesos dos arcos individuais do caminho, dividido pelo comprimento do caminho. Se o caminho é denotado  $\rho$ , então o peso médio é igual a  $|\rho|_w/|\rho|_l$ . Se este caminho é um circuito, seu peso médio é chamado de ciclo médio.

O objetivo é encontrar o máximo desses ciclos médios, onde o máximo é tomado de todos os circuitos no grafo (circuitos vazios não são considerados). Considerando um grafo de precedência  $G(A)$ , o peso máximo de todos os circuitos de comprimento  $j$  que passam pelo nó  $i$  do grafo pode ser escrito como  $(A^j)_{ii}$ , o máximo destes pesos máximos sobre todos os nós é  $\bigoplus_{i=1}^n (A^j)_{ii}$  que pode ser escrito como traço de  $(A^j)$ , ou seja:

$$\lambda = \bigoplus_{j=1}^n (\text{traço}(A^j))^{1/j}$$

**Definição 2.5.11** (Autovalor e autovetor [BCOQ92] [HOvdW06]). Seja  $A \in \mathbb{R}_{max}$  uma matriz quadrada. Se  $\lambda \in \mathbb{R}_{max}$  é um escalar e  $v \in \mathbb{R}_{max}$  um vetor que contém ao menos um elemento finito de modo que

$$A \otimes v = \lambda \otimes v$$

então  $\lambda$  é chamado de autovalor de  $A$  e  $v$  é um autovetor associado ao autovalor  $\lambda$ .

Para o caso de matrizes irredutíveis, a solução  $\lambda$  é única e calculada por meio do traço de  $A$ , conforme mostrado anteriormente.

**Teorema 2.5.1** (Irredutibilidade [BCOQ92]). *A condição necessária e suficiente para uma matriz quadrada  $A \in \mathbb{R}_{max}$  ser irredutível, é que o grafo de precedência  $G(A)$  seja fortemente conexo.*

**Teorema 2.5.2** ([BCOQ92]). *Se  $A$  é irredutível, ou se  $G(A)$  é fortemente conectada, então existe um e somente um autovalor (mas possivelmente vários autovetores). Este autovalor é igual ao máximo ciclo médio do grafo.*

$$\lambda = \max_{\zeta \in \mathcal{C}(A)} \frac{|\zeta|_w}{|\zeta|_l}, \quad (2.26)$$

sendo  $\zeta$  o intervalo do conjunto de circuitos de  $G(A)$  e  $\mathcal{C}(A)$  o conjunto de todos os circuitos elementares de  $G(A)$ .

O conjunto de caminhos de  $i$  para  $j$  de tamanho  $k > 1$  é denotado por  $P(i, j; k)$ , para um arco  $(i, j)$  em  $G(A)$ , o peso de  $(i, j)$  é dado por  $a_{ji}$  e o peso de um caminho em  $G(A)$  é definido como a soma dos pesos dos arcos que constituem o caminho, o peso de um caminho  $\rho$  é definido como

$$|\rho|_w = \bigoplus_{k=1}^{\infty} A^{\oplus k} a_{i_{k+1}} i_k \quad (2.27)$$

Um grafo de comunicação de  $G(A)$  e as potências de  $A$  estão relacionados entre si. No teorema seguinte, os elementos de  $[A^{\otimes k}]_{ji}$  produzem os máximos caminhos de peso  $k$  do nó  $i$  para o nó  $j$ , desde que tal caminho exista.

**Teorema 2.5.3** (Máximos Caminhos de Peso  $k$  [HOvdW06]). *Seja  $A \in \mathbb{R}_{max}^{n \times n}$ . Então, para todo  $k > 1$ :*

$$[A^{\otimes k}]_{ji} = \max\{|\rho|_w : \rho \in P(i, j; k)\}, \quad (2.28)$$

*Sendo  $[A^{\otimes k}]_{ji} = \varepsilon$  no caso em que  $P(i, j; k)$  é vazio, isto é, quando não existe caminho de tamanho  $k$  do nó  $i$  para o nó  $j$  em  $G(A)$ .*

Demonstração esta que pode ser encontrada em Hidergott et al. ([HOvdW06]).

Supondo que o interesse é calcular o maior tempo de percurso de uma estação  $i$  para uma estação  $j$  dos vários caminhos existentes, então, para  $A \in \mathbb{R}_{max}^{n \times n}$ ,

$$A^+ = \bigoplus_{k=1}^{\infty} A^{\oplus k} \quad (2.29)$$

O elemento  $[A^+]_{ij}$  produz o máximo peso de qualquer caminho de  $j$  para  $i$  (o valor  $[A^+]_{ij} = +\infty$  é possível). De fato, a definição

$$[A^+]_{ij} = \max\{[A^{\oplus k}]_{ij} : k \geq 1\},$$

Sendo  $[A^{\oplus k}]_{ij}$  o máximo peso do caminho de  $j$  para  $i$  de tamanho  $k$ .

**Lema 2.5.1.** *Seja  $A \in \mathbb{R}_{max}^{n \times n}$  tal que qualquer circuito em  $G(A)$  possui peso médio do circuito menor ou igual a  $\varepsilon$ . Então, é seguro afirmar que*

$$A^+ = \bigoplus_{k=1}^{\infty} A^{\oplus k} = A \oplus A^{\oplus 2} \oplus A^{\oplus 3} \oplus \dots \oplus A^{\oplus n} \in \mathbb{R}_{max}^{n \times n}. \quad (2.30)$$

Demonstração em [HOvdW06].

**Teorema 2.5.4** (Relação entre Autovalor e Ciclicidade em uma Matriz Irredutível [BCOQ92] [HOvdW06]). *Seja  $A \in \mathbb{R}_{max}^{n \times n}$  uma matriz irredutível, então*

$$\exists k_0 \in \mathbb{N} \text{ tal que } \forall k \geq k_0 : A^{\otimes k+\varsigma} = \lambda^{\otimes \varsigma} \otimes A^{\otimes k}$$

*sendo  $\lambda$  o autovalor (único) da matriz  $A$  e  $\varsigma$  a ciclicidade de  $A$ .*

**Definição 2.5.12** (Ciclicidade [BCOQ92] [HOvdW06]). Seja  $A \in \mathbb{R}_{max}^{n \times n}$  tal que o grafo de comunicação contém ao menos um circuito. A ciclicidade de  $A$ , denotada por  $\varsigma(A)$ , é a ciclicidade do grafo crítico de  $A$ .

**Lema 2.5.2** (Ciclicidade [BCOQ92] [HOvdW06]). Seja  $A \in \mathbb{R}_{max}^{n \times n}$  uma matriz irredutível com ciclicidade  $\varsigma = \varsigma(A)$ . Então, a ciclicidade da matriz  $A^{\otimes \varsigma}$  é igual a 1.

Essa ciclicidade define um comportamento periódico em regime permanente, após um regime transitório.

$$x(k + \varsigma) = A^{(k+\varsigma)} \otimes x(0), \quad (2.31)$$

$$= \lambda^{\otimes \varsigma} A^{\otimes k} \otimes x(0), \quad (2.32)$$

$$= \lambda^{\otimes \varsigma} \otimes x(k). \quad (2.33)$$

Para um grafo de eventos com ciclicidade igual a 1, tem-se que:

$$x(k + 1) = A \otimes x(k) = \lambda \otimes x(k). \quad (2.34)$$

### 2.5.3 Solução de Equações Lineares

As equações que apresentam  $x(k)$  em ambos os lados são chamadas de implícitas. A solução da equação implícita  $x = Ax \oplus b$  requer a análise de  $G(A)$ , de forma que:

**Teorema 2.5.5** ([BCOQ92]). Se existem apenas circuitos com pesos não positivos em  $G(A)$ , há uma solução para  $x = Ax \oplus b$  que é dada por  $x = A^*b$ . Além disso, se os pesos dos circuitos são negativos, a solução é única (Teorema 3.17 de Bacceli et al. [BCOQ92])

Se  $x = A^*b$  existe, é uma solução, como de fato,

$$A(A^*b) \oplus b = (e \oplus AA^*)b = A^*b, \quad (2.35)$$

portanto,  $x \geq A^*b$ . Além disso, se todos os circuitos do grafo têm pesos negativos, então  $A^k \rightarrow \varepsilon$  quando  $k \rightarrow \infty$ .

Dessa forma, uma definição importante para a solução equações lineares na álgebra max-plus é que para qualquer  $A \in \mathbb{R}_{max}^{n \times n}$ :

$$A^* = I \oplus A^+ = \bigoplus_{k \geq 0} A^{\oplus k}. \quad (2.36)$$

Sendo  $I$  uma matriz identidade. Assim,

$$A^* = \bigoplus_{k=0}^{n-1} A^{\oplus k}. \quad (2.37)$$

O operador  $(^*)$  é chamado estrela de Kleene [BCOQ92].

**Teorema 2.5.6** ([BCOQ92]). *Seja  $A \in \mathbb{R}_{max}^{n \times n}$  e  $b \in \mathbb{R}_{max}^n$ . Se o grafo de comunicação  $G(A)$  possui máximo peso médio de circuito menor ou igual a  $e$ , então o vetor  $x = A^* \otimes b$  soluciona a equação  $x = (A \otimes x) \oplus b$ . Além disso, se os pesos dos circuitos em  $G(A)$  são negativos, então a solução é única.*

Será mostrado que:

$$A^* \otimes b = A \otimes (A^* \otimes b) \oplus b.$$

Então,  $A^*$  existe, implicando em

$$\begin{aligned} A^* \otimes b &= \bigoplus_{k \geq 0} A^{\otimes k} \otimes b, \\ &= \left( \bigoplus_{k \geq 1} A^{\otimes k} \otimes b \right) \oplus (I \otimes b), \\ &= A \otimes \left( \bigoplus_{k \geq 0} A^{\otimes k} \otimes b \right) \oplus (I \otimes b), \\ &= A \otimes (A^* \otimes b) \oplus b, \end{aligned}$$

a fim de provar a unicidade sob a condição de que os circuitos tem pesos médios negativos, argumentado como se segue. Supõe-se que  $x$  é uma solução de  $x = b \oplus (A \otimes x)$ ; substituindo-se sucessivamente a expressão de  $x$  em  $b \oplus (A \otimes x)$ , segue que

$$\begin{aligned} x &= b \oplus (A \otimes b) \oplus (A^{\otimes 2} \otimes x), \\ &= b \oplus (A \otimes b) \oplus (A^{\otimes 2} \otimes b) \oplus (A^{\otimes 3} \otimes x), \\ &= b \oplus (A \otimes b) \oplus \dots \oplus (A^{\otimes(k-1)} \otimes b) \oplus (A^{\otimes k} \otimes x), \\ &= \bigoplus_{l=0}^{k-1} (A^{\otimes l} \otimes b) \oplus (A^{\otimes k} \otimes x). \end{aligned}$$

As entradas de  $A^{\otimes k}$  são os pesos máximos dos caminhos de peso  $k$ . Para  $k$  grande o suficiente, qualquer caminho contém necessariamente uma ou mais cópias de determinados circuitos elementares como subcaminhos, e como  $k$  tende para  $\infty$ , o número de circuitos elementares obrigatórios tende para  $\infty$ . Desde que os circuitos possuam peso negativo, os elementos de  $A^{\otimes k}$  tendem para  $\varepsilon$ , quando  $k$  tende para  $\infty$ , isto é,

$$\lim_{k \rightarrow \infty} A^{\otimes k} \otimes x = \varepsilon.$$

Portanto, deixando  $k$  tender para  $\infty$ , o valor de  $x$  será igual a  $A^* \otimes b$ .

$$\lim_{k \rightarrow \infty} \bigoplus_{l=0}^{k-1} (A^{\otimes l} \otimes b) = \left( \lim_{k \rightarrow \infty} \bigoplus_{l=0}^{k-1} A^{\otimes l} \right) \otimes b = A^* \otimes b.$$

**Exemplo 2.5.1.** Considere as seguintes equações de estado de um sistema, obtidas utilizando-se a álgebra max-plus, tal que:

$$x(k) = A_0x(k) \oplus A_1x(k-1) \oplus B_0u(k) \oplus B_1u(k-1),$$

$$y(k) = C_0x(k) \oplus C_1x(k-1).$$

$$x(k) = \begin{pmatrix} \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon \\ e & 1 & \varepsilon \end{pmatrix} x(k) \oplus \begin{pmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 \end{pmatrix} x(k-1) \oplus \begin{pmatrix} 3 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{pmatrix} u(k) \oplus \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & 1 \\ \varepsilon & \varepsilon \end{pmatrix} u(k-1),$$

$$y(k) = \begin{pmatrix} \varepsilon & \varepsilon & 3 \end{pmatrix} x(k) \oplus \begin{pmatrix} \varepsilon & e & \varepsilon \end{pmatrix} x(k-1).$$

Para eliminar a parte implícita de  $x(k)$ , procedendo de acordo como visto anteriormente, serão realizadas sucessivas substituições de  $x(k)$  em

$$x(k) = A_0x(k) \oplus A_1x(k-1) \oplus B_0u(k) \oplus B_1u(k-1).$$

Sendo necessário que sejam feitas algumas simplificações para facilitar a realização das sucessivas substituições, como:

$$x(k) = A_0x(k) \oplus W, \text{ sendo } W = A_1x(k-1) \oplus B_0u(k) \oplus B_1u(k-1).$$

e sabendo-se que:

$$\lim_{k \rightarrow \infty} A^{\otimes k} \otimes x = \varepsilon.$$

Dessa forma, como  $A_0$  tem máximo ciclo médio negativo [BCOQ92], o Teorema 2.5.6 assegura que:

$$x(k) = A_0^*(A_1x(k-1)) \oplus B_0u(k) \oplus B_1u(k-1).$$

Então,

$$x(k) = \begin{pmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & 2 \end{pmatrix} x(k-1) \oplus \begin{pmatrix} 3 & \varepsilon \\ 4 & \varepsilon \\ 5 & \varepsilon \end{pmatrix} u(k) \oplus \begin{pmatrix} \varepsilon & \varepsilon \\ \varepsilon & 1 \\ \varepsilon & 2 \end{pmatrix} u(k-1).$$

**Exemplo 2.5.2.** Considere a equação bidimensional.

$$x = \begin{pmatrix} -1 & 2 \\ -3 & -1 \end{pmatrix} x \oplus \begin{pmatrix} e \\ 2 \end{pmatrix}.$$

Então,

$$b = \begin{pmatrix} e \\ 2 \end{pmatrix}, \quad Ab = \begin{pmatrix} 4 \\ 1 \end{pmatrix}, \quad A^2b = \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \quad A^3b = \begin{pmatrix} 3 \\ e \end{pmatrix}, \quad A^4b = \begin{pmatrix} 2 \\ e \end{pmatrix}, \dots$$

Assim,

$$x = \begin{pmatrix} e \\ 2 \end{pmatrix} \oplus \begin{pmatrix} 4 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 3 \\ 1 \end{pmatrix} \oplus \begin{pmatrix} 3 \\ e \end{pmatrix} \oplus \begin{pmatrix} 2 \\ e \end{pmatrix} \oplus \dots = \begin{pmatrix} 4 \\ 2 \end{pmatrix}.$$

Esta é a única solução.

Após a solução da equação implícita, demonstrada anteriormente, temos que a forma geral de  $x(k)$  pode ser representada pela forma canônica

$$x(k) = A_0^*(A_1x(k-1) \oplus \dots \oplus B_0u(k) \oplus B_1u(k-1) \oplus \dots).$$

As definições e os teoremas vistos nesta seção foram extraídos de Baccelli et al. [BCOQ92], Cohen [Coh01], Cuninghame-Green e Butkovic [CGB03], Heidergott et al. [HOvdW06] e Garcia [Gar07].

## 2.6 Elementos da Teoria da Residuação

A equação  $f(x) = b$  aparece em diversos problemas que envolvem GET modelados por dioides. Aqui, assume-se que  $f$  é um mapeamento isotônico de um dioide  $S$  em um dioide  $C$ . Assume-se ainda que os dioides  $S$  e  $C$  são completos de forma a garantir a existência de ínfimos e supremos. É claro que  $f$  não é sobrejetiva, a equação não terá solução para alguns valores de  $b$ . Além disso, se  $f$  não é injetiva, a equação não terá solução única. Uma maneira de lidar com esse problema é considerar não somente as soluções possíveis, mas as subsoluções, ou seja, valores de  $x$  tais que  $f(x) \preceq b$ . Analogamente, definem-se as supersoluções  $f(x) = b$  como sendo os valores de  $x$  tais que  $f(x) \succeq b$ . Essencialmente a teoria da residuação busca a máxima subsolução ou a mínima supersolução para a inequação. Maiores detalhes para o assunto em questão, podem ser encontrados em [BJ72].

**Exemplo 2.6.1** (Mapeamento isotônico).

Na álgebra convencional não é preservada a ordem. Considere a desigualdade:

$$\begin{aligned} 3 &> 2 && \text{Multiplicando-se ambos os lados por } -1, \\ (-1).3 &< (-1).2 && \text{O resultado será:} \\ -3 &< -2 && \text{A ordem é invertida.} \end{aligned}$$

Na álgebra max-plus a ordem é preservada. Considere novamente a mesma desigualdade:

$$\begin{aligned} 3 &> 2 && \text{Multiplicando-se ambos os lados por } -1, \\ (-1) \otimes 3 &> (-1) \otimes 2 \\ (-1 + 3) &> (-1 + 2) && \text{O resultado será:} \\ 2 &> 1 && \text{A ordem é preservada.} \end{aligned}$$

**Definição 2.6.1** (Resíduo e Mapeamento Residuável [Mai03]). Um mapeamento isotônico  $f : S \rightarrow \xi$ , no qual  $S$  e  $\xi$  são conjuntos ordenados, é um mapeamento residuável para todo  $y \in \xi$  existe a maior subsolução para a equação  $f(x) = y$  (que é denotada por  $f^\sharp(y)$ ). O mapeamento  $f^\sharp$  é chamado de resíduo de  $f$ .

**Teorema 2.6.1** (Residuação [BCOQ92] [Mai03]). *Seja  $f : S \rightarrow \xi$  um mapeamento isotônico no qual  $S$  e  $\xi$  são conjuntos ordenados, então  $f$  é residuável se e somente se  $f^\sharp$  é o único mapeamento isotônico tal que*

$$f \circ f^\sharp(y) \preceq y \quad e \quad f^\sharp \circ f(x) \succeq x \quad (2.38)$$

$\forall x \in S$  e  $\forall y \in \xi$ .

### 2.6.1 Residuação do Produto para o Dioide Matricial

Considerando matrizes escalares  $n \times n$  com elementos pertencentes a um dioide “escalar”  $S$ , as operações de soma e de produto de matrizes são definidas de maneira natural a partir da soma e da multiplicação no dioide  $S$ , ou seja:

$$(A \oplus B)_{ij} = A_{ij} \oplus B_{ij} \text{ e } (A \otimes B)_{ij} = \bigoplus_{k=1}^n A_{ik} \otimes B_{kj}. \quad (2.39)$$

Esse conjunto de matrizes com operações definidas acima é um dioide que é representado por  $S^{n \times n}$ . Assim, o mapeamento  $L_A$  é definido sobre o dioide matricial  $S^{n \times n}$  como

$$L_A : S^{n \times n} \rightarrow S^{n \times n}, \quad (2.40)$$

$$X \mapsto AX. \quad (2.41)$$

**Teorema 2.6.2** ([BCOQ92]). *Considere o dioide  $S$  completo e  $S^{n \times n}$  o dioide das matrizes com elementos em  $S$  e ainda as matrizes  $A, B \in S^{n \times n}$ . O supremo da inequação  $AX \preceq B$  existe e é dado pela matriz  $L_A^\sharp(B) \in S^{n \times n}$ , também representada por  $\frac{B}{A} = A \backslash B$ . Os elementos dessa matriz são calculados por:*

$$(A \backslash B)_{ij} = \bigwedge_{l=1}^n A_{il} \backslash B_{lj}. \quad (2.42)$$

$\backslash$  significa divisão à esquerda; resíduo do produto à esquerda. De acordo com [Mai03], uma aplicação interessante da residuação diz respeito à solução da equação  $AX = B$ . Pode-se mostrar facilmente que essa equação tem solução se e somente se  $A(A \backslash B) = B$ . De fato, se  $X$  é solução da equação, então  $X \preceq A \backslash B$ . Por isotonia da multiplicação, tem-se que  $AX = B \preceq A(A \backslash B) \preceq B$ .

## 2.7 Conclusão

Neste capítulo, foram abordadas muitas características e propriedades da estrutura algébrica chamada de álgebra max-plus. Essas características e propriedades são muito semelhantes as encontradas em estruturas matemáticas mais familiares, como a álgebra convencional. Foram utilizadas



operações matriciais, aplicações para soluções de sistemas de equações max-plus, e foi desenvolvida uma abordagem com o objetivo de mostrar o significado para autovalores.

Utilizando-se alguns exemplos, foi possível observar que a álgebra max-plus é uma ferramenta muito importante na análise de sistemas de modelados utilizado-se essa álgebra, principalmente quando o objetivo é formular e resolver problemas de otimização. Existem inúmeras aplicações que utilizam a álgebra max-plus, e todas essas aplicações certamente não estão limitadas as apresentadas nesta seção. O objetivo foi apresentar propriedades, definições e teoremas importantes para solucionar o problema do controle do fluxo de entrada para sistemas modelados por grafos de eventos temporizados (que serão abordados no próximo capítulo), utilizando-se o tema de uma forma compreensível e de fácil leitura, buscando incentivar o leitor a pesquisar novas aplicações da álgebra max-plus.

No capítulo a seguir, será tratado o tópico sobre sistemas a eventos discretos, abordando especificamente as Redes de Petri com o objetivo de solidificar os conceitos sobre modelagens de sistemas utilizando-se essa ferramenta gráfica, proporcionado a base necessária para o entendimento de grafos de eventos temporizados e a representação matemática utilizando-se equações de datadores. O próximo capítulo e o atual são as bases para o desenvolvimento da proposta de controle desenvolvida nesta tese.

## Capítulo 3

# Modelagem Matemática de Sistemas

Um sistema a eventos discretos é um sistema a estado discreto, dirigido por eventos, ou seja, sua evolução de estado depende inteiramente da ocorrência de eventos discretos assíncronos no tempo. Neste sentido, Attié [Att98] escreve que “quando o espaço de estados de um sistema é naturalmente descrito por um conjunto discreto, e as transições de estado são observadas somente em pontos discretos do tempo, associam-se estas transições a eventos. O conceito de evento é um desses conceitos primitivos, cuja compreensão deve ser deixada à intuição, mais do que a uma exata definição. Não se pode, porém, deixar de enfatizar que um evento deve ser pensado como sendo de ocorrência instantânea e como causador de uma transição no valor (discreto) do estado do sistema”.

Segundo Cury [Cur01], um sistema é uma parte limitada do Universo que interage com o mundo externo através das fronteiras que o delimitam. Os sistemas de interesse percebem as ocorrências no mundo externo através da recepção de estímulos, denominados eventos. A ocorrência de um evento causa, em geral, uma mudança interna no sistema, a qual pode ou não se manifestar a um observador externo. Além disso, uma mudança pode ser causada pela ocorrência de um evento interno ao próprio sistema, tal como o término de uma atividade ou o fim de uma temporização. Em qualquer caso, essas mudanças se caracterizam por serem abruptas e instantâneas: ao perceber um evento, o sistema reage imediatamente, acomodando-se em tempo nulo numa nova situação, onde permanece até que ocorra um novo evento. A ocorrência desses eventos pode depender de fatores alheios ao sistema, de modo que este não tem, em geral, como prevê-los.

O assunto de sistema a eventos discretos, descrito neste trabalho, está baseado em [CL99]. Neste capítulo será feita uma revisão dos principais conceitos e dos fundamentos da teoria de sistemas a eventos discretos, juntamente com o estudo de grafos de eventos temporizados. O capítulo está estruturado da seguinte forma: na Seção 3.1 são apresentados os fundamentos da teoria de sistemas a eventos discretos, com a introdução dos conceitos de evento, sistemas dirigidos pelo tempo e sistemas baseados em eventos, tratados em comparação aos sistemas dinâmicos de variáveis contínuas; na Seção 3.2 são estudadas as rede de Petri, sendo então apresentados os fundamentos de redes de Petri, as equações de estado e as dinâmicas da rede de Petri para uma classe especial de redes; na Seção 3.3 é introduzida a estrutura de temporização nas redes de Petri, gerando as redes de Petri temporizadas; na Seção 3.4 os grafos de eventos temporizados são abordados juntamente com a representação das equações utilizando-se datadores; na Seção 3.5 é tratada a questão das matrizes

de transferência com uso da transformada- $\gamma$ ; na Seção 3.6 é vista a dinâmica dos GET; na Seção 3.7 são apresentados alguns exemplos de aplicações e, finalmente, na Seção 3.8 é feita a conclusão do capítulo.

### 3.1 Sistemas a Eventos Discretos

Nesta seção serão apresentados os principais conceitos para o estudo de sistemas a eventos discretos.

#### 3.1.1 Evento

“Evento” é um conceito primitivo e necessita de uma boa base intuitiva para compreender o seu significado. Deve ser enfatizado que um evento deve ser pensado como alguma coisa acontecendo instantaneamente e que causa transições de um valor de estado para outro. Um evento pode ser identificado como uma ação específica; por exemplo, alguém aperta um botão, um computador deixa de funcionar, a chave de ignição de um automóvel é ligada etc. ou pode ser o resultado de várias condições que, de repente, acontecem.

Neste trabalho será usado o símbolo “ $\tau$ ” para denotar um evento. Ao considerar um sistema afetado por tipos diferentes de eventos, define-se um conjunto “ $E$ ” cujos elementos são todos estes eventos. Claramente,  $E$  é um conjunto discreto.

O conceito de evento pode ser melhor entendido com a ajuda do seguinte exemplo: um sistema de armazenamento de cargas contendo produtos acabados produzidos por uma fábrica. Sempre que um novo produto é produzido no processo de fabricação, este é direcionado para o armazém e é armazenado lá. Um caminhão aparece periodicamente e é carregado com um certo número de produtos acabados, que são consideradas como saídas do armazém. Neste caso pode-se perfeitamente verificar que há, no mínimo, dois eventos: um evento é a “chegada de produto” e o outro é a “chegada de caminhão”. Neste caso, pode-se definir um conjunto de eventos  $E = \{P, T\}$  no qual  $P$  denota o evento “chegada de produto”, e  $T$  denota o evento “chegada de caminhão”, que corresponde à “saída do produto”.

#### 3.1.2 Sistemas Dirigidos pelo Tempo e Sistemas Baseados em Eventos

Em sistemas de estados contínuos o estado muda geralmente com mudanças de tempo. Isto é particularmente evidente em modelos a tempo-discreto: um sinal de “clock” determina a sequência de amostras a serem obtidas, pois é esperado que a cada marcação desse sinal, ocorra uma mudança no estado do sistema. Neste caso, a variável de tempo ( $t$ , em tempo contínuo, ou  $k$ , em tempo-discreto) é uma variável independente que aparece como sendo o argumento de toda a contribuição de entrada dos estados, e em funções de saída. Por essa razão, esses sistemas são denominados dirigidos pelo tempo.

Em sistemas de estados discretos, as mudanças de estado só ocorrem em certos pontos por transições instantâneas e, a cada uma dessas transições, pode-se associar um evento. Suponha que exista um relógio pelo qual é tomado o tempo, e considere as duas possibilidades:

1. A toda marcação do sinal de clock, um evento  $\tau$  será selecionado de um conjunto fixo  $E$ . Se nenhum evento acontecer, pode-se pensar em um “evento nulo” como pertencendo a  $E$  cuja pro-

priedade é não causar nenhuma mudança de estado.

2. Em vários momentos de tempo (não necessariamente conhecidos com antecedência, e não coincidindo com as marcações de tempo), algum evento determinado irá ocorrer.

Há uma diferença fundamental entre as possibilidades 1 e 2. Em 1, as transições de estado são sincronizadas pelo relógio, isto é, a toda marcação de tempo, um evento (ou nenhum evento) é selecionado. O tempo é responsável por toda e qualquer possível transição de estado. Em 2, todo evento  $\tau$  de  $E$  define um processo distinto pelo qual os momentos de tempo quando  $\tau$  acontece são determinados. As transições de estado são o resultado das combinações destes processos de eventos assíncronos e simultâneos. Além disso, estes processos não precisam ser independentes um do outro.

A distinção entre a possibilidade 1 e a possibilidade 2 dá origem às definições de sistemas dirigidos pelo tempo (1) e sistemas dirigidos por eventos (2). É importante ressaltar que a idéia de transições de estado baseadas em eventos corresponde a uma noção familiar em computação, que é de uma “interrupção” em sistemas de computador. Enquanto muitas das funções em um computador são sincronizadas por um relógio, e são controladas pelo tempo, outras são resultados de chamadas assíncronas que podem acontecer a qualquer hora como, por exemplo, o pedido de um usuário externo ou uma mensagem de intervalo pode acontecer como resultado de eventos específicos, mas completamente independentes do relógio do computador.

### 3.1.3 Propriedades Características de Sistemas a Eventos Discretos

A maioria dos sistemas de controle em engenharia são baseados em modelos de equações diferenciais ou em equações a diferenças lineares. Para usar estes modelos matemáticos, esses sistemas devem satisfazer as seguintes propriedades: devem ser de estado contínuo; com o mecanismo de transição de estado dirigido pelo tempo. A primeira propriedade permite definir o estado por meio de variáveis contínuas que podem assumir qualquer valor real (ou complexo). Quantidades físicas comuns como posição, velocidade, aceleração, temperatura, pressão e fluxo, estão nesta categoria desde que se possam definir naturalmente as derivadas para estas variáveis contínuas. A segunda propriedade vem o fato de que o estado geralmente evolui em função do tempo.

Os sistemas considerados neste trabalho são os Sistemas Dinâmicos a Eventos Discretos (SDED) ou, mais amplamente, Sistemas a Eventos Discretos (SED). As suas principais características são: (i) o espaço de estado é um conjunto discreto; (ii) o mecanismo de transição de estados é baseado em eventos. Essas propriedades levam a seguinte definição de SED.

**Definição 3.1.1** (Um Sistema a Eventos Discretos - SED [CL99]). É um sistema de estado discreto baseado em eventos, isto é, a evolução dos estados depende somente da ocorrência de eventos discretos assíncronos.

Muitos sistemas, particularmente tecnológicos, são na realidade sistemas de estados discretos. Até mesmo se este não for o caso, para muitas aplicações de interesse, uma visão de estado discreto de um sistema complexo pode ser necessária. Alguns exemplos simples de sistemas de estados discretos são: (i) O estado de uma máquina pode ser selecionado de um conjunto como {LIGADA,

DESLIGADA} ou {OCUPADO, OCIOSO}; (ii) um computador que executa um programa pode ser visto como estando em um de três estados: {ESPERANDO POR INSTRUÇÕES, EXECUTANDO, PARADO}; (iii) qualquer tipo de inventário que consiste de valores discretos (por exemplo, produtos, unidades monetárias, pessoas) tem um espaço de estado natural nas grandezas não negativas  $\{0, 1, 2, \dots\}$ , (iv) a maioria dos jogos pode ser modelado como tendo um espaço de estado discreto (em xadrez, por exemplo, toda possível configuração do tabuleiro define um estado); o espaço resultante é enorme, mas é discreto.

A propriedade baseada em eventos de SED decorre do fato de que o estado só pode mudar no tempo em pontos discretos, que correspondem fisicamente a ocorrências assíncronas de eventos discretos. De um ponto de desenvolvimento de um modelo, isto tem a seguinte implicação: se for possível identificar um conjunto qualquer de “eventos” que podem causar uma transição de estado, então o tempo já não serve ao propósito de dirigir tal sistema e não pode ser uma variável independente apropriada.

As duas características fundamentais que distinguem Sistemas Dinâmicos de Variáveis Contínuas (SDVC) de SED são claramente mostradas ao se comparar trajetórias típicas de cada uma destas classes de sistema, como na Figura (3.1). Para o SDVC mostrado, o espaço de estado  $X$  é o conjunto de números reais  $\mathbb{R}$ , e  $x(t)$  pode assumir algum valor fixo. A função  $x(t)$  é a solução da equação diferencial  $\dot{x}(t) = f[x(t), u(t), t]$ , sendo  $u(t)$  a entrada. Para o SED, o espaço  $X$  é fixo e discreto, de forma que  $X = \{s_1, s_2, s_3, s_4, s_5, s_6\}$ . De acordo com a trajetória ilustrada na Figura (3.1b), o estado só muda de um valor para outro se um evento ocorrer. Vê-se, inclusive, que um evento pode acontecer, mas não causar uma transição de estado, como no caso de  $\tau_6$ .

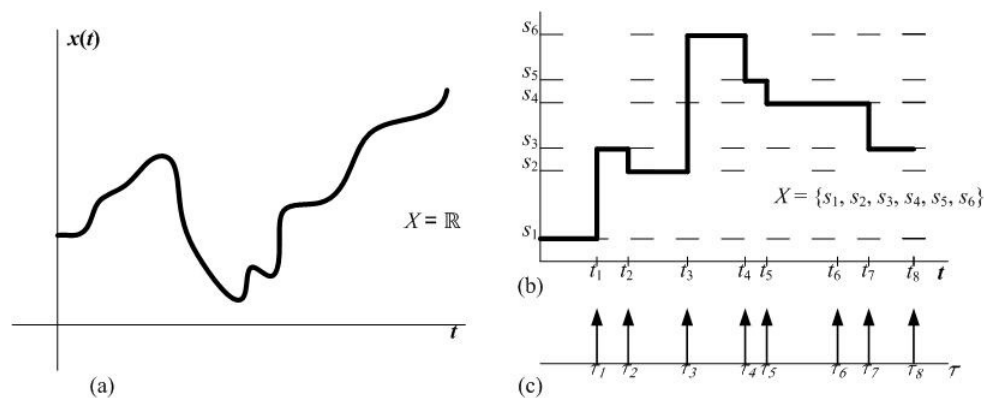


Figura 3.1: Comparação de caminhos de amostra para Sistemas Dinâmicos de Variáveis Contínuas (SDVC) (a) e Sistemas de Evento Discretos (b).

### 3.1.4 Exemplos de Sistemas a Eventos Discretos

Nesta seção serão apresentados três exemplos de SED utilizados no mundo real e experiências comuns em engenharia. O primeiro desses exemplos representa uma estrutura que servirá para representar muitos SED de interesse.

### Sistemas de Filas

O termo fila decorre de um fato intrínseco que em muitos dos sistemas mais comuns, para se usar certos recursos, deve-se esperar. Por exemplo, para usar os recursos de um caixa de banco, as pessoas formam uma fila e esperam; para usar o recurso de um caminhão, produtos acabados esperam em um armazém. De forma semelhante, para usar os recursos da CPU, várias tarefas esperam em algum lugar no computador até que seja dado acesso às mesmas por mecanismos potencialmente complexos.

Há três elementos básicos em um sistema de filas:

1 - As entidades que fazem a espera para utilização dos recursos. Estas entidades são usualmente denominadas clientes.

2 - Os recursos para os quais a espera é realizada. Desde que os recursos provejam alguma forma de serviço aos clientes, devem ser genericamente chamados de servidores.

3 - O espaço onde a espera é realizada. A esse elemento dá-se o nome fila.

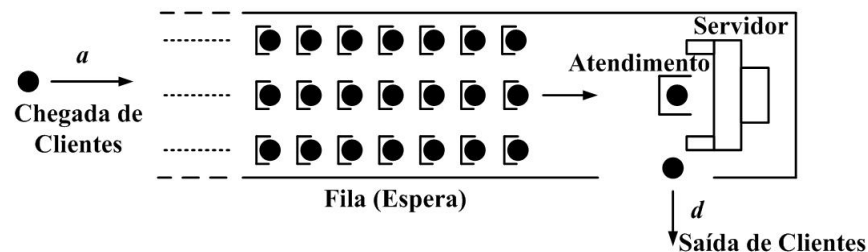


Figura 3.2: Um sistema de fila.

A cada chegada, o cliente, ou se dirige ao SERVIDOR e é servido, ou tem que esperar primeiro na FILA até que o servidor esteja disponível. Após ser atendido, cada cliente parte. Exemplos de clientes são: pessoas (por exemplo, esperando em um banco ou em um ponto de ônibus), mensagens transmitidas de algum meio de comunicação, tarefas, trabalhos ou transações executadas em um sistema de computador, produção em um processo de fabricação e carros que usam uma rede de estradas. Exemplos de servidores são: pessoas (por exemplo, caixas de banco ou caixas de saída de supermercado), canais de comunicação responsáveis pela transmissão de mensagens, processadores de computador ou dispositivos periféricos, vários tipos de máquinas usadas na fabricação e semáforos que regulam o fluxo de carros. Exemplos de filas são encontrados em vários locais, como por exemplo, banco, pontos de ônibus ou supermercados. Porém, filas também estão presentes em redes de comunicação ou sistemas de computador onde também são alocadas formas menos tangíveis de clientes, como telefonemas ou tarefas a serem executadas em áreas de espera.

Um sistema de fila é ilustrado na Figura (3.2). Os clientes são vistos como chegando à fila e partindo do servidor. Supõe-se ainda que o processo de servir os clientes, normalmente leva uma quantidade estritamente positiva de tempo (caso contrário não haveria espera). Assim, um servidor pode ser visto como um “bloco de atraso” que retém um cliente por algum tempo até a realização do serviço.

Visto como um SED, o sistema de fila da Figura (3.2) tem um conjunto de eventos  $E = \{a, d\}$ , no qual  $\{a\}$  denota um evento de chegada e  $\{d\}$  denota um evento de saída. Uma variável de estado é o número de clientes na fila ou o comprimento da fila. Assim, o espaço de estados é o conjunto de valores não negativos  $X = \{0, 1, 2, \dots\}$ .

### Sistemas de Manufatura

Em um processo industrial, os clientes são as peças ou partes de peças da produção. Essas peças estão dispostas para o acesso aos vários servidores da fábrica que são as máquinas que executam operações específicas e dispositivos de manipulação de material, como robôs e correias transportadoras. Quando as peças não estão sendo trabalhadas, elas estão armazenadas em um estoque ou em uma fila sobre a esteira transportadora de produtos (peças) até que o servidor libere o acesso para a próxima operação que está disponível. Por causa de limitações físicas, filas em um sistema industrial têm normalmente capacidades finitas.

Uma vez mais, modelos de filas provêm uma conveniente descrição para sistemas industriais. Um exemplo com duas máquinas é ilustrado na Figura (3.3), onde as peças passam por duas máquinas, sendo a capacidade da primeira fila da primeira máquina muito maior que a da segunda máquina, considerando-se também o estoque anterior a primeira máquina pode-se dizer que é infinito, enquanto a capacidade da fila da segunda máquina é limitada a dois. Como resultado, é possível que uma parte de serviço da máquina 1 seja completado, porém a máquina 2 esteja ocupada e além disso a fila esteja completa. Neste caso, a peça tem que permanecer na máquina 1 embora não requeira mais nenhum serviço; além disso, são forçadas outras peças a esperar o acesso na máquina 1 permanecendo em fila. O conjunto de eventos fixado para este exemplo é  $E = \{a, c_1, d_2\}$ , sendo  $a$  uma chegada para a primeira máquina,  $c_1$  é uma conclusão de serviço da primeira máquina e  $d_2$  é uma partida para a fila da segunda máquina. Observe que o evento  $c_1$

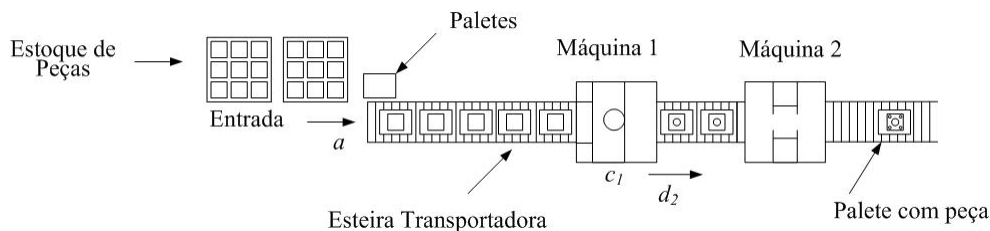


Figura 3.3: Sistema industrial de filas.

não implica em movimento de uma peça da máquina 1 para a fila da máquina 2, desde que esta possibilidade esteja bloqueada. O estado do sistema pode ser definido como um vetor  $x = [x_1, x_2]^T$  correspondendo aos comprimentos de fila das duas máquinas. Neste caso,  $x_2$  é restrito aos valores  $\{0, 1, 2, 3\}$ . Porém, note que quando  $x_2 = 3$ , a máquina 1 é bloqueada, pois acabou de executar o serviço na peça e a fila da segunda máquina está completa. Para modelar o fenômeno de bloqueio necessitamos introduzir uma variável adicional  $B$  que  $x_2$  pode gerar. O espaço de estados se torna o conjunto discreto  $X = \{(x_1, x_2) : x_1 \geq 0, x_2 \in \{0, 1, 2, 3, B\}\}$ . Para ilustrarmos a flexibilidade do processo modelado (dependendo do nível de detalhe que se deseja capturar) pode-se gerar um

espaço de estados alternativo que pode ser:  $X = \{(x_1, x_2) : x_1 \in \{I, W, B\} \text{ e } x_2 \in \{I, W\}\}$ , sendo  $x_1$  o estado da primeira máquina, que pode assumir os seguintes valores: inativo ( $I$ ), trabalhando ( $W$ ) ou bloqueado ( $B$ ), e  $x_2$  é o estado da segunda máquina, que pode assumir os seguintes valores: inativo ( $I$ ) ou trabalhando ( $W$ ). Neste modelo, não são focalizados os comprimentos das filas, mas sim os estados lógicos de cada máquina.

### Sistemas de Tráfego

Considere, agora, como exemplo uma interseção em T (Figura 3.4). Há quatro tipos de movimentos de veículos: (a) veículos vindo de ponto 1 e virando para o ponto 2; (b) veículos vindo de 1 e virando para o ponto 3; (c) veículos que vão diretamente do ponto 2 ao 3, e (d) veículos que vão do ponto 3 ao 2. O semáforo funciona da seguinte forma: fica vermelho para os veículos vindo da posição 1 e verde para os veículos vindo das posições 2 e 3, permitindo assim os movimentos “c” e “d”, ou ao contrário, vermelho para os veículos vindo das posições 2 e 3 e verde para os veículos vindo da posição 1, permitindo os movimentos “a” e “b”. Neste caso, o conjunto de eventos é determinado por:

$$E = \{a_{12}, a_{13}, a_{23}, a_{32}, d_{12}, d_{13}, d_{23}, d_{32}, g, r\},$$

sendo  $a_{12}, a_{13}, a_{23}$ , e  $a_{32}$  as chegadas de veículo em cada uma das quatro possibilidades e  $d_{12}, d_{13}, d_{23}$  e  $d_{32}$  são as partidas de veículo quando o semáforo permite o tráfego,  $g$  e  $r$  indicam o estado do semáforo.

Um possível espaço de estado é definido pelos comprimentos de fila formados pelos quatro tipos de veículo e o estado do próprio semáforo, isto é :

$$X = \{(x_{12}, x_{13}, x_{23}, x_{32}, y)\} : x_{12}, x_{13}, x_{23}, x_{32} \geq 0, y \in \{g_1, g_2, g_3, r_1, r_2, r_3\},$$

sendo  $x_{12}, x_{13}, x_{23}$ , e  $x_{32}$  os quatro comprimentos de fila, e  $y$  é o estado da luz ( $g_i$  e  $r_i$  denotam, respectivamente, verde e vermelho para os veículos que vem dos pontos indicados).

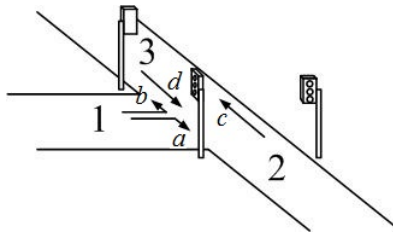


Figura 3.4: Uma interseção T controlada por semáforos.

### 3.2 Redes de Petri

Uma rede de Petri é um dispositivo que manipula eventos de acordo com regras estabelecidas. Uma de suas características principais é a existência de condições explícitas sob as quais um evento pode ou não ser habilitado, permitindo representações de SED muito gerais, cuja operação depende de esquemas de controle potencialmente complexos.



Os principais modelos utilizados para tratar de sistemas a eventos discretos são os seguintes [CL99]:

- Redes de Petri com e sem temporização;
- Redes de Petri Controladas com e sem temporização;
- Cadeias de Markov;
- Teoria das Filas;
- Processos Semi-Markovianos Generalizados (GSMP) e Simulação;
- Álgebra de Processos;
- Álgebra Max-Plus;
- Lógica Temporal e Lógica Temporal de Tempo Real;
- Teoria de Linguagens e Autômatos.

Dentre os modelos citados anteriormente, dois merecem atenção: as Redes de Petri Controladas e o modelo de Ramadge e Wonham (1989), denominado Teoria de Controle Supervisório (TCS), baseado em Linguagens e Autômatos. Diferentemente dos outros modelos que enfatizam a análise de sistemas, além de construir os controladores através de ajustes, os dois modelos citados são dotados de procedimentos de síntese de controladores. Estes modelos, por esta característica, têm dado forte contribuição ao desenvolvimento da teoria de controle de sistemas automatizados de manufatura [OS08].

Os modelos de Ramadge-Wonham (temporizados ou não), baseado na Teoria de Autômatos e/ou Linguagens, e o de Redes de Petri Controladas (temporizadas ou não) são dotados de procedimentos de síntese de controladores. Estes modelos têm dado forte contribuição ao desenvolvimento da teoria de Controle de SED. De todo modo, nenhum dos modelos acima serve atualmente como paradigma [Cur01].

### 3.2.1 Fundamentos de Redes de Petri

A definição de uma rede de Petri é feita em dois passos: primeiro, define-se o grafo da rede de Petri, também chamado estrutura da rede de Petri; em seguida, junta-se a esse grafo um estado inicial, um conjunto de estados marcados, e uma função de transição rotulada, resultando no modelo completo da rede de Petri.

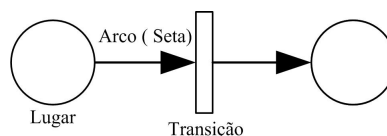


Figura 3.5: Elementos de uma rede de Petri.

Em uma rede de Petri (Figura 3.5), os eventos estão associados às “transições”. Para que uma transição aconteça, várias condições devem ser satisfeitas. Estas condições são localizadas nos próprios estados ou “lugares” com suas informações relacionadas a cada transição. Os lugares podem ser definidos como “entradas” ou “saídas” para uma transição. Transições, lugares, e as relações entre esses definem os componentes básicos de um grafo da rede de Petri. O grafo de uma rede de Petri tem dois tipos de nós (lugares e transições) e arcos conectando estes nós. Por esta razão a rede de Petri é um grafo bipartido, no sentido que os arcos não podem conectar diretamente nós do mesmo tipo, isto é, os arcos conectam nós de lugares para nós de transição e nós de transição para nós de lugares. Pode-se, então apresentar a seguinte definição.

**Definição 3.2.1** (Grafo da rede de Petri [CL99]). Um grafo ou estrutura da rede de Petri é um grafo ponderado bipartido  $(P, T, A, w)$ , sendo  $P$  o conjunto finito de lugares,  $T$  é o conjunto finito de transições,  $A \subseteq (P \times T) \cup (T \times P)$  é o conjunto de arcos dos lugares para as transições e das transições para os lugares no gráfico e  $w : A \rightarrow \{1, 2, 3, \dots\}$  é a função dos pesos dos arcos (um número inteiro positivo).

Em uma rede de Petri, é conveniente usar  $I(t_j)$  para representar o conjunto de lugares de entrada da transição  $t_j$ . Da mesma forma,  $O(t_j)$  representa o conjunto de lugares de saída da transição  $t_j$ . Assim, temos

$$I(t_j) = \{p_i \in P : (p_i, t_j) \in A\}, \quad O(t_j) = \{p_i \in P : (t_j, p_i) \in A\}.$$

Notação semelhante pode ser usada para descrever as transições de entrada e de saída para um determinado lugar  $p_i$ :  $I(p_i)$  e  $O(p_i)$ .

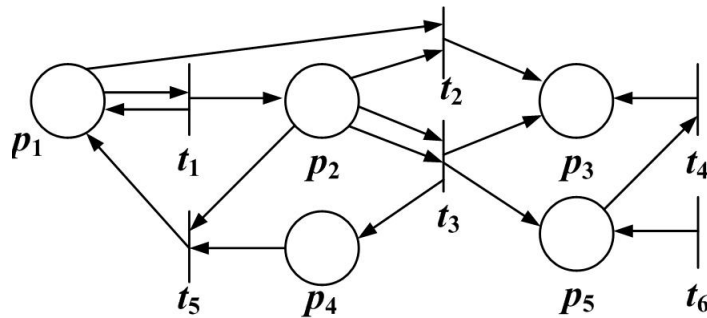


Figura 3.6: Rede de Petri com 5 lugares e 6 transições.

Para tornar mais clara a Definição (3.2.1), considere o grafo de uma rede de Petri ilustrado na Figura (3.6). Tem-se que o conjunto de lugares é  $P = \{p_1, p_2, p_3, p_4, p_5\}$ , o conjunto de transições é  $T = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ , O conjunto de arcos  $A = \{(p_1, t_1), (p_1, t_2), (p_2, t_2), (p_2, t_3), (p_2, t_5), (p_4, t_5), (p_5, t_4), (p_5, t_6), (t_1, p_1), (t_1, p_2), (t_2, p_3), (t_3, p_3), (t_3, p_4), (t_3, p_5), (t_4, p_3), (t_5, p_1), (t_6, p_5)\}$ , e o conjunto dos pesos dos arcos  $w(p_1, t_1) = 1, w(p_1, t_2) = 1, w(p_2, t_2) = 1, w(p_2, t_3) = 2, w(p_2, t_5) = 1, w(p_4, t_5) = 1, w(p_5, t_4) = 1, w(p_5, t_6) = 1, w(t_1, p_1) = 1, w(t_1, p_2) = 1, w(t_2, p_3) = 1, w(t_3, p_3) = 1, w(t_3, p_4) = 1, w(t_4, p_3) = 1, w(t_5, p_1) = 1, w(t_6, p_5) = 1,.$

Note que como a transição  $t_4$  e  $t_6$  não possuem lugares de entrada, então os eventos correspondentes a  $t_4$  e  $t_6$  acontecem de forma incondicional. Em contraste, o evento correspondente à transição  $t_2$ , depende de certas condições relacionadas aos lugares  $p_1$  e  $p_2$  para que possa ocorrer.

Voltando à idéia de que as transições em um grafo de redes de Petri representam os eventos que fazem a evolução de um SED e que os lugares descrevem as condições sob as quais esses eventos podem ocorrer, tornam-se, então, necessários mecanismos que identifiquem se essas condições foram, de fato, satisfeitas ou não. Isto é feito atribuindo-se “fichas” aos lugares para indicar que a condição descrita por aquele lugar é satisfeita. Essas fichas definem uma marca. Formalmente, uma marcação,  $x$ , de um grafo da rede de Petri  $(P, T, A, w)$  é uma função  $x : P \rightarrow \mathbb{N} = \{0, 1, 2, \dots\}$ .

Assim, a marcação de  $x$  define um vetor linha  $x = [x(p_1), x(p_2), \dots, x(p_n)]$ , sendo  $n$  o número de lugares na rede de Petri e a  $i$ -ésima entrada deste vetor indica o número de fichas no lugar  $p_i$ ,  $x(p_i) \in \mathbb{N}$ . Em um grafo da rede de Petri, uma ficha é indicada por um ponto escuro posicionado no interior do círculo que define o lugar.

**Definição 3.2.2** (Rede de Petri marcada [CL99]). Uma rede de Petri marcada é uma Quintupla  $(P, T, A, w, x)$ , sendo  $(P, T, A, w)$  um grafo da rede de Petri e  $x$  é a marcação de um conjunto de lugares  $P$ , isto é,  $x = [x(p_1), x(p_2), \dots, x(p_n)] \in \mathbb{N}^n$  é o vetor linha associado a  $x$ .

Para ilustrar o conceito acima, considere a rede de Petri representada pelo grafo da Figura (3.7). Nessa figura estão ilustradas duas possíveis marcações, correspondentes aos vetores linha  $x_1 = [2, 0]$  e  $x_2 = [3, 1]$ .

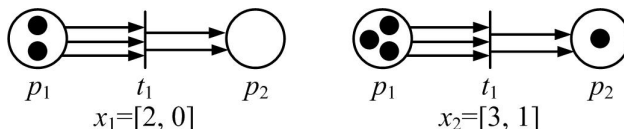


Figura 3.7: Duas marcações,  $x_1$  e  $x_2$ , ao gráfico de rede de Petri.

*Observação 3.2.1.* (a) Por simplicidade, uma rede de Petri marcada será, de agora em diante, referida simplesmente como Rede de Petri; (b) O número de fichas atribuídas a um lugar é um número inteiro arbitrário e não negativo. Segue-se, então, que o número de estados que se pode ter é, em geral, infinito. Assim, o espaço de estado  $X$ , de uma rede de Petri, com  $n$  lugares é definido por todos os vetores  $n$ -dimensionais, isto é,  $X = \mathbb{N}^n$ .

Enquanto o termo “marcação” é mais comum que “estado” na literatura sobre rede de Petri, o termo estado é consistente com o papel de estado em sistemas dinâmicos.

### 3.2.2 Evolução Dinâmica das Redes de Petri

A definição (3.2.2) não descreve explicitamente os mecanismos de transição das redes de Petri, embora este seja um ponto crucial na utilização das redes de Petri para modelar SED dinâmico. Para tanto a seguinte definição é necessária.

**Definição 3.2.3** (Transição habilitada [CL99]). Uma transição  $t_j \in T$  em uma rede de Petri é dita estar habilitada se  $x(p_i) \geq w(p_i, t_j)$  para todo  $p_i \in I(t_j)$ , sendo que  $I(t_j)$  denota o conjunto de lugares conectados à transição  $t_j$  por meio de arcos.

De acordo com a definição (3.2.3) a transição  $t_j$  na rede de Petri estará habilitada quando o número de fichas no lugar  $p_i$  é maior ou igual ao peso da seta que conecta  $p_i$  a  $t_j$ , para todo lugar  $p_i$  que são entradas para a transição  $t_j$ . Note na Figura (3.7) que para o estado  $x_1$ ,  $x(p_1) = 2 < w(p_1, t_1) = 3$ , e, portanto,  $t_1$  não está habilitada. Por outro lado para o estado  $x_2$ , tem-se  $x(p_1) = 3 = w(p_1, t_1)$  e então,  $t_1$  está habilitada.

O mecanismo de transição de estado em redes de Petri é dado pelo movimento das fichas nas conexões da rede, conseqüentemente, pela mudança do estado da rede de Petri. Quando uma transição está habilitada, diz-se que ela pode disparar. A função de transição de estado de uma rede de Petri é definida por meio da mudança no estado da rede de Petri devido ao disparo de uma transição habilitada. Algumas vezes pode haver diversas transições habilitadas e poder-se-ia pensar em disparos simultâneos. No presente trabalho, será suposto que os disparos acontecem um de cada vez.

**Definição 3.2.4** (Dinâmicas da rede de Petri [CL99]). A função de transição de estado,  $f : \mathbb{N}^n \times T \rightarrow \mathbb{N}^n$ , da rede de Petri  $(P, T, A, w, x)$  é definida pela transição  $t_j \in T$  se e somente se

$$x(p_i) \geq w(p_i, t_j) \quad \text{para todo } p_i \in I(t_j). \quad (3.1)$$

Se  $f(x, t_j)$  é definida, então o próximo vetor de estados  $x' = f(x, t_j)$  é definido como

$$x'(p_i) = x(p_i) - w(p_i, t_j) + w(t_j, p_i), \quad i = 1, \dots, n. \quad (3.2)$$

A condição (3.1) assegura que a função de transição de estado seja definida unicamente por transições que são habilitadas. Assim, o próximo estado, definido pela condição (3.2) depende explicitamente dos lugares de entrada e de saída de uma transição e dos pesos dos arcos que conectam esses lugares à transição. É importante observar que o número de fichas não precisa necessariamente ser conservado após o disparo de uma transição em uma rede de Petri. Em geral, é inteiramente possível que depois de vários disparos de transição, o estado resultante seja  $x = [0, \dots, 0]$ , ou que o número de fichas em um ou mais lugares cresça arbitrariamente depois de um número arbitrariamente grande de disparos de transição.

Para ilustrar o processo de disparo de transições e mudanças de estado de uma rede de Petri, considere a rede de Petri da Figura (3.8a), onde o estado “inicial” é  $x_0 = [1, 1, 0, 1]$ . É possível verificar que a única transição habilitada é  $t_1$ , uma vez que ela requer uma única ficha do lugar  $p_1$  e tem-se  $x_0(p_1) = 1$ . Em outras palavras,  $x_0(p_1) = w(p_1, t_1)$ , e a condição (3.1) é satisfeita para a transição  $t_1$ . Quando  $t_1$  dispara, uma ficha é removida de  $p_1$ , e uma ficha é colocada no lugar de  $p_2$ , como pode ser visto no gráfico da rede de Petri (3.8b), tem-se  $x_1(p_2) = 2$  e  $x_1(p_1) = 0$ . Esse mesmo resultado poderia ser obtido também aplicando-se diretamente a Equação (3.2) para obter

o novo estado  $x_1 = [0, 2, 0, 0]$ , como ilustrado na Figura (3.8b). Neste estado, a única transição habilitada é  $t_2$ .

Considere, agora, o disparo da transição  $t_2$ . Duas fichas são removidas do lugar de entrada,  $p_2$ . Como o lugar de saída é  $p_3$ , então, uma ficha é colocada no lugar  $p_3$ . O novo estado é  $x_2 = [0, 0, 1, 0]$ , como ilustrado na Figura (3.8c). Neste estado,  $t_1$  e  $t_2$  já não estão habilitados, mas  $t_3$  é habilitada.

Após o disparo da transição  $t_3$ . Uma ficha é removida do lugar de entrada,  $p_3$ , e é colocado no lugar de saída, que é  $p_4$ . O novo estado é  $x_3 = [0, 0, 0, 1]$ , como ilustrado na Figura (3.8d). Neste estado,  $t_1, t_2, t_3$  e  $t_4$  não estão habilitadas. Têm-se que  $x_3(p_1) < w(p_1, t_1)$ ,  $x_3(p_2) < w(p_2, t_2)$ ,  $x_3(p_3) < w(p_3, t_3)$  e  $x_3(p_4) = w(p_4, t_1)$ , neste estado a transição  $t_1$  não está habilitada pois somente uma das condições para seu disparo é satisfeita ( $x_3(p_4) = w(p_4, t_1)$ ), mas a outra condição não está ( $x_3(p_1) < w(p_1, t_1)$ ). Como a rede de Petri não possui transições de entrada para que possam ser adicionadas fichas ao sistema, assim as vê-se que nenhuma transição está habilitada e, assim, nenhuma mudança de estado adicional é possível, isto é, o estado  $[0, 0, 0, 1]$  é um estado de “trancamento definitivo” desta rede de Petri.

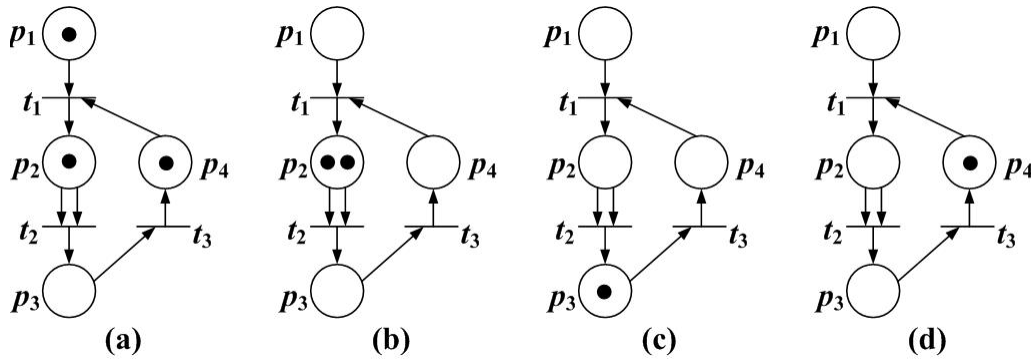


Figura 3.8: Sequências de disparos de transições em uma rede de Petri.

Uma observação importante sobre o comportamento dinâmico de redes de Petri é que nem todos os estados em  $\mathbb{N}^n$  podem necessariamente ser alcançados em um gráfico da rede de Petri com um dado estado inicial. Por exemplo, ao examinar o grafo da Figura (3.7), tem-se que para o estado inicial  $x_2 = [3, 1]$ , o único estado que pode ser alcançado de  $x_2$  é  $[0, 4]$ . Isto leva à definição de conjunto de estados alcançáveis,  $R[(P, T, A, w, x)]$ , da rede de Petri  $(P, T, A, w, x)$ . Neste sentido, primeiramente, é necessário estender a função de transição de estado  $f$  do domínio  $\mathbb{N} \times T$  ao domínio  $\mathbb{N}^n \times T^*$ , isto é:

$$\begin{aligned} f(x, \varepsilon) &:= x, \\ f(x, st) &:= f(f(x, s), t) \quad \text{para } s \in T^* \text{ e } t \in T. \end{aligned}$$

Sendo que o símbolo  $\varepsilon$  é interpretado como a ausência de disparo de transição e o operador  $(*)$  é denominado de estrela de Kleene.

**Definição 3.2.5** (Estados Alcançáveis [CL99]). O conjunto de estados alcançáveis da rede de Petri

$(P, T, A, w, x)$  é  $R[(P, T, A, w, x)] := \{y \in \mathbb{N}^n : \exists s \in T^*(f(x, s) = y)\}$ .

### 3.2.3 Equações de Estado

Considere novamente a Equação (3.2), que descreve como o valor de estado de um lugar individual muda com o disparo de uma transição. Não é difícil ver que é possível gerar um sistema de equações a partir da Equação (3.2) para obter o próximo estado da rede de Petri  $x' = [x'(p_1), x'(p_2), \dots, x'(p_n)]$  a partir do estado atual  $x = [x(p_1), x(p_2), \dots, x(p_n)]$  dado que uma transição particular,  $t_j$ , tenha disparado. Para tanto, deve-se, primeiro, definir o vetor disparo  $u$ , isto é um vetor linha de dimensão  $l$  da forma  $u = [0, \dots, 0, q, 0, \dots, 0]$ , onde um único 1 aparece na  $j$ -ésima posição,  $j \in \{1, \dots, l\}$ , para indicar o fato que a  $j$ -ésima transição está, neste momento, disparando. Além disso, define-se a matriz de incidência  $A$  de uma rede de Petri, uma matriz  $l \times n$  cujo elemento  $(j, i)$  é da forma

$$a_{ji} = w(t_j, p_i) - w(p_i, t_j). \quad (3.3)$$

Usando a matriz de incidência  $A$ , pode-se agora obter a seguinte equação de estados

$$x' = x + uA, \quad (3.4)$$

que descreve o processo de transição de estado como resultado de uma “entrada”  $u$ , isto é, uma transição particular disparando. O  $i$ -ésimo elemento da Equação (3.4) é precisamente a Equação (3.2). Portanto,  $f(x, t_j) = x + uA$ , sendo  $f(x, t_j)$  a função de transição definida anteriormente. O argumento  $t_j$  nesta função indica que a  $j$ -ésima entrada em  $u$  é, não zero. A equação de estado fornece uma ferramenta algébrica conveniente e uma alternativa à análise gráfica para descrever o processo de transições após disparos e mudanças de estado de uma rede de Petri.

Para ilustrar a evolução de um SED a partir das equações de estado, considere a rede de Petri da Figura (3.8a), com o estado inicial  $x_0 = [2, 0, 0, 1]$ . Pode-se, primeiramente, escrever a matriz de incidência por inspeção do gráfico da rede de Petri, que neste caso é:

$$A = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix}.$$

A entrada  $(1, 2)$ , por exemplo, é dada por  $w(t_1, p_2) - w(p_2, t_1) = 1 - 0$ . Usando a Equação (3.4), a equação de estado quando a transição  $t_1$  dispara no estado  $x_0$  é

$$x_1 = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 \end{bmatrix} = \begin{bmatrix} -1 & 1 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & -1 \end{bmatrix},$$

$$x_1 = \begin{bmatrix} 2 & 0 & 0 & 0 \end{bmatrix} + \begin{bmatrix} -1 & 1 & 1 & 0 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \end{bmatrix},$$

que é precisamente o que foi obtido no exemplo de Figura (3.8b). Similarmente, os outros estados também podem ser obtidos.

### 3.2.4 Modelos de Redes de Petri para Sistemas com Filas

Considere o grafo da Figura (3.9) em que três eventos ou transições dirigem um sistema com filas, quais sejam: chegada de cliente ( $a$ ), começo do serviço ( $s$ ) e serviço completo e partida do cliente ( $c$ ). A partir desses eventos é possível formar o conjunto de transição  $T = \{a, s, c\}$ . A transição  $a$  é espontânea e não requer condições (lugares de entrada). Por outro lado, a transição  $s$  conta com duas condições: a presença de clientes na fila, e que o servidor esteja inativo. Essas duas condições serão representadas por dois lugares de entradas para esta transição, lugar  $Q$  (fila) e lugar  $I$  (servidor inativo). Finalmente, a transição  $c$  requer que o servidor esteja ocupado, assim será introduzido um lugar de entrada  $B$  (servidor ocupado) para isto. Assim, o conjunto de lugares desse sistema é  $P = \{Q, I, B\}$ .

O gráfico da rede de Petri completo, junto com o simples modelo de sistema de fila, é ilustrado nas Figuras (3.9a) e (3.9b). Nenhuma ficha é colocada em  $Q$ , indicando que a fila está vazia, e uma ficha é colocado em  $I$ , indicando que o servidor está inativo. Isto define o estado inicial  $x_0 = [0, 1, 0]$ . Uma vez que a transição de estado  $a$  está sempre habilitada, é possível gerar vários caminhos de amostra possíveis. Como um exemplo, a Figura (3.9c) mostra o estado  $[2, 0, 1]$  resultante do disparo da sequência de transições  $\{a, s, a, a, c, s, a\}$ . Este estado corresponde a dois clientes esperando na fila, enquanto um terceiro está em serviço (a primeira chegada na sequência já tem partido depois da transição  $c$ ).

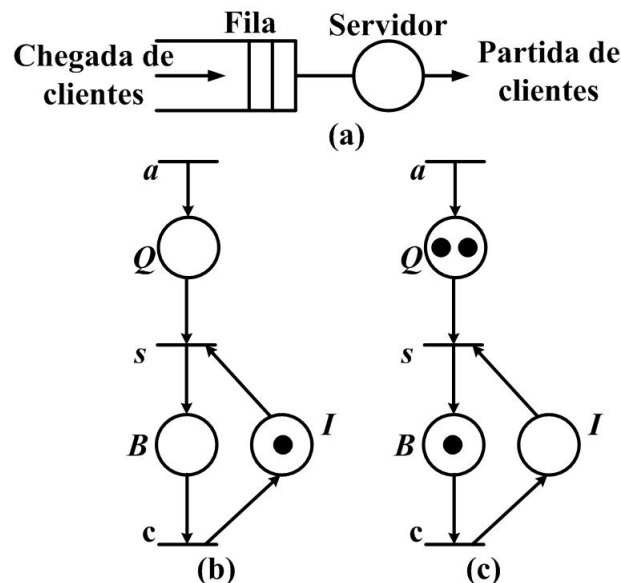


Figura 3.9: (a) Simples sistema de fila, (b) Modelo de rede de Petri para um sistema de fila simples com estado inicial  $[0, 1, 0]$ . (c) Modelo de rede de Petri de um sistema de fila simples com estado inicial  $[0, 1, 0]$  depois de disparada a sequência  $\{a, s, a, a, c, s, a\}$  [CL99].

### 3.2.5 Redes de Petri Limitada

Existem redes de Petri nas quais os lugares são limitados em um número máximo de fichas. Nesse caso, a limitação é uma propriedade inerente a rede de Petri. Apesar disso, redes de Petri podem ser definidas sem limitações como uma propriedade estrutural. Uma rede de Petri limitada não estruturada é  $k$ -limitada se não existe estado possível no qual algum lugar possa conter mais do que  $k$ -fichas. Uma rede de Petri é “segura” se ela é 1-limitada. A Figura (3.10) ilustra um trecho de uma rede de Petri em que os lugares podem receber uma quantidade  $k$ -fichas, ou seja, uma rede ilimitada. Limitações de certos lugares em uma rede de Petri limitada podem ser feitas

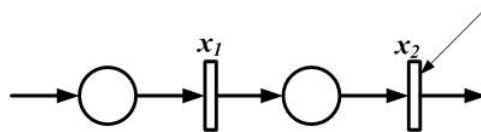


Figura 3.10: Trecho de uma rede de Petri ilimitada.

em uma rede não-limitada ao introduzir-se um lugar entre as transições em que se deseja controlar a quantidade de fichas. A figura (3.11a) ilustra a introdução de um lugar entre as transições  $x_1$  e  $x_2$  em um trecho da rede de Petri. Pode-se verificar que a transição  $x_1$  está habilitada para disparar. Como os pesos dos arcos são iguais a 1, o disparo da transição consumirá uma ficha do lugar  $P_1$  e uma ficha do lugar  $P_r$  conforme ilustra a Figura (3.11b). Nessa situação a transição  $x_1$  não estará mais habilitada para disparar, pois o lugar  $P_r$  não possui nenhuma ficha que habilite  $x_1$ , e o lugar  $P_2$  permanecerá com a única ficha que recebeu do disparo de  $x_1$ . Após o disparo da transição  $x_2$  (Figura 3.11c), o lugar  $P_r$  receberá uma ficha, então habilitando novamente a transição  $x_1$  para um novo disparo (Figura 3.11d). Esse procedimento garante que o lugar  $P_2$  fique limitado a receber no máximo uma ficha. Esse tipo de limitação controla a quantidade de fichas que o lugar  $P_2$  pode receber, e pode ser aplicada em todos os lugares da rede de Petri, tornando a rede limitada.

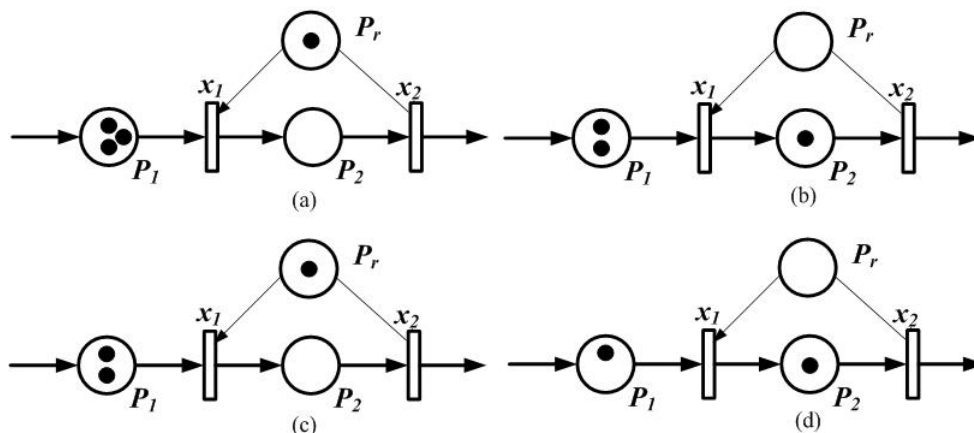


Figura 3.11: Rede de Petri com lugar seguro,  $P_2$ .



### 3.3 Modelos Temporizados

O estudo de modelos temporizados de SED será restrita a uma descrição de entrada que é completamente especificada, a fim de se ter uma compreensão clara das dinâmicas de um SED, com eventos básicos contendo tempo, independente da caracterização probabilística de sua entrada. Não serão aprofundados os estudos com respeito a redes de Petri  $T$ -temporizadas, pois o modelo a ser estudado nessa tese está restrito a uma subclasse das redes de Petri chamada de grafos de eventos temporizados.

#### 3.3.1 Redes de Petri Temporizadas

Os modelos de Redes de Petri temporizados são basicamente constituídos de dois tipos de temporização, uma associada aos lugares e outra temporização associada às transições.

##### Rede de Petri $T$ -temporizada

É uma rede de Petri em que a cada transição está associado um único parâmetro temporal (sua duração de disparo). Um tempo, possivelmente de valor diferente de zero, é associado com cada transição. Desde que uma transição torna-se habilitada, seu disparo acontecerá depois de decorrido o tempo de temporização da transição. Após o disparo as fichas dos lugares de entrada da transição serão transportadas para os lugares de saída da transição. Quando o tempo de temporização da transição é igual a zero, o disparo da transição ocorrerá tão logo a transição esteja habilitada, disparo instantâneo.

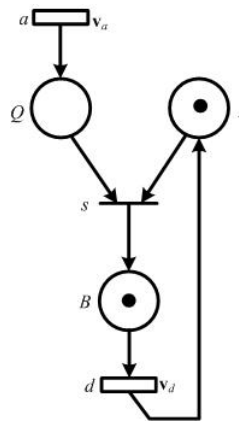


Figura 3.12: Modelo de uma rede de Petri dependente do tempo de um SED [CL99].

##### Rede de Petri $P$ -Temporizada

Diferentemente do modelo  $T$ -temporizado, uma rede de Petri  $P$ -temporizada associa a cada lugar uma temporização. Aqui o entendimento é que quando uma ficha é depositada no lugar, a mesma deverá permanecer no mínimo um determinado período de tempo neste lugar (esta ficha é dita ser indisponível por este tempo). Após esse tempo as fichas então se tornam disponíveis. Somente fichas disponíveis são consideradas para habilitar transições. Essa Rede de Petri Temporizada será definida formalmente pois ela é a ferramenta gráfica utilizada na modelagem dos sistemas

tratados nesta tese.

**Definição 3.3.1** (Rede de Petri  $P$ -temporizada [Wan98]). Uma rede de Petri  $P$ -temporizada é uma dupla  $(R, D)$  sendo  $R$  a rede de Petri e  $D$  uma função que associa a cada lugar  $p_i \in P$  um número real não negativo, sendo  $D(p_i) = d_i$  o valor de tempo associado ao lugar.

### 3.4 Grafos de Eventos Temporizados

Os Grafos a Eventos Temporizados são uma subclasse de redes de Petri apropriados para modelar SED os quais apresentam sincronia de tarefas. O comportamento dinâmico dos GET é usualmente descrito por equações não-lineares que envolvem os operadores “max” e “+” (ou “min” e “+”) na álgebra convencional. Um fato interessante é que esse comportamento não-linear dos GET é linearizado quando se utiliza uma álgebra conhecida como álgebra max-plus.

As Redes de Petri podem ser aplicadas na modelagem de sistemas que envolvam a concorrência por recursos e a sincronização de tarefas. Essas situações são ilustradas na Figura (3.13). A figura

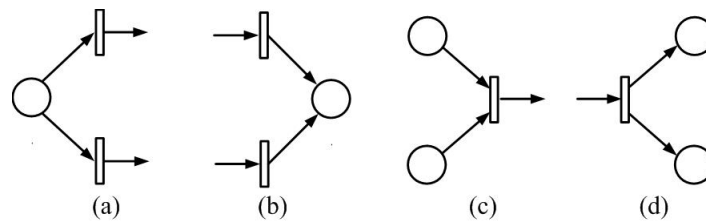


Figura 3.13: Configurações para fluxo de fichas numa Rede de Petri.

mostra as situações prováveis existentes em um fluxo de fichas em uma Rede de Petri. O fenômeno de concorrência é ilustrado na Figura (3.13) (a) e (b), e o fenômeno de sincronização é ilustrado na Figura (3.13) (c) e (d). A condição ilustrada em (a) modela a concorrência ao fornecimento de matéria-prima para um lugar. Já a concorrência ao consumo de fichas é ilustrado na figura (b), o que modela, por exemplo, a situação em que diversos processadores “disputam” um recurso. Em (c) é mostrada a sincronização ao consumo de insumos de diversos lugares, e em (d) é ilustrada a sincronização ao fornecimento de insumos. Essas situações podem ocorrer, por exemplo, quando um processador necessita que diversas informações estejam disponíveis em uma mesma data para iniciar o seu processamento, ou quando um processador fornece, na mesma data, recursos a outros processadores.

Os grafos de eventos temporizados são uma subclasse de Redes de Petri, na qual as situações de concorrência não podem existir (Figura 3.13 (a) e (b)). Isto significa que as situações de concorrência foram modificadas, ou estabelecidos critérios para impedir a ocorrência das mesmas. Estruturalmente, um GET possui somente um arco de entrada e somente um arco de saída para cada lugar. A Temporização de uma Rede de Petri é importante quando se deseja avaliar o desempenho de um SED, como por exemplo, determinar a taxa de produção do sistema ou verificar se o instante de saída de cada produto está de acordo com o calendário especificado. Em um GET, a temporização é feita associando-se a cada lugar um tempo de permanência para as fichas. Esses

grafos permitem modelar e analisar o desempenho de processos temporizados, como, por exemplo, processos de montagens.

**Definição 3.4.1** (Grafo de Eventos Temporizados - GET). Um grafo de eventos temporizados é uma Rede de Petri na qual cada lugar tem um tempo de espera associado e somente uma transição de entrada e somente uma transição de saída.

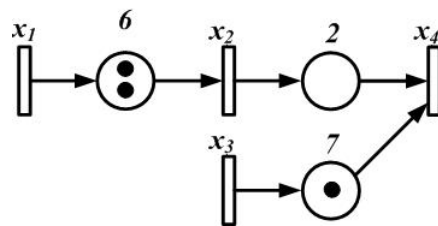


Figura 3.14: Exemplo de um GET.

A temporização de um GET pode estar associada ao tempo de disparo da transição, no entanto não há nenhuma perda de generalidade se a temporização for colocada nos denominados lugares do grafo.

**Definição 3.4.2** (Tempo de Disparo). O tempo de disparo de uma transição é o tempo que decorre entre o início e a conclusão do disparo da transição.

As fichas que serão consumidas após o disparo da transição permanecem nos seus lugares durante o tempo de disparo.

A temporização associada aos lugares pode representar o tempo de transporte ou o tempo de comunicação. Quando uma transição produz um disparo e uma ficha é colocada em um lugar, essa ficha não pode contribuir imediatamente para a habilitação das transições a jusante; ela deve primeiro passar algum tempo de permanência naquele lugar.

**Definição 3.4.3** (Tempo de permanência). O tempo de permanência em um lugar é o tempo que uma ficha deve permanecer no local antes de contribuir para a habilitação das transições a jusante.

### 3.4.1 Descrição de um GET por Datadores

É importante considerar que os GET são definidos por transições, ou seja, todos os lugares têm transições anteriores e posteriores. Isto não implica em nenhuma restrição, pois:

- Qualquer lugar de entrada tem suas fichas providas de transições externas controladas.
- Qualquer lugar de saída pode ser seguido por uma transição em que serão ativadas por fichas nos locais anteriores às transições.

As transições de entrada são denominadas de  $u_l$ , sendo  $l = 1, 2, 3, \dots$ , que são os datadores (*daters*) associados as entradas. Da mesma forma, as transições de saída serão chamadas de  $y_p$ ,

sendo  $p = 1, 2, 3, \dots$ , e por último, as transições internas serão denominadas  $x_n$ ,  $n = 1, 2, 3, \dots$ , sendo  $l$  o número de entradas do sistema,  $p$  o número de saídas do sistema e  $n$  o número de transições internas do sistema.

Uma transição dispara imediatamente após ser habilitada, ou seja, sempre que todos os lugares anteriores à transição, que possuem arcos de ligação entre eles e a referida transição, possuam ao menos uma ficha, e cada uma dessas fichas tenha cumprido o tempo de permanência estabelecido pelo lugar. Assim, as transições são disparadas tão logo seja possível.

As condições iniciais que precisam ser conhecidas antes de iniciar-se uma simulação dos disparos de um GET são:

- Os instantes de tempo nos quais as transições de entrada serão ativadas (decisões externas) durante todo o transcurso da simulação;
- Os instantes de tempo nos quais as fichas presentes dentro do sistema, antes do início da simulação, estarão disponíveis, ou seja, cumprirão seus tempos de permanência (pode-se considerar que as fichas presentes dentro do sistema já cumpriram seus tempos de permanência antes do início da simulação).

Conhecidas todas essas possibilidades, é possível determinar quando ocorrerão os disparos das transições de entrada, das transições internas e das transições de saída.

### 3.4.2 Representação das Equações Utilizando-se Datadores

Para uma transição interna chamada de  $x_i$ , uma variável correspondente  $x_i(k)$  será interpretada como o instante no qual ocorrerá o  $k$ -ésimo disparo da mesma. Desde o começo da simulação os disparos sucessivos de uma transição são numerados de maneira sequencial, a partir de um ponto de referência (geralmente zero, ou um valor negativo). Logo, a função  $k \mapsto x_i(k)$  não é decrescente (devido ao fato de que vários disparos podem ocorrer simultaneamente, pode ocorrer de não ser estritamente crescente).

O tempo pode ser medido em escala real, racional ou inteira. Por isto,  $x_i(k) \in \mathbb{R} \circ \mathbb{Q} \circ \mathbb{Z}$ . É costume omitir da composição de permutações o símbolo  $\circ$ , e simplesmente justapor os operandos. Assim  $\sigma_1 \circ \sigma_2$  é denotada por  $\sigma_1\sigma_2$ , onde entendemos que primeiro aplicamos  $\sigma_1$  e depois  $\sigma_2$ .

As equações de datadores resultam nas seguintes considerações:

- Se a transição  $x_i$  está localizada posteriormente a transição  $x_j$ , e separada desta por um lugar denominado  $P_{ij}$ , então o  $k$ -ésimo disparo de  $x_i$  consumirá a ficha produzida pelo disparo número  $k - M_{ij}$  de  $x_j$ , sendo  $M_{ij}$  o número de fichas no estado inicial marcado do lugar  $P_{ij}$  (Figura 3.15);
- Se o tempo de espera, ou tempo de permanência, neste lugar  $P_{ij}$  é  $t_{ij}$ , o disparo de número  $k$  da transição  $x_i$  não pode ocorrer até que se tenha transcorrido ao menos  $t_{ij}$  unidades de tempo desde o disparo  $k - M_{ij}$  da transição  $x_j$ ;
- Tendo em conta esta consideração para todas as transições  $x_j$  anteriores a  $x_i$ , o máximo de todos estes instantes determinará o instante do disparo número  $k$  da transição  $x_i$ .

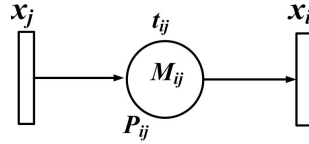


Figura 3.15: Simbologia utilizada em um GET com duas transições e um lugar.

Finalmente, seja  $p$  o índice de todas as transições anteriores a  $x_i$ . A equação fundamental de um GET é:

$$x_i(k) = \max_{j \in p} (x_j(k - M_{ij}) + t_{ij}). \quad (3.5)$$

### Exemplo

Na Figura (3.16), as transições  $x_1$  e  $x_3$  não dependem de condições iniciais para que possam disparar, essas transições são chamadas transições de entrada do GET, a transição  $x_4$  não possui um lugar para depositar fichas após o seu disparo, esse tipo de transição é considerada uma transição de saída. As transições que possuem lugares de entrada e lugares de saída são chamadas de transições internas do grafo. O comportamento dinâmico desse tipo de grafo de eventos temporizados pode ser completamente descrito utilizando-se os operadores “max” e “+”.

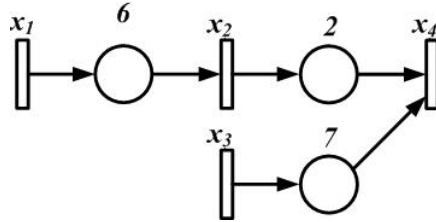


Figura 3.16: Rede de Petri  $P$ -temporizada (GET).

Após o primeiro disparo de  $x_1$  (Figura 3.16), que é  $x_1(0)$ , que acontece em um determinado instante de tempo, a transição  $x_2$  não poderá disparar imediatamente, pois a ficha deverá permanecer 6 unidades de tempo no lugar entre essas transições, então o primeiro disparo de  $x_2$  ocorrerá em  $x_2(0) = x_1(0) + 6$ . A transição de saída  $x_4$  só poderá disparar após os lugares anteriores a ela possuírem fichas e com o tempo de permanência cumprido. O disparo de  $x_4$  ocorre quando o último tempo de permanência for cumprido dos lugares anteriores à transição, ou seja,  $x_4(0) = \max\{x_2(0) + 2, x_3(0) + 7\}$ . O tempo de permanência das fichas nos lugares é interpretado em sua dinâmica como sendo um atraso.

Caso deseje-se saber qual o instante de tempo do  $k$ -ésimo disparo de uma transição, e só avaliar a evolução das suas equações dinâmicas do sistema, que são escritas da seguinte forma (lembrando que o disparo inicial acontece quando  $k = 0$ ):

$$\begin{aligned} x_2(k) &= x_1(k) + 6; \\ x_4(k) &= \max\{x_2(k) + 2, x_3(k) + 7\}. \end{aligned}$$

Considere agora a Figura (3.17). O grafo apresenta fichas nos lugares entre as transições. Nesta situação deseja-se avaliar como ocorrerá a evolução dinâmica do GET considerando que já ocorreram dois disparos da transição  $x_1$  e um disparo da transição  $x_3$ . Contudo, o instante de tempo que ocorreram é irrelevante para o desenvolvimento da dinâmica do GET, uma vez que para  $k < 0$  o valor de  $x(k) = \varepsilon$ .

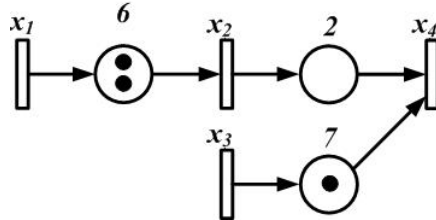


Figura 3.17: GET com fichas.

Nessa situação, a cada disparo de  $x_1$  e  $x_3$ , deve-se considerar que tais transições já tiveram seus disparos iniciais, de tal forma que:

$$\begin{aligned}x_2(k) &= x_1(k-2) + 6; \\x_4(k) &= \max\{x_2(k) + 2, x_3(k-1) + 7\}.\end{aligned}$$

O operador “max” está relacionado com a sincronização do consumo de recursos e o operador “+” com o tempo de processamento das diversas tarefas do processo. De uma maneira geral, o comportamento dinâmico de um GET pode ser descrito utilizando a álgebra max-plus, na qual o operador max é definido como  $\oplus$ , e o operador + é definido como  $\otimes$ . Pode-se reescrever o sistema anterior como:

$$\begin{aligned}x_2(k) &= x_1(k-2) \otimes 6; \\x_4(k) &= x_2(k) \otimes 2 \oplus x_3(k-1) \otimes 7.\end{aligned}$$

As equações acima ainda requerem que se especifique o estado inicial do sistema, ou seja, os valores de  $x_1(k)$  e  $x_3(k)$  para  $k < 0$ .

### 3.4.3 Inicialização

As equações anteriores são válidas sempre que as fichas consideradas sejam produzidas por disparos das transições durante a simulação. Se o número de disparos começa em  $k = 0$ , estas equações são válidas quando  $k \geq M_{ij}$ . No entanto, as equações são válidas sem restrições quando as fichas presentes nas condições iniciais do lugar não contribuem ao máximo. Isto é correto desde que a seguinte condição seja suficiente:

As fichas do estado inicial marcado, isto é, todos os disparos anteriores a  $k = 0$ , estão prontas para uso no tempo  $-\infty$ . Fala-se então de condições iniciais canônicas.

Consequentemente, a marcação inicial da rede já cumpriu o tempo de atraso de seus respectivos

lugares e pode contribuir imediatamente para a habilitação de uma transição. Note que a marcação inicial promove deslocamentos na numeração dos disparos associados a uma transição e a descrição do modelo depende desta marcação inicial. Tem-se, portanto, um sistema de equações recursivas lineares numa nova álgebra, denominada álgebra max-plus, como visto anteriormente.

#### 3.4.4 Forma Matricial

Da equação genérica (3.5), resulta uma equação geral para representar datadores de um GET, dada por:

$$x(k) = A_0x(k) \oplus A_1x(k-1) \oplus \dots \oplus B_0u(k) \oplus B_1u(k-1) \oplus \dots, \quad (3.6)$$

$$y(k) = C_0x(k) \oplus C_1x(k-1) \oplus \dots \oplus D_0u(k) \oplus D_1u(k-1) \oplus \dots, \quad (3.7)$$

sendo que,  $x(\cdot)$ ,  $u(\cdot)$  e  $y(\cdot)$  são vetores de dimensões  $n$ ,  $m$  e  $p$  respectivamente.  $A_i$ ,  $B_i$ ,  $C_i$  e  $D_i$  são matrizes de dimensões  $n \times n$ ,  $n \times m$ ,  $p \times n$  e  $p \times m$ ; o número máximo de matrizes (não nulas) é igual ao número máximo de condições iniciais em cada lugar em um GET.

A regra para formar as matrizes é a seguinte:

- O Elemento  $(r, s)$  da matriz  $A_i$  é determinado da seguinte forma: se  $r$  é uma transição interna imediatamente posterior a uma transição interna  $s$  e se existem  $i$  fichas na condição inicial do lugar  $P_{rs}$ , então  $(A_i)_{rs}$  não é nula e é igual ao tempo de espera do lugar  $P_{rs}$ . Em outras palavras, considera-se as transições do GET como sendo nós e os lugares como sendo arcos, mantendo unicamente os nós internos e os arcos internos com exatamente  $i$  fichas iniciais, então este é o grafo de precedência de  $A_i$  (com o peso dos arcos igual ao tempo de espera dos lugares correspondentes).
- Da mesma forma encontra-se  $B_i$ , considera-se que no grafo são mantidos somente os nós correspondentes às transições de entradas e arcos com exatamente  $i$  fichas iniciais entre uma transição de entrada e uma transição interna; este é o grafo de transição correspondente.
- Analogamente  $C_i$  é formada, com base no grafo que mantém os nós internos, nós de saída e arcos com exatamente  $i$  fichas iniciais entre uma transição interna e uma transição de saída, sendo este o grafo de transição da matriz  $C_i$ .
- A matriz  $D_i$  é formada de igual maneira, somente que agora são considerados unicamente nós de entrada e nós de saída, e arcos com exatamente  $i$  fichas iniciais.

As condições iniciais são  $x(k) = \varepsilon$  para todo  $k$  negativo, assumindo que o primeiro disparo das transições que modificam os estados iniciais ocorrem em  $k = 0$ .

#### 3.4.5 Forma Canônica - Eliminação da Parte Implícita

A Equação (3.6) é uma equação implícita, já que o termo  $x(k)$  está presente em ambos os lados da equação. Do ponto de vista matemático é possível solucionar essa questão (de acordo com um Teorema 2.5.6). Visto que a menor solução da Equação (3.6) nessa álgebra é dada por:

$$x(k) = A_0^*(A_1x(k-1) \oplus \dots \oplus B_0u(k) \oplus B_1u(k-1) \oplus \dots), \quad (3.8)$$

Então, esse simples argumento para eliminação da parte implícita deve ser cuidadosamente analisado, sendo, portanto, necessário analisar o que isso significaria, respondendo as seguintes questões:

1 - Tendo em conta que os estados estão confinados em  $\mathbb{R}_{max}$ , alguns elementos de  $x(k)$  podem resultar em valores iguais a  $+\infty$ . O que isso resultaria na prática?

2 - Será essa a menor solução? Qual o significado dessa seleção?

Com respeito ao fato de  $x(k)$  ser infinito, isso implicaria que o  $k$ -ésimo disparo da transição  $x_i$  irá ocorrer no tempo  $+\infty$ , ou seja, nunca ocorrerá. Logo, é razoável assumir que tais situações não podem existir. Se  $A_0$  não possui circuitos, isso conduzirá a matriz a uma forma estritamente triangular inferior, desde que exista uma apropriada numeração das transições internas, logo, a Equação (3.6) não será verdadeiramente implícita. Este é um caso óbvio quando a série de potências que define a estrela de Kleene (que será vista no capítulo seguinte) possuir um número finito de termos não nulos.

Com relação à segunda pergunta, na análise realizada, a presença de circuitos sem marcação inicial, mas com zero fichas nos lugares é um tanto controversa. Do ponto de vista prático, essas redes de Petri estão em uma situação que não devem ser consideradas como prontas para o disparo inicial. No entanto, do ponto de vista algébrico, embora a singularidade seja perdida na solução, a menor solução segue sendo dada pela dita equação. De um ponto de vista prático, os disparos das transições de um circuito sem fichas e com tempo de permanência nulo podem ser justificados assumindo-se que o sistema permite o empréstimo de fichas de fora do circuito durante o tempo zero. Logo, cada transição envolvida nesses ciclos pode ser disparada.

### 3.4.6 Equações de Estado

A etapa posterior da obtenção da forma canônica é transformar as equações para limitar os atrasos unitários somente na parte interna do sistema. Desta forma, pretende-se chegar às seguintes equações canônicas.

$$X(k) = AX(k-1) \oplus Bu(k), \quad (3.9)$$

$$Y(k) = CX(k) \oplus Du(k), \quad (3.10)$$

com novas definições para  $X$ ,  $A$ ,  $B$ ,  $C$ ,  $D$ . Esta é a manipulação padrão na teoria de sistemas, que é realizada através da incorporação do novo vetor de estados  $X$  de suficientes versões atrasadas das variáveis anteriores do vetor  $x_i$  e de  $u_j$  de tal forma que  $X(k-1)$  contenha toda a informação necessária para calcular  $X(k)$ . Com esta nova representação, o novo GET correspondente terá exatamente uma ficha na marcação inicial em todos os lugares indicados entre duas transições



internas, e não terá ficha nos lugares localizados entre transições de entrada e as transições internas, e entre transições internas e as transições de saída. A manipulação necessária para alcançar as equações canônicas será explicada durante o desenvolvimento do exemplo a seguir.

### 3.4.7 Exemplo

Considera-se o GET da Figura (3.18). A partir da mesma, é obtida a forma matricial da equação de datadores.

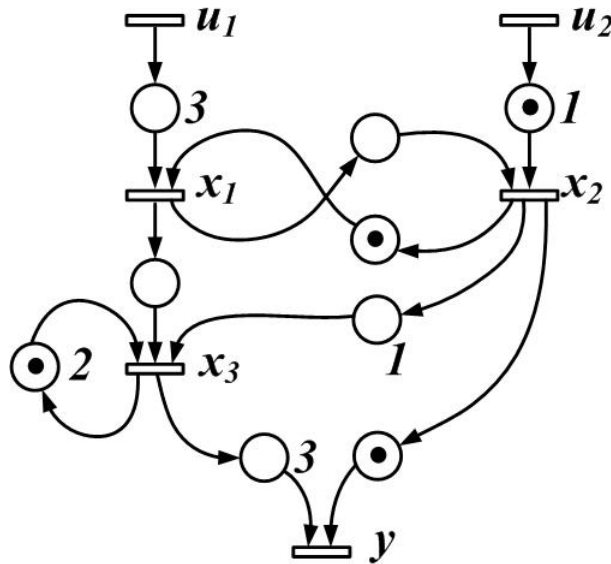


Figura 3.18: Um GET com duas entradas e uma saída.

$$\begin{aligned}
 x(k) &= \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon \\ 1 & \varepsilon & \varepsilon \\ e & 1 & \varepsilon \end{bmatrix} x(k) \oplus \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 2 \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 3 & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix} u(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & 1 \\ \varepsilon & \varepsilon \end{bmatrix} u(k-1), \\
 y(k) &= \begin{bmatrix} \varepsilon & \varepsilon & 3 \end{bmatrix} x(k) \oplus \begin{bmatrix} \varepsilon & e & \varepsilon \end{bmatrix} x(k-1).
 \end{aligned}$$

Eliminado-se a parte implícita da equação (Teorema 2.5.6, subseção 2.5.3), tem-se que:

$$\begin{aligned}
 x(k) &= A_0^* A_1 x(k-1) \oplus A_0^* B_0 u(k) \oplus A_0^* B_1 u(k-1), \\
 y(k) &= C_0 x(k) \oplus C_1 x(k-1).
 \end{aligned}$$

Assim,

$$x(k) = \begin{bmatrix} \varepsilon & e & \varepsilon \\ \varepsilon & 1 & \varepsilon \\ \varepsilon & 2 & 2 \end{bmatrix} x(k-1) \oplus \begin{bmatrix} 3 & \varepsilon \\ 4 & \varepsilon \\ 5 & \varepsilon \end{bmatrix} u(k) \oplus \begin{bmatrix} \varepsilon & \varepsilon \\ \varepsilon & 1 \\ \varepsilon & 2 \end{bmatrix} u(k-1),$$

$$y(k) = \begin{bmatrix} \varepsilon & \varepsilon & 3 \end{bmatrix} x(k) \oplus \begin{bmatrix} \varepsilon & e & \varepsilon \end{bmatrix} x(k-1).$$

### 3.5 Matrizes de Transferência com uso da Transformada- $\gamma$

#### 3.5.1 Transformada- $\gamma$

A condição inicial considerada nas equações de estado para  $k < 0$  é que  $x(k) = \varepsilon$ . Então,

$$x(k) = \bigoplus_{i=0}^k A^i B u(k-i), \quad (3.11)$$

como  $A^0 = e$ , matriz identidade. Logo,

$$y(k) = \bigoplus_{i=0}^k C A^i B u(k-i) \oplus D u(k). \quad (3.12)$$

$h(i) = C A^i B$  é definida como resposta ao impulso do sistema.

Uma tarefa na teoria de sistemas é compor sistemas para obter sistemas complexos a partir de blocos mais simples de construir. As principais operações são série, paralelo e realimentação, como é mostrado na Figura (3.19).

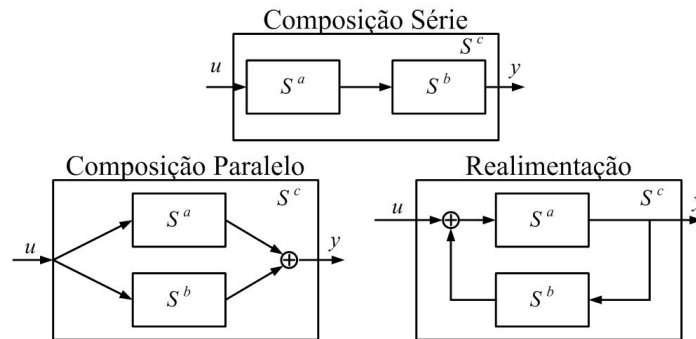


Figura 3.19: Composição de Sistema.

Realizar essas operações com as representações de entrada-saída de 3.12 não é uma tarefa muito fácil. Este é o motivo pelo qual a transformada- $z$  foi introduzida na teoria clássica de sistemas. A transformada- $\gamma$  apresentada é equivalente à transformada- $z$ , mas a primeira é aplicada para sistemas max-plus lineares. Para uma sequência  $\{u(k)\}_{k \in \mathbb{Z}}$ , a transformada- $\gamma$ , denominada  $U(\gamma)$ ,

é a série de potências formal da variável  $\gamma$  com coeficientes  $u(\cdot)$ , definida como:

$$U(\gamma) = \bigoplus_{k \in \mathbb{Z}} u(k) \gamma^k. \quad (3.13)$$

Observa-se que

$$\gamma U(\gamma) = \bigoplus_{k \in \mathbb{Z}} u(k) \gamma^{k+1} = \bigoplus_{k \in \mathbb{Z}} u(k-1) \gamma^k. \quad (3.14)$$

é a transformada- $\gamma$  de uma sequência  $\{v(k)\}_{k \in \mathbb{Z}}$  com  $v(k) = u(k-1)$ . Essa observação se traduz de forma simbólica:

$$\gamma u(k) = u(k-1), \quad (3.15)$$

sendo  $\gamma$  uma variável abstrata cujo significado é o de um operador de retardo de contagem.

Neste trabalho, são denominados de “dioide de datadores” ou simplesmente “datadores” as equações representadas sem a manipulação da transformada- $\gamma$ , e, “dioide da série de potências formal”, as equações que são manipuladas pela transformada- $\gamma$ .

### 3.5.2 Obtenção da Matriz de Transferência de um Sistema

Considerando-se as equações de estado, e considerando a transformada- $\gamma$  das variáveis  $u(\cdot)$ ,  $x(\cdot)$ ,  $y(\cdot)$  da Equação (3.9), obtém-se,

$$X(\gamma) = \gamma AX(\gamma) \oplus BU(\gamma). \quad (3.16)$$

Essa é uma equação implícita de  $X(\gamma)$  no dioide de matrizes da série de potências em  $\gamma$  com coeficientes na álgebra max-plus. Por diversas razões, é conveniente considerar álgebra max-plus completa, como  $\overline{\mathbb{Z}}_{max}$  ou  $\overline{\mathbb{R}}_{max}$ .

A menor solução da equação anterior é

$$X(\gamma) = (\gamma A)^* BU(\gamma). \quad (3.17)$$

Aplicando esse resultado em (3.10), é obtido

$$Y(\gamma) = (C(\gamma A)^* B \oplus D)U(\gamma). \quad (3.18)$$

Essa é a relação que expressa o comportamento entrada-saída do sistema, dando a transformada- $\gamma$  de saída como função da transformada- $\gamma$  de entrada do sistema.

O operador

$$H(\gamma) = C(\gamma A)^* B \oplus D. \quad (3.19)$$

é denominado a matriz de transferência do sistema. Trata-se de uma matriz cujo número de linhas é igual à dimensão do vetor de saída  $y$  do sistema e, o número de colunas é igual à dimensão do vetor de entrada  $u$ , seus elementos são séries formais de potência em  $\gamma$  com coeficientes em  $\overline{\mathbb{Z}}_{max}$  ou  $\overline{\mathbb{R}}_{max}$ .

**Exemplo**

Considere as seguintes equações recursivas:

$$x_1(k) = x_2(k-1) \oplus 3u_1(k);$$

$$x_2(k) = 1x_1(k) \oplus 1u_2(k-1);$$

$$x_3(k) = x_1(k) \oplus 1x_2(k) \oplus 2x_3(k-1);$$

$$y(k) = x_2(k-1) \oplus 3x_3(k).$$

As transformadas- $\gamma$  dessas equações recursivas são:

$$X_1 = \gamma X_2 \oplus 3U_1;$$

$$X_2 = 1X_1 \oplus 1\gamma U_2;$$

$$X_3 = X_1 \oplus 1X_2 \oplus 2\gamma X_3;$$

$$Y = \gamma X_2 \oplus 3X_3.$$

Substituindo-se  $X_1$  em  $X_2$ , obtém-se:

$$X_2 = 1\gamma X_2 \oplus 4U_1 \oplus 1\gamma U_2;$$

$$X_2 = (1\gamma)^*(4U_1 \oplus 1\gamma U_2).$$

Assim,

$$X_1 = (3 \oplus 4\gamma(1\gamma)^*)U_1 \oplus 1\gamma^2(1\gamma)^*U_2.$$

Substituindo-se  $X_1$  e  $X_2$  em  $X_3$ , tem-se que:

$$X_3 = (3 \oplus 4\gamma(1\gamma)^*)U_1 \oplus 1(1\gamma)^*(4U_1 \oplus 1\gamma U_2) \oplus 2\gamma X_3;$$

$$X_3 = (3 \oplus (1\gamma)^*(4\gamma \oplus 5))U_1 \oplus 2\gamma(1\gamma)^*U_2 \oplus 2\gamma X_3;$$

$$X_3 = (1\gamma)^*(5U_1 \oplus 2\gamma U_2) \oplus 2\gamma X_3;$$

Então:

$$X_3 = (2\gamma)^*(1\gamma)^*(5U_1 \oplus 2\gamma U_2); \quad X_3 = (2\gamma)^*(5U_1 \oplus 2\gamma U_2).$$

Substituindo-se  $X_2$  e  $X_3$  em  $Y$ , tem-se que:

$$Y = \gamma(1\gamma)^*(4U_1 \oplus 1\gamma U_2) \oplus 3(2\gamma)^*(5U_1 \oplus 2\gamma U_2).$$

$$Y = 8(2\gamma)^*U_1 \oplus 5\gamma(2\gamma)^*U_2.$$

Fatorando-se  $5(2\gamma)^*$  na expressão de  $Y$ , a matriz de transferência é obtida:

$$Y = 5(2\gamma)^*(3U_1 \oplus U_2) = (8(2\gamma)^* \quad 5(2\gamma)^*) \begin{pmatrix} U_1 \\ U_2 \end{pmatrix}.$$

Outras representações, em outros dioides, podem ser utilizadas para descrever o comportamento dinâmico de um GET [BCOQ92]. Em razão da grande semelhança entre todas essas possibilidades de representação, os SED descritos por GET e por equações em dioides são chamados simplesmente de sistemas max-plus lineares.

**3.6 Dinâmica dos GET**

A injeção de fichas, provido pelas transições de entrada do sistema, de forma indefinida e sem controle, promove o que é denominado de instabilidade do GET. Em face a isso, o controle do

fluxo de entrada de um sistema, sem prejudicar sua dinâmica e características é de fundamental importância na análise desses sistemas.

### 3.6.1 Noções de Estabilidade de GET

A idéia de estabilidade de um GET está relacionada ao número de fichas de um lugar. Diz-se que o GET é estável se o número de fichas é sempre limitado em todos os lugares internos [Mai03]. Neste trabalho, a estabilidade do sistema é garantida pelo sistema de realimentação.

**Definição 3.6.1** (Estabilidade Interna [Mai03] [Com98]). Um GET é internamente estável se, para qualquer entrada, o número de fichas de seus lugares internos permanece limitado.

A estabilidade de um GET só pode ser assegurada se todas as componentes funcionam a uma mesma taxa de produção [Mai03], caso contrário haverá acumulação de fichas nos lugares de entrada do GET. Em sistemas de manufatura a taxa de produção é determinada pela quantidade de produtos que são produzidos pela unidade de tempo, tal que:

$$\text{Taxa de Produção} = \frac{1}{\lambda}, \quad (3.20)$$

$\lambda$  é o autovalor da matriz  $A$  do sistema.  $\lambda$  também é chamado de taxa de produção, no sentido de referir-se ao intervalo de tempo de saída de cada produto acabado do sistema produtivo operando em regime permanente.

**Teorema 3.6.1** (Condições de Estabilidade para GET [Com98]). *Um GET conexo é estável se e somente se pelo menos uma das duas condições seguintes é verdadeira.*

1. *O seu subgrafo interno é fortemente conexo.*
2. *Todas as componentes de entradas fortemente conexas (CEFC's) possuem a mesma taxa de produção  $\lambda_0 \leq \lambda_j$ , sendo que  $\lambda_j$  é a taxa de produção de qualquer outro componente fortemente conexo  $C_i$  do GET.*

Esse teorema diz que uma maneira de garantir a estabilidade é assegurar que os componentes sejam fortemente conexos ou que as taxas de produção dos componentes de entradas sejam inferiores ou iguais à capacidade de processamento de cada componente do GET. Caso a taxa de entrada seja maior que a capacidade de processamento poderá haver acumulação de produtos em algum lugar do sistema [Mai03].

**Definição 3.6.2** (Controlabilidade Estrutural [Mai03] [BCOQ92]). Um grafo de eventos discretos é estruturalmente controlável se existe um caminho para toda transição interna a partir de pelo menos uma transição de entrada.

**Definição 3.6.3** (Observabilidade Estrutural [Mai03]). Um grafo de eventos é estruturalmente observável se existe um caminho de toda transição interna até pelo menos uma transição de saída.

**Teorema 3.6.2** (Estabilização via Realimentação [Mai03]). *Todo grafo de eventos estruturalmente controlável e observável pode ser estabilizado por meio de uma realimentação de saída.*

A demonstração desse teorema é baseada no fato de que todo grafo de eventos estruturalmente controlável e observável pode ser estabilizado se uma conexão efetiva via realimentação for estabelecida a partir de todas as saídas para todas as entradas. Este procedimento assegura que o GET em malha fechada é fortemente conexo [Mai03].

### 3.7 Exemplos de Aplicações

Um exemplo de grafos de eventos temporizados (redes de Petri) será ilustrado, e suas equações serão obtidas a partir de toda teoria dada até o momento.

**Exemplo 3.7.1.** Considere um GET de um sistema de manufatura (Figura 3.20) com três entradas de alimentação ( $u_1$ ,  $u_2$  e  $u_3$ ) e oito postos de trabalho. Esses oito postos de trabalho podem ser considerados como sendo máquinas em um processo produtivo. Cada máquina pode receber no máximo uma peça por vez, isso é indicado pela condição de restrição em cada posto de trabalho. Nesse sistema existem três lugares, após cada entrada de alimentação, que são considerados como estoques de entrada do processo produtivo.

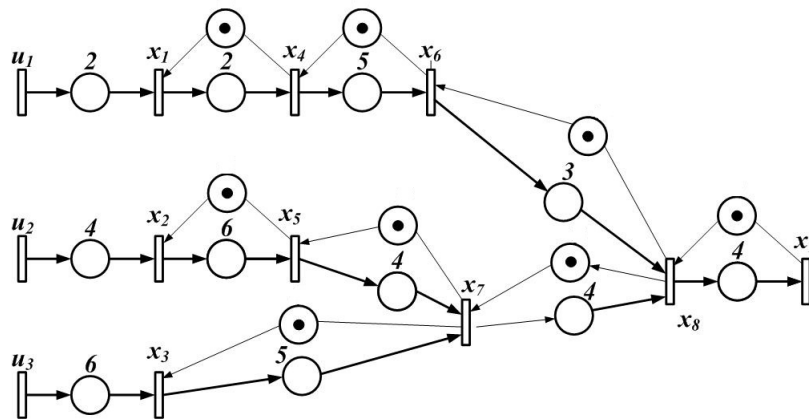


Figura 3.20: Modelo de um sistema de manufatura com 8 postos de trabalho, 3 entradas de alimentação e 8 restrições entre as transições internas.

Observa-se que  $u_1$ ,  $u_2$  e  $u_3$  são transições que não possuem condições iniciais para disparar, são transições independentes, essas transições são consideradas as transições de entrada do sistema. A transição  $x_9$  é uma transição de saída do sistema. Todas as demais transições são consideradas como transições internas do sistema.

A dinâmica interna do sistema, começando por  $x_1$ , pode ser analisada da seguinte forma. Em  $x_1$  pode-se perceber que existem dois arcos (setas) de entrada na transição, uma proveniente do lugar de saída de  $u_1$  e o outro arco vindo do lugar de uma saída de  $x_4$ , percebe-se que um dos lugares de entrada de  $x_1$  possui uma ficha de um disparo proveniente de  $x_4$ , este tipo de disparo é considerado como ocorrido no tempo  $-\infty$ . Então, a equação da transição  $x_1$  depende do maior tempo das condições iniciais para habilitarem  $x_1$  para o disparo, que é dado da seguinte forma:  $x_1(k) = \max\{u_1(k) + 2; x_4(k - 1)\}$ .

Observando o sistema por completo, todas as transições internas do sistema dependem de



$$x(k) = A_0x(k)A_1x(k-1) \oplus B_0u(k),$$

$$y(k) = C_0x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} x(k).$$

O sistema é instável, não existe controle em sua taxa de alimentação, as entradas disparam infinitas vezes, existe excesso de fichas nos lugares da rede. Considerando que os tempos no sistema estão em segundos, temos que:

- Taxa de produção do sistema ( $\frac{1}{\lambda}$ ):

$$\lambda = \bigoplus_{j=1}^n (\text{traço}(A^j))^{1/j} = 6;$$

$$\text{Taxa de produção} = 0,166 \text{ peças/segundo.}$$

- Produção do sistema em 1 hora:

$$\text{Produção} = 3600 \cdot \frac{1}{6} = 600 \text{ peças/hora.}$$

- Instantes de saída dos produtos ( $y(k)$ ):

$$y(1) = 24, \quad y(2) = 30, \quad y(3) = 36, \quad y(4) = 42, \quad y(5) = 48, \dots$$

### 3.8 Conclusão

Neste capítulo foram apresentados, entre outros, os principais conceitos sobre grafos de eventos temporizados, essa é a segunda ferramenta necessárias à compreensão das propostas de controle propostas nesta tese. Os grafos de eventos temporizados e a álgebra max-plus (vista no capítulo anterior) são as ferramentas utilizadas para a modelagem dos sistemas regidos pela ocorrência de eventos e que possuem somente fenômenos de sincronização e atraso de tempo. A dinâmica desses sistemas pode ser descrita por equações de datadores e que são tratadas como sistemas de equações lineares por meio da álgebra max-plus.

Com os grafos de eventos temporizados pode-se definir de forma gráfica como são desenvolvidas as dinâmicas das sequências de operações para cada sistema com o objetivo de obter-se um sistema modelado.

Na literatura, muitos trabalhos envolvendo controle de sistemas que utilizam a modelagem por grafos de eventos em conjunto com a álgebra max-plus. Essa ação conjunta das duas ferramentas acontece pelo uso das representação por meio de equação de datadores, obtidas de forma recursiva pela análise da dinâmica de disparos das transições em um modelo representado por um GET.

Finalmente, ao encerrar-se o capítulo, foi abordado um exemplo de GET com o propósito de encontrar as equações recursivas da dinâmicas do sistema e a forma matricial das equações por datadores. No capítulo a seguir apresentados os métodos de controle propostos para o controle do fluxo de entrada e para sincronização de transições em sistemas modelados por grafos de eventos temporizados.





## Capítulo 4

# Métodos de Controle Desenvolvidos

O presente capítulo trata, em seu conteúdo, dos resultados desta tese, a saber, os dois métodos de controle baseados na utilização de dióides, representados, por sua vez, por equações de datadores para sistemas modelados por GET. A estratégia utilizada pelos métodos, para alcançarem seus objetivos, é controlar o fluxo de entrada de matéria-prima no sistema produtivo, estabelecendo o que pode ser chamado de um calendário de entrada de material, fixando os tempos da primeira alimentação do sistema e a taxa de entrega das demais alimentações.

Esse controle do fluxo de entrada é obtido por um controlador, ou sistema de realimentação, que possui três parâmetros: (a) Taxa de alimentação, (b) Tempos de atraso nas entradas de produção e (c) Número de fichas no sistema realimentado. Com esses parâmetros, é possível atender às duas condições específicas de restrições ou controle, impostas ao sistema, abordadas nesta tese, trabalhando na política de gestão *just-in-time*.

O primeiro método de controle proposto tem por objetivo determinar o melhor controlador possível para o controle do fluxo de entrada de sistemas de manufatura modelados por GET. O melhor controlador possível é o que retarda ao máximo a entrada de matéria-prima no sistema, de tal forma que os instantes de saída dos produtos finais não sofram atrasos e atendam às especificações de demanda, além de evitar que os estoques de entrada do sistema e os estoques internos excedam suas capacidades de armazenamento, chegando ao ponto de serem denominados por *overflow*.

Nesse método, os parâmetros do controlador a serem calculados são os tempos de atraso nas entradas de produção e o número de fichas no sistema realimentado. A taxa de produção do sistema é calculada de acordo com Baccelli et al. [BCOQ92].

O segundo método de controle proposto tem por objetivo sincronizar o disparo de várias transições, impondo ao sistema condições de restrições, de tal forma que duas ou mais transições possam disparar sempre no mesmo instante de tempo ou com diferenças de tempo previamente determinadas. O objetivo é alcançado calculando-se os três parâmetros do controlador, ou seja, a taxa de alimentação, os tempos de atraso nas entradas de produção e o número de fichas no sistema realimentado. Diferentemente do primeiro método, os atrasos nas entradas de alimentação não precisam ser os máximos atrasos, mas o suficiente para garantir a sincronização. Na grande maioria dos casos, a taxa de produção do sistema é modificada, passando a operar numa taxa de produção maior.

Para o desenvolvimento dessas propostas, será utilizado como ferramenta computacional o pacote de simulação MaxPlus do Scicoslab<sup>1</sup>. Provavelmente Scilab/Scicos é atualmente a mais completa alternativa de software livre de pacotes comerciais para modelagem de sistemas dinâmicos com pacotes de simulação como o MATLAB/Simulink e MATRIXx/SystemBuild [CCN10].

Os métodos apresentados são computacionalmente eficientes no controle do fluxo de entrada de grafos de eventos temporizados, e podem ser aplicados numa ampla variedade de sistemas de produção.

A principal contribuição do primeiro método é a forma simplificada do cálculo dos tempos de atrasos nas entradas do sistema modelado e o cálculo do número de fichas do controlador, maximizando os instantes de entradas de matéria-prima no sistema de alimentação e minimizando o número de fichas em todo o sistema realimentado, a fim de garantir a estabilidade do sistema e a máxima produtividade.

Ao contrário dos resultados prévios apresentados em [DML12] e [DML13], na presente contribuição o problema e a solução são tratados mais formalmente como sendo um problema de otimização multiobjetivo, que visa minimizar o número de fichas de realimentação (número de recursos no sistema) e maximizar a data de admissão de material no sistema. É mostrado matematicamente que o controlador proposto otimiza esses dois recursos. Também é mostrado que o controlador pode ser implementado em tempo polinomial, e que o sistema de realimentação é estável. Em [DML12] foram apresentados os primeiros resultados da aplicação do método para sistema com várias entradas e uma única saída, e em [DML13] o método foi aplicado para sistemas com múltiplas entradas e múltiplas saídas.

A abordagem apresentada nesta tese tem a vantagem de evitar a manipulação e o cálculo de funções de transferência de série de potências formal [CHBF01] [MHSMC05] [MHMC03] [CLHB03] [MMLH05], procedimentos que seriam mais custosos. Do ponto de vista prático, a estrutura de controle pode ser usada em situações de acordo com as quais, um único produto é feito a partir de várias entradas de alimentação, operando com a mesma taxa de produção. Este é um caso comum, por exemplo, em uma indústria de produção de eletroeletrônicos, produzindo ao mesmo tempo, com diversas linhas de produção. Também, este método pode ser aplicado para sistemas com várias entradas e várias saídas, sem restrições de emprego. Ao longo deste capítulo será mostrado que o método é computacionalmente eficiente.

A principal contribuição do segundo método é propor um método utilizando-se datadores na sincronização do disparo de várias transições de um sistema modelado por GET. Os resultados apresentados nesta tese mostram que a sincronização é obtida de forma eficiente, e que em questão

---

<sup>1</sup>ScicosLab é um pacote de software livre de código aberto proporcionando um ambiente multi-plataforma para computação científica. É baseado no oficial Scilab 4.x (BUILD4) de distribuição, inclui também a modelagem e Scicos ferramenta de simulação (Scicos fornece um editor gráfico de diagramas de blocos para a construção e simulação de sistemas dinâmicos) e uma série de outras caixas de ferramentas, em especial a caixa de ferramentas de álgebra MaxPlus. A última versão estável do ScicosLab é ScicosLab 4.4.1 a partir de abril de 2011 [CCN10]. ScicosLab é o novo nome de ScilabGtk. Esta mudança de nome foi decidida para evitar toda a confusão com o Scilab, que não é mais desenvolvido no INRIA *Institut National de Recherche en Informatique et en Automatique*. ScicosLab é desenvolvido por alguns dos pesquisadores que originalmente desenvolveram no INRIA o Scilab.

de sincronização, é o primeiro trabalho que aborda o tema com uma quantidade tão grande de sincronizações dentro de um mesmo sistema. O método de controle pode resolver problemas comuns encontrados em muitos sistemas de produção, que é o da sincronização de atividades, eliminando-se os tempos de espera. Este tipo de controle é adequado na aplicação de processos especiais, de forma que duas ou mais atividades podem ser controladas, de modo que os seus tempos de processamento terminem em instantes de tempo determinados ou sincronizados.

A eliminação desses tempos de espera dentro do sistema pode significar a redução das perdas de matéria-prima. Nesse método, uma solução geral é apresentada para a sincronização dessas várias transições. Vários exemplos são apresentados, ilustrando a sincronização de duas, três e quatro transições internas de um sistema produtivo.

A estrutura de controle utilizada é a mesma nas duas propostas de controle, apresentando uma situação na qual um único produto é feito a partir de várias entradas e com a mesma taxa de produção, comum em uma indústria de produção de eletroeletrônicos. Dessa característica de processo é definida uma classe de GET em que os métodos são aplicados, com pressupostos e condições canônicas iniciais que serão descritos na seção seguinte.

#### 4.1 Pressupostos e Procedimentos de Modelagem

Considere um sistema de manufatura que pode ser modelado por GET, de forma que: (a) o sistema deve ter  $l$  entradas ( $u_1, u_2, \dots, u_l$ ), sendo que o número de entradas é sempre maior que um ( $l > 1$ ); (b) o número de transições internas é representado pelo índice  $n$ , de tal forma que  $x_1, x_2, \dots, x_n$  são as transições internas do sistema; (c) o sistema não deve possuir fichas nos lugares que fazem parte dos caminhos elementares que representam o fluxo do produto dentro do sistema produtivo, somente existem fichas indicando os estados ou condições de operação dos postos de trabalho e máquinas, por exemplo: máquina ociosa ou máquina em operação; (d) existem três tipos de lugares no sistema modelado por GET, o primeiro representando os postos de trabalho, o segundo representando os estoques, e o terceiro representando o status de operação dos postos de trabalho/máquinas; (e) o tempo de permanência da ficha em um lugar corresponde ao tempo de processamento de matérias-primas em máquinas, tempo de processamento em estações de trabalho, tempo de manutenção/limpeza, ou algum trabalho que é realizado na estocagem da matéria-prima; (f) o processo começa com a matéria-prima sendo alimentada pelas transições de entrada. A quantidade de fichas presentes no estado inicial do sistema representa a máxima capacidade das máquinas, ou estações de trabalho em operação. Formalmente, o tipo de sistema de manufatura de interesse (que é comum a várias áreas de interesse com aplicação na indústria) pode ser descrito por um GET. Essa classe de GET será definida como “GET máquina-estoque” ou GET máquina-buffer.

**Definição 4.1.1.** Um  $GET_{MB}$  é um a grafo de eventos temporizados no qual a capacidade de processamento das máquinas é no máximo de uma peça por vez. O sistema modelado não possui fichas nos lugares que fazem parte dos caminhos elementares que representam o fluxo do produto dentro do sistema produtivo entre cada entrada de alimentação e a saída dos produtos do sistema.

As condições iniciais para o  $GET_{MB}$  são: (a) Máquinas ociosas, (b) Estoques vazios, e (c)

Estoques com capacidade ilimitada, mas não infinita (modelo real).

A construção dos modelos utilizando máquinas e estoques (Figura 4.1) devem possuir as seguintes características:

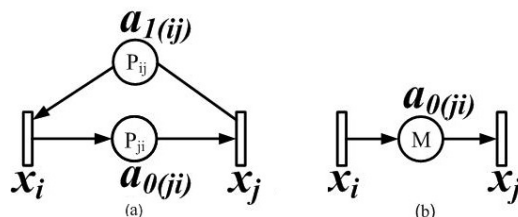


Figura 4.1: a) Máquina, e b) Estoque.

- 1 - A transição  $x_i$  representa a transição de entrada de matéria-prima nas máquinas ou postos de trabalho, e  $x_j$  representa a transição de saída da matéria-prima processada da máquina ou posto de trabalho, a mesma representação é usada para um buffer;
- 2 - Os tempos de operações das máquinas ou postos de trabalho serão representados pelos elementos de uma matriz  $A_0$ , mais especificamente  $a_{0(ji)}$ . Os status ou condição das máquinas serão representados pela matriz  $A_1$ , mais especificamente por  $a_{1(ij)}$ . Para o sistema  $GET_{MB}$  o tempo de ociosidade da máquina é especificado que seja zero, ou seja, assim que uma máquina executar uma atividade e liberar a matéria-prima estará imediatamente habilitada para receber novo material, dessa forma,  $a_{1(ij)} = 0$ , para  $i, j = 1, 2, \dots$ ;
- 3 - A quantidade de fichas nos lugares do sistema será representada por  $M, P_{ij}$  e  $P_{ji}$ ;
- 4 - Para um Buffer:  $M \in \mathbb{Q}$ , de modo que  $M = 0, 1, 2, \dots, \infty$ ;
- 5 - Para uma máquina/posto de trabalho:  $P_{ij}$  e  $P_{ji} \in \mathbb{Q}$ , sendo:  $P_{ij} \in \{0, 1, \dots\}$ , a capacidade de processamento; e  $P_{ji} \in \{0, 1, \dots\}$ , 0 - máquina sem matéria-prima, e, 1 - executando atividade. A Figura 4.1 descreve partes de um modelo utilizando as instruções de 1 a 5.

Para enumerar as transições do  $GET_{MB}$  os seguintes procedimentos devem ser adotados:

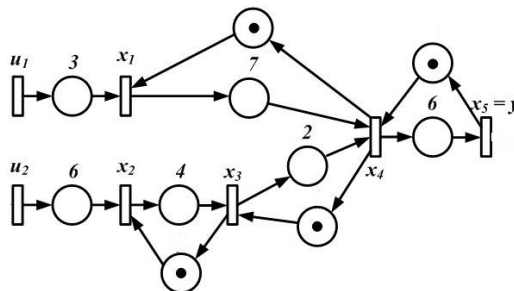


Figura 4.2: Exemplo de numeração das transições em um  $GET_{MB}$ .

- 1 - As transições de entrada do sistema devem ser representadas por:  $u_1, u_2, u_3, \dots$ ;
- 2 - As transições internas devem ser representadas por  $x_1, x_2, x_3, \dots$ , (Figura 4.2). As transições devem ser numeradas de acordo com o sentido do fluxo produtivo, de forma que uma transição predecessora ( $x_i$ , transição que indica o recebimento de matéria-prima nos lugares do sistema), deve ter o índice  $i$  menor que o índice  $j$  da transição sucessora ( $x_j$ , transição que quando disparada, libera a matéria-prima processada para o posto de trabalho seguinte). Esse procedimento de modelagem garante que as informações de operação do modelo, durante a operação matricial, numa sequência de fluxo dentro da matriz, produzam dados completos entre os instantes de disparos das transições.

Uma nota importante, os procedimentos adotados anteriormente não são baseados em qualquer trabalho existente, eles foram introduzidos neste trabalho para explicar como o modelo é construído e para enfatizar as necessidades de condições iniciais e de algumas limitações da proposta deste trabalho, ou seja, máquinas e estoques estão vazias no início do processo. A hipótese de estoques com matéria-prima ou máquinas em operação antes de ocorrer a alimentação do sistema é de alguma forma não adequada ao funcionamento prático do sistema, esse tipo de ocorrência não garante o cálculo dos atrasos máximos da entradas de alimentação. Esta é uma situação que só ocorre quando há falhas no sistema de produção.

Quando as orientações para numeração das transições são seguidas, como definido anteriormente, as matrizes  $A_0$  e  $A_1$  passam a possuir formas interessantes, elas assumem a forma de matrizes triangulares inferior e superior, respectivamente, facilitando a manipulação e ordenação dos dados. Não existem caminhos fechados (circuitos) em  $A_0$ , isto é,  $\nexists \rho(x_i, x_i) \forall i \leq n$ . Os valores da diagonal principal de  $A_1$  são atribuídos como sendo cada um igual a zero, isto é,  $a_{1(11)} = a_{1(22)} = a_{1(33)} = \dots = a_{1(nn)} = e = 0$ , essa condição do  $GET_{MB}$  é usada para indicar que a contagem dos tempos de disparo começa quando  $t = 0$  (*zero*), também garantirá que não haverá perda de informações durante as operações matriciais de  $\oplus$  e  $\otimes$ . Os elementos de  $B_0$  são os tempos de permanência da matéria-prima quando esses chegam aos estoques de entrada do sistema.

## 4.2 Método de Controle I - Maximização do atraso do fluxo de entrada

Em modelagem de sistemas utilizando-se os GET, é comum que as entradas de alimentação sejam modeladas como não tendo condições prévias ou condições iniciais para habilitar seus disparos (Figura 4.3). Pensa-se que tais transições possuem um lugar predecessor que possui um estoque infinito de fichas (recursos) e que essas fichas estão prontas para serem consumidas. Isso conduz ao problema de que as transições de entrada podem injetar fichas no sistema modelado de forma não controlada, excedendo a capacidade de estoque do sistema, que não é e nem pode ser considerado como infinito, pois não atenderia as condições reais de um sistema de manufatura, e que, segundo os conceitos de estabilidade vistos nesta tese, geram instabilidade no sistema.

Numa condição real de um ambiente fabril, isso significaria estar em um ambiente onde a possibilidade de movimentação de material é prejudicada, e que os operadores dos postos de trabalho

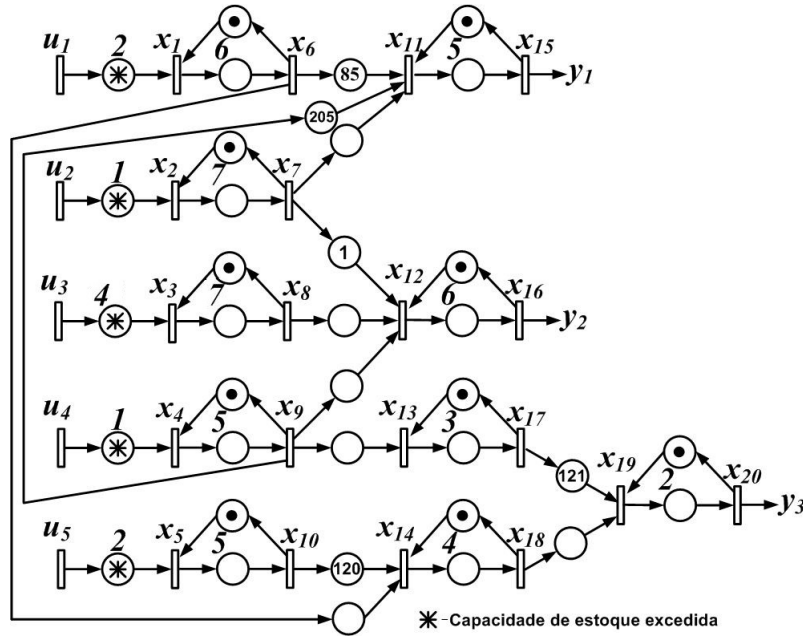


Figura 4.3: Exemplo de um sistema modelado por GET sem controle nas entradas de alimentação.

estariam sem espaço para movimentação adequada pelo número excessivo de estoque (matéria-prima) ao seu redor, comprometendo a atividade de produção e qualidade do produto.

Para o desenvolvimento do método proposto nesta tese, o sistema pode ser modelado em termos de duas variáveis de estado:  $x(k)$  e  $x(k-1)$ . Então, em seguida, o método deverá ser aplicado para sistemas mais abrangentes. O sistema de interesse será modelado pelas equações  $x(k) = A_0 \otimes x(k) \oplus A_1 \otimes x(k-1) \oplus B_0 \otimes u(k)$  e  $y(k) = C \otimes x(k)$ , sendo  $A_0 \in (\mathbb{Z}_{max})^{n \times n} \mid [A_0]_{ij} = \varepsilon \forall i = j$ , se  $\exists [A_0]_{ij}$  então  $\nexists [A_0]_{ji} \forall i \neq j$ . Os elementos de  $A_0$  são os tempos de operação das máquinas ou postos de trabalho, e  $A_1 \in (\mathbb{Z}_{max})^{n \times n} \mid [A_1]_{ij} = e$  (zero)  $\forall i = j$  (Os instantes de tempo de disparo de todas as transições começando do tempo zero, esta é a condição para garantir que não haverá perda de informações dos tempos de disparo), se  $\exists [A_1]_{ji}$  então  $\nexists [A_1]_{ij}, \forall j \neq i$ . Os elementos de  $A_1$  são os tempos ou condições de restrições das máquinas ou postos de trabalho.  $B_0 \in (\mathbb{Z}_{max})^{n \times l}$  tal que em cada coluna existe um elemento diferente de  $\varepsilon$ , esse é o tempo que a matéria-prima deve permanecer no estoque inicial do sistema.

As equações anteriores foram tratadas em [BCOQ92] e podem ser equivalentemente representadas por:

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k), \quad (4.1)$$

$$y(k) = C \otimes x(k). \quad (4.2)$$

sendo  $A = A_0^* \otimes A_1$ ,  $B = A_0^* \otimes B_0$  e  $A_0^* = I \oplus A_0 \oplus A_0^{\otimes 2} \oplus A_0^{\otimes 3} \oplus \dots$  ( $I$  é a matriz identidade e o operador  $(*)$  é chamado de estrela de Kleene. Demonstrações podem ser encontradas em

[BCOQ92]).

As seguintes condições iniciais canônicas são assumidas:

- (1)  $x(k) = \varepsilon$  e  $y(k) = \varepsilon \quad \forall \quad k < 0$ .
- (2) Não existem estados marcados ou fichas nos lugares que são considerados como postos de trabalhos ou estoques do sistema quando  $k < 0$ .
- (3) Como consequência direta dos pressupostos assumidos, assume-se que no início do processo, as fichas presentes no sistema indicam o estado das máquinas (uma ficha, indicando que a máquina está ociosa).

Substituindo-se as condições iniciais canônicas  $x(k) = \varepsilon, \forall k < 0$ , na Equação (4.1), resulta, para  $k = 0$ , em:  $x(0) = A_0^* \otimes A_1 \otimes x(-1) \oplus A_0^* \otimes B_0 \otimes u(0)$ , desde que  $x(-1) = \varepsilon$ , então  $x(0) = A_0^* \otimes B_0 \otimes u(0)$  e  $y(0) = C \otimes A_0^* \otimes B_0 \otimes u(0)$ .

Como resultado, a saída mais rápida do primeiro produto  $y_r(0)$  para  $u(0) = [0]$  é dada por:

$$y_r(0) = C \otimes A_0^* \otimes B_0 \otimes [0]. \quad (4.3)$$

Até a saída do primeiro produto, o sistema opera em regime transitório. Após a saída do primeiro produto, o sistema passa a operar em regime permanente. No regime permanente, o  $GET_{MB}$  começa a produzir peças em intervalos de tempos iguais dados por  $\beta$ , sendo  $\beta \geq \lambda$  e  $\lambda$  é o menor intervalo de tempo entre duas saídas sucessivas, calculado por [BCOQ92]:

$$\lambda = \bigoplus_{j=1}^n (\text{trace}(A^j))^{1/j}. \quad (4.4)$$

Dado  $\beta$ , as datas de produção desejadas (trajetórias de referência) devem respeitar:

$$y_r(k) = \beta^{\otimes k} \otimes y_r(0), \quad (4.5)$$

tal que  $y_r(k) = \varepsilon$  para  $k < 0$ .

**Proposição 4.2.1.** *Dado  $\beta \geq \lambda$ , um limite superior para  $u(k)$  é dado por*

$$u_M(k) = (C \otimes B) \setminus y_r(k) = \beta^{\otimes k} u_M(0). \quad (4.6)$$

*Demonstração.* : Desde que,

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k), \quad (4.7)$$

$$y(k) = C \otimes x(k) \leq y_r(k), \quad (4.8)$$



Um limite superior para  $u_M$  é obtido usando-se a Teoria da Residuação, dado por:

$$(C \otimes A \otimes x(k-1)) \oplus (C \otimes B \otimes u(k)) \leq y_r(k) \quad (4.9)$$

$$\Rightarrow u(k) \leq u_M(k) = (C \otimes B) \setminus y_r(k) \quad (4.10)$$

$$\Rightarrow u_M(k) = \beta^{\otimes k} \otimes u_M(0) \quad \forall k \geq 0. \quad (4.11)$$

□

#### 4.2.1 Síntese do Controlador

Um controlador realimentado é proposto baseado nos conceitos de estabilidade de GET [Com98]. A estrutura do controlador é ilustrada na Figura 4.4, com parâmetros definidos como:  $[\varphi]$ ,  $\beta$  e  $m$ , tal que:  $[\varphi]$  é o vetor com os tempos de atraso das entradas de alimentação do sistema;  $\beta$  é a taxa de disparo do controlador; e  $m$  é o número de fichas do controlador.

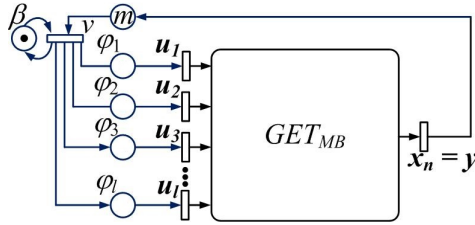


Figura 4.4: Sistema de realimentação para otimizar os estoques de entrada.

Esse controlador estabelece o controle do fluxo de entrada de material no sistema, evitando que os estoques de entrada excedam suas capacidades de armazenamento. Isto é obtido atrasando-se ao máximo a entrada de matéria-prima no sistema. Os critérios para alcançar o controlador desejado é a maximização dos tempos de entradas da matéria-prima, minimizando o estoque interno do sistema. Como resultado, o problema de controle pode ser formulado como um problema multiobjetivo:

**Maximizar  $(u, y)$  e Minimizar  $m$** , sendo  $\varphi$ ,  $\beta$ , e  $m$  os parâmetros de projeto, tal que:

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k),$$

$$y(k) = C \otimes x(k) \leq y_r(k),$$

$$u(k) = [\varphi] \otimes v(k),$$

$$v(k) = \beta \otimes v(k-1) \oplus y(k-m).$$

sendo  $v(0) = 0$  e  $y_r$  são dados pela Equação (4.5).

**Proposição 4.2.2.** Dado  $\beta \geq \lambda$ , o problema de otimização multiobjetivo tem solução ótima se  $m = \left\lceil \frac{y_r(0)}{\beta} \right\rceil$  e  $[\varphi] = (C \otimes A_0^* \otimes B_0) \setminus y_r(0)$ , sendo  $y_r(0) = C \otimes A_0^* \otimes B_0 \otimes u(0)$ , e que  $y_r(k)$  seja

definido como a saída mais rápida dos produtos (Equação 4.5).

*Demonstração.* Para demonstrar a proposição, primeiro, é mostrado que os pressupostos conduzem  $u(k)$  para um valor máximo, porque ele é igual a seu limite superior dado pela Proposição 4.2.1.

É assumido que  $v(0) = 0 \Rightarrow u(0) = [\varphi] \otimes v(0) \Rightarrow u(0) = \varphi$ , que é máximo, de acordo com a Proposição 4.2.1.

1 - O cálculo dos tempos de atraso das entradas ( $\varphi$ ) é feito para o primeiro disparo de  $u(k)$  ( $k < m$ ), para esta condição,  $y(k - m) = \varepsilon$ , então:

$$v(k) = \beta \otimes v(k - 1) \oplus y(k - m) \Rightarrow v(1) = \beta \otimes v(0) \Rightarrow v(2) = \beta \otimes v(1) = \beta^{\otimes 2} \times v(0) \Rightarrow v(k) = \beta^{\otimes k} \otimes v(0).$$

$u(k) = [\varphi] \otimes v(k) = [\varphi] \otimes \beta^{\otimes k} \otimes v(0) = \beta^{\otimes k} \otimes u(0)$ , que é máximo de acordo com a Proposição 4.2.1.

Assim, devido a estrutura proposta para o  $GET_{MB}$ , inicialmente o sistema possui estoques vazios e as máquinas estão ociosas, assim  $y(k) = \beta^k \otimes y(0)$  para  $k < m$ , desde que  $u(k) = \beta^{\otimes k} u(0)$ .

2 - O número mínimo de fichas  $m$  é calculado quando  $k = m$ , assim:

$$v(m) = \beta \otimes v(m - 1) \oplus y(0),$$

$$\text{Desde que } v(m - 1) = \beta^{\otimes(m-1)} \otimes v(0) \Rightarrow$$

$$v(m) = \beta \otimes \beta^{\otimes(m-1)} \otimes v(0) \oplus y(0),$$

$$v(m) = \beta^{\otimes m} \otimes v(0) \oplus y(0),$$

Para assegurar que o próximo disparo das transições (Controlador realimentado) é ótimo, isto é,  $u(m) = \beta^{\otimes m} u(0)$ , é necessário que  $\beta^{\otimes m} \otimes v(0) \geq y(0)$ , mas  $v(0) = 0$ , assim:  $\beta^{\otimes m} \geq y(0)$ .

$$\text{Para } k = m + 1, v(m + 1) = \beta^{\otimes(m+1)} \otimes v(0) \oplus y(1),$$

Como  $u(1) = \beta \otimes u(0)$  e devido a estrutura do  $GET_{MB}$  (estoques vazios e as máquinas estão ociosas),  $y(1) = \beta \otimes y(0)$ ,

$$v(m + 1) = \beta^{\otimes(m+1)} \otimes v(0) \oplus \beta \otimes y(0),$$

$$v(m + 1) = \beta \otimes (\beta^{\otimes m} \otimes v(0) \oplus y(0)).$$

A condição que garante  $v(m + 1) = \beta^{\otimes(m+1)} \otimes v(0)$ , tal que,  $u(m + 1)$  é máximo, é  $\beta^{\otimes m} \otimes v(0) \geq y(0)$ . Este procedimento pode ser repetido para todo  $k \geq m$ , assim a otimização é assegurada se  $\beta^{\otimes m} \otimes v(0) \geq y(0)$ . Então:

$$\beta^{\otimes m} \geq y(0) \Rightarrow m \cdot \beta \geq y(0) \Rightarrow m \geq \frac{y(0)}{\beta}.$$

Desde que  $m \in \mathbb{Z}$  so  $m = \left\lceil \frac{y_r(0)}{\beta} \right\rceil$ . Como resultado  $m$  não pode ser menor que  $\frac{y_r(0)}{\beta}$ . Isto pode ser visto observando-se que o método de controle proposto introduz um circuito de peso médio igual a  $\frac{y_r(0)}{m}$  no grafo de precedência do circuito fechado do  $GET$ . Se  $m$  é estritamente menor que  $\frac{y_r(0)}{\beta}$ , pode-se ver que este grafo terá um circuito de peso médio [BCOQ92] maior do que  $\beta$ , o que significa, na prática, que o período  $\beta$  não é possível.  $\square$

**Proposição 4.2.3.** *O método de realimentação proposto obtém, nos seus cálculos, valores, dos tempos de atrasos nas entradas, maiores ou iguais que os obtidos usando-se as técnicas por modelo de referência..*

*Demonstração.* Comparando-se os resultados, utilizando-se a transformada- $\gamma$ , tem-se que: o método de controle proposto conduz para tempos de entrada dados por  $u_{JIT}(\gamma) = H(\gamma) \backslash (H(\gamma) \otimes v)$ , enquanto que utilizando-se as técnicas por modelo de referência, as entradas são dadas por  $u_F(\gamma) = (F(\gamma) \otimes (H(\gamma))^*) \otimes v \leq (H(\gamma) \backslash (H(\gamma))) \otimes v \leq H(\gamma) \backslash (H(\gamma) \otimes v)$ .  $\square$

Como consequência, o sistema de controle proposto garante os maiores atrasos possíveis nas entradas de matéria-prima do sistema de produção, mantendo a máxima taxa de produção do sistema. O método de controle proposto conduz a maiores atrasos (ou pelo menos iguais) das entradas quando comparados aos resultados obtidos usando-se as técnicas por modelo de referência.

#### 4.2.2 Complexidade Computacional do Método I

A complexidade de um problema está relacionada com o tempo de execução do algoritmo. Um problema pode ser resolvido em tempo polinomial se existe um algoritmo para resolvê-lo no tempo  $O(n^k)$ ,  $k \in \mathbb{R}$ . Assim, antes de tudo, será introduzida uma propriedade de matrizes quadradas em dioides:

*Propriedade 4.2.1.*  $I \oplus A_0 \oplus A_0^{\otimes 2} \oplus \dots \oplus A_0^{\otimes n} = (I \oplus A_0)^{\otimes n}$ .

*Demonstração.* Usando a propriedade de idempotência de um dioide:

$$\begin{aligned} (I \oplus A_0)^{\otimes 2} &= (I \oplus A_0) \otimes (I \oplus A) = I \oplus A \oplus A^{\otimes 2}. \\ (I \oplus A_0)^{\otimes 3} &= (I \oplus A) \otimes (I \oplus A)^{\otimes 2} = I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3}. \\ (I \oplus A_0)^{\otimes 4} &= (I \oplus A) \otimes (I \oplus A_0)^{\otimes 3} = I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus A^{\otimes 4}. \\ &\vdots = \vdots \\ (I \oplus A_0)^{\otimes n} &= I \oplus A \oplus A^{\otimes 2} \oplus A^{\otimes 3} \oplus A^{\otimes 4} \dots \oplus A^{\otimes n}. \end{aligned}$$

$\square$

*Propriedade 4.2.2.* O método proposto tem complexidade polinomial.

*Demonstração.* As seguintes observações são importantes para obter a complexidade polinomial do algoritmo:

1. Para uma matriz quadrada  $A$   $n \times n$ , a complexidade da multiplicação é  $O(n^4)$ .
2. A complexidade da residuação de duas matrizes  $A$  ( $n \times n$ ) e  $B$  ( $n \times l$ ) é igual a  $O(n \times n \times l)$ .

Como  $n$  é muito maior que  $l$ , portanto, usando a propriedade 4.2.1, a complexidade da operação  $(A_0^* \otimes B_0) \backslash (C \otimes A_0^* \otimes B_0 \otimes u(0))$  é  $O(n^4)$ .

Logo, este algoritmo pode ser executado em tempo polinomial.  $\square$

### 4.2.3 Exemplo Ilustrativo: Comparação entre os Resultados do Método I e do Modelo de Referência

No exemplo a seguir, os resultados do método proposto podem ser comparados com os resultados obtidos usando-se as técnicas de realimentação por modelo de referência, tal como proposto por Cottenceau et al. [CHBF01].

Considere um sistema de manufatura com três entradas de alimentação, três estoques de entrada, quinze postos de trabalho ( $w_1, w_2, \dots, w_{15}$ ) e cinco condições de restrições nos postos de trabalho (Figura 4.5). O objetivo deste exemplo é encontrar o máximo atraso possível para as entradas de alimentação, sem comprometer a demanda de produção do sistema. Para solucionar o problema, será aplicado o método proposto para determinar os parâmetros do controlador de realimentação que garantem o máximo atraso nas entradas.

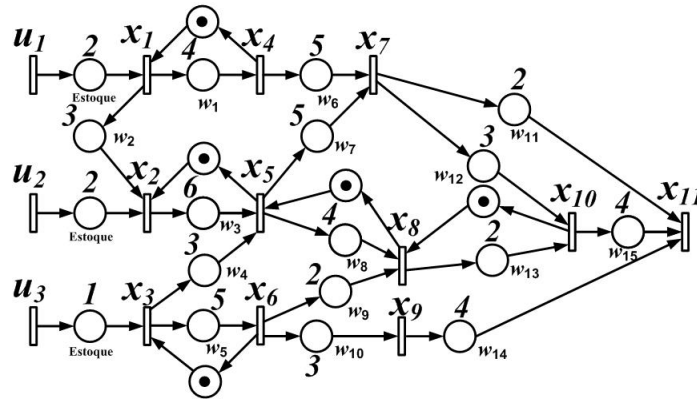


Figura 4.5: Sistema de manufatura proposto.

As equações recursivas são obtidas por inspeção do GET, de tal forma que:

$$\begin{aligned}
 x_1(k) &= 2 \otimes u_1(k) \oplus x_4(k-1); \\
 x_2(k) &= 2 \otimes u_2(k) \oplus 3 \otimes x_1(k) \oplus x_5(k-1); \\
 x_3(k) &= 1 \otimes u_3(k) \oplus x_6(k-1); \\
 x_4(k) &= 4 \otimes x_1(k); \\
 x_5(k) &= 6 \otimes x_2(k) \oplus 3 \otimes x_3(k) \oplus x_8(k-1); \\
 x_6(k) &= 5 \otimes x_3(k); \\
 x_7(k) &= 5 \otimes x_4(k) \oplus 5 \otimes x_5(k); \\
 x_8(k) &= 4 \otimes x_5(k) \oplus 2 \otimes x_6(k) \oplus x_{10}(k-1); \\
 x_9(k) &= 3 \otimes x_6(k); \\
 x_{10}(k) &= 3 \otimes x_7(k) \oplus 2 \otimes x_8(k); \\
 x_{11}(k) &= 2 \otimes x_7(k) \oplus 4 \otimes x_{10}(k) \oplus 4 \otimes x_9(k).
 \end{aligned}$$

A modelagem matricial é dada por:

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 6 & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 5 & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & 4 & 4 & \varepsilon & \varepsilon \end{bmatrix}, \quad B_0 = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & 2 & \varepsilon \\ \varepsilon & \varepsilon & 1 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$A_1 = \begin{bmatrix} e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}$$

## Resultados

Obtidas as matrizes do sistema, o próximo passo é calcular a taxa de produção do sistema, que é obtida calculando-se  $\lambda = \bigoplus_{j=1}^n (\text{traço}(A^j))^{1/j}$ ,  $A = A_0^* \otimes A_1$ . Então, para o controlador do método proposto garantir a maior taxa de produção do sistema o valor de  $\beta$  será igual ao valor de  $\lambda$ . Dessa forma, o valor obtido é  $\beta = 6$ . Em seguida são calculados os valores de  $[\varphi]$  e  $m$ .

$$[\varphi] = (C \otimes B) \setminus y(0) = (C \otimes A_0^* \otimes B_0) \setminus (C \otimes A_0^* \otimes B_0 \otimes u(0)) = [0 \quad 3 \quad 7],$$

$$m = \left\lceil \frac{y(0)}{\beta} \right\rceil = 4.$$

Esses são os parâmetros do controlador encontrados utilizando-se o método proposto, ilustrados na Figura 4.6.

Sem o controle proposto, as entradas de alimentação injetariam material no sistema de forma desordenada, excedendo a capacidade nos estoques internos do sistema, gerando a instabilidade do mesmo. Os melhores tempos de saída dos produtos são:  $y(1) = 23$ ,  $y(2) = 29$ ,  $y(3) = 35$ ,  $y(4) = 41$ ,  $y(5) = 47 \dots$

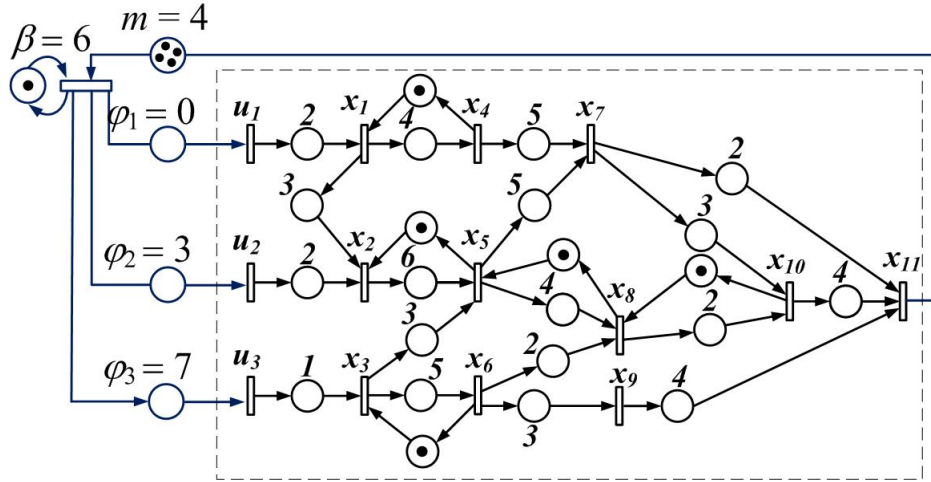


Figura 4.6: Sistema de realimentação obtido aplicando-se o método proposto.

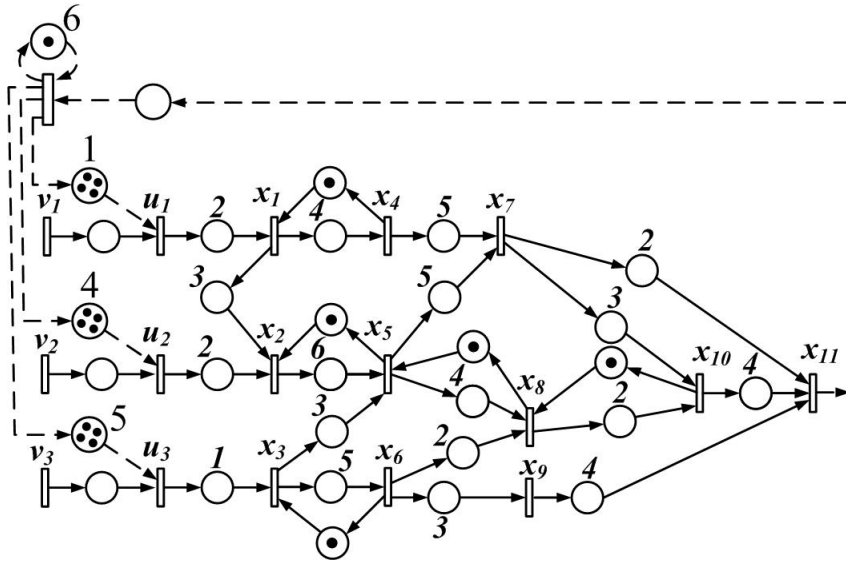


Figura 4.7: Sistema de realimentação obtido aplicando-se o método por modelo de referência [CHBF01].

Utilizando-se a notação por série de potência (utilizando-se a transformada-*gamma*), a seguir, serão comparados os dois métodos de controle.

No método proposto:

$$y_{JIT} = 23 \otimes (6 \otimes \gamma)^* \text{ (resposta rápida, toda matéria-prima está disponível em } t = 0\text{)}.$$

No modelo de referência:

$$H = [23 \otimes (6 \otimes \gamma)^* \quad 20 \otimes (6 \otimes \gamma)^* \quad 16 \oplus (25 \otimes \gamma)(6 \otimes \gamma)^*],$$

$$G_{ref} = H,$$

$$F_c = H \setminus G_{ref} \setminus H = [(1 \otimes \gamma^4) \otimes (6 \otimes \gamma)^* \quad (4 \otimes \gamma^4) \otimes (6 \otimes \gamma)^* \quad (5 \otimes \gamma^4) \otimes (6 \otimes \gamma)^*],$$

$$u_F = (F \otimes H)^* \otimes v = [0 \oplus (24 \otimes \gamma^4) \otimes (6 \otimes \gamma)^* \quad 0 \oplus (27 \otimes \gamma^4) \otimes (6 \otimes \gamma)^* \quad 0 \oplus (28 \otimes \gamma^4) \otimes (6 \otimes \gamma)^*],$$

$$u_{JIT} = H \setminus (H \otimes v) = [(6 \otimes \gamma)^* \quad 3 \otimes (6 \otimes \gamma)^* \quad 7 \otimes (6 \otimes \gamma)^*] \text{ (Método proposto).}$$

$u_F \oplus u_{JIT} \neq u_F \Rightarrow u_F < u_{JIT}$ , mas:  $u_F \oplus u_{JIT} = u_{JIT}$ .

Os valores de  $u_{JIT} = H \setminus (H \otimes v) = [(6 \otimes \gamma)^* \quad 3 \otimes (6 \otimes \gamma)^* \quad 7 \otimes (6 \otimes \gamma)^*]$ , dos tempos de atraso, são iguais aos valores obtidos pelo método I, dados abaixo:

$$[\varphi] = (C \otimes B) \setminus y(0) = (C \otimes A_0^* \otimes B_0) \setminus (C \otimes A_0^* \otimes B_0 \otimes u(0)) = [0 \quad 3 \quad 7].$$

Para efeitos de comparação, são mostrados os resultados obtidos pelas duas técnicas, as primeiras séries de entradas no sistema são mostradas:

$$u_F = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 24 \\ 27 \\ 28 \end{bmatrix}, \dots \quad (4.12)$$

$$u_{JIT} = \begin{bmatrix} 0 \\ 3 \\ 7 \end{bmatrix}, \begin{bmatrix} 6 \\ 9 \\ 13 \end{bmatrix}, \begin{bmatrix} 12 \\ 15 \\ 19 \end{bmatrix}, \begin{bmatrix} 18 \\ 21 \\ 25 \end{bmatrix}, \begin{bmatrix} 24 \\ 27 \\ 31 \end{bmatrix}, \dots \quad (4.13)$$

Os resultados obtidos utilizando-e a estrutura de modelo de referência (4.12) impõe ao sistema a necessidade de estoque inicial correspondente a quatro disparos de entradas. Este estoque inicial, em muitos casos, indica que o sistema não atende a questão de JIT nos disparos iniciais, indicando a necessidade de fornecer a matéria-prima somente quando o posto de trabalho necessitar. Somente a partir do quinto disparo das entradas de alimentação é que o estoque inicial vai desaparecer.

No método proposto os instantes de tempo de disparos das transições de entrada (4.13), liberando o material para o sistema produtivo, ocorre no exato momento em que os postos de trabalho necessitam, garantindo o JIT. O controlador do método proposto opera na mesma taxa de produção do sistema, da mesma forma o método por modelo de referência.

Comparando os valores dos instantes de tempo de disparos das transições de entrada (4.12) e (4.13), pode ser visto que  $u_F < u_{JIT}$ . Através deste exemplo numérico, a eficácia de o método proposto é demonstrada, assegurando o maior atraso de entradas de alimentação, sem afetar os instantes de tempo das saídas dos produtos.

As datas das saídas dos produtos nos dois métodos são os mesmas, isto é:

$$y_F(0) = 23, y_F(1) = 29, y_F(2) = 35, y_F(3) = 41, \dots$$

$$. y_{JIT}(0) = 23, y_{JIT}(1) = 29, y_{JIT}(2) = 35, y_{JIT}(3) = 41, \dots$$

Conclui-se que os objetivos foram alcançados na implementação do método proposto, sendo possível verificar que os resultados obtidos pelo método proposto para os valores dos tempos de atraso das entradas de alimentação são melhores do que os obtidos pela estratégia de controle por modelo de referência.

Na próxima seção, os método proposto é aplicado para sistemas MIMO (*multiple-input and multiple-output*) e os resultados são comparados quando utilizando as técnicas por modelo de re-

ferência aplicadas por Cottenceau et al. (2003) [CLHB03].

#### 4.2.4 Aplicação do Método I para Sistemas com Várias Entradas e Várias Saídas

Embora as condições iniciais estabelecidas para aplicação do método I estejam direcionadas para a estrutura do  $GET_{MB}$ , isso não impede que ele possa ser aplicado em outros sistemas, desde que esses sistemas não possuam fichas nos caminhos elementares entre entradas e saídas do sistema. Para ilustrar que o método I não é limitado a sistemas SISO (*Single Input, Single Output*), o método será aplicado em um sistema com várias entradas e várias saídas, e com capacidade de processamento das máquinas maior que uma peça por vez.

O sistema considerado para aplicação é o ilustrado na Figura 4.8 com quatro máquinas, duas entradas de alimentação e duas saídas. Os resultados da aplicação do método proposto nesta tese serão comparados com os resultados obtidos aplicando o método por modelo de referência. O resultado do controlador obtido na aplicação do método por modelo de referência é mostrado em linhas pontilhadas na mesma figura. Para essa aplicação o objetivo é mostrar que é possível, efetuando-se algumas adaptações em trabalhos futuros, aplicar o método I para uma classe maior de sistemas modelados por GET.

A matriz de transferência do sistema é:

$$H = \begin{bmatrix} 6(1\gamma)^* & 7(1\gamma)^* \\ \varepsilon & 20(15\gamma)^* \end{bmatrix}. \quad (4.14)$$

A matriz de controle que estabiliza o GET, sem diminuir a taxa de produção original, otimizando o número de fichas no sistema e garantindo o maior atraso inicial nas entradas do sistema, tanto quanto possível, é dada por:

$$F_c = \begin{bmatrix} 1\gamma \oplus 8\gamma^2 \oplus 21\gamma^3(15\gamma)^* & 1\gamma^2(15\gamma)^* \\ 0\gamma \oplus 7\gamma^2 \oplus 20\gamma^3(15\gamma)^* & 0\gamma^2(15\gamma)^* \end{bmatrix}. \quad (4.15)$$

Os resultados de  $H$  e  $F_c$  e todo o procedimento adotado, bem como todo o desenvolvimento das equações utilizando as técnicas por modelo de referência, podem ser encontrados em [CLHB03].

Agora, após os resultados da aplicação do método por modelo de referência serem conhecidos, os parâmetros do controlador serão calculados utilizando-se o método proposto nesta tese, método I, para que os resultados de ambos os métodos possam ser comparados.

O problema de otimização multiobjetivo, método I, demonstrado anteriormente, tem solução ótima se

$m = \left\lceil \frac{y_r(0)}{\beta} \right\rceil$ ,  $\beta \geq \lambda$  e  $[\varphi] = (C \otimes A_0^* \otimes B_0) \not\leq y_r(0)$ , sendo  $y_r(0) = y(0)$  quando  $u(0) = [0]$  e  $y(0)$  é definido pela Equação (4.3).

Para sistemas MIMO, o método é aplicado individualmente para cada entrada, em seguida, os valores dos tempos de saída são somados de acordo com a álgebra max-plus, ou seja, é tomado o maior valor, de forma que  $y_r(0) = y_1(0) \oplus y_2(0) \oplus y_3(0) \oplus \dots \oplus y_q(0)$ . Os valores dos tempos de







dados por:

$$\begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 7 \\ 20 \end{bmatrix}, \begin{bmatrix} 8 \\ 35 \end{bmatrix}, \begin{bmatrix} 9 \\ 50 \end{bmatrix}, \begin{bmatrix} 10 \\ 65 \end{bmatrix}, \begin{bmatrix} 11 \\ 80 \end{bmatrix}, \dots \quad (4.16)$$

Para efeitos de comparação, os resultados obtidos dos instantes de tempo de entradas das matérias-primas, obtidos pelas duas técnicas, são mostrados. As primeiras séries obtidas são:

$$u_F = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 8 \\ 7 \end{bmatrix}, \begin{bmatrix} 20 \\ 20 \end{bmatrix}, \begin{bmatrix} 35 \\ 35 \end{bmatrix}, \begin{bmatrix} 50 \\ 50 \end{bmatrix}, \dots \quad (4.17)$$

$$u_{JIT} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 16 \\ 15 \end{bmatrix}, \begin{bmatrix} 31 \\ 30 \end{bmatrix}, \begin{bmatrix} 46 \\ 45 \end{bmatrix}, \begin{bmatrix} 61 \\ 60 \end{bmatrix}, \dots \quad (4.18)$$

Neste exemplo, semelhantemente ao que ocorreu na análise do exemplo anterior, comparando-se os valores dos instantes de tempo de disparo das transições de entrada (4.17) e (4.18), pode ser visto que  $u_F < u_{JIT}$ , através desse exemplo numérico, a eficácia do método proposto é demonstrada, garantindo o maior atraso nas entradas de alimentação, sem a necessidade de formação de estoque.

$$\begin{bmatrix} y_{1_F}(0) \\ y_{2_F}(0) \end{bmatrix} = \begin{bmatrix} 7 \\ 20 \end{bmatrix}, \quad \text{e} \quad \begin{bmatrix} y_{1_{JIT}}(0) \\ y_{2_{JIT}}(0) \end{bmatrix} = \begin{bmatrix} 7 \\ 20 \end{bmatrix}.$$

A aplicação de ambos os métodos, no caso, o método I e o método por modelo de referência, afetam os instantes de saída de  $y_1$ , pois esta passa a operar na mesma taxa de produção de  $y_2$ . Após o controle, os dois métodos garantem que o sistema opere na mesma taxa de saída dos produtos, no entanto o resultado do método proposto apresenta melhor resultado para evitar a formação de estoque inicial nas entradas do sistema.

Outras aplicações do método I proposto nesta tese, para sistemas com múltiplas entradas e múltiplas saídas, podem ser encontradas em [DML13]. Um dos resultados é mostrado a seguir.

### Segundo Exemplo de Aplicação do Método I para Sistemas MIMO

Considere um sistema MIMO com estoques internos, estoques de entrada e com saídas com diferentes taxas de produção. Os postos de trabalho ou estoques onde não aparece o tempo de permanência são considerados como tendo valor igual a zero.

O sistema consiste de cinco entradas de alimentação ( $u_1, u_2, u_3, u_4, u_5$ ), vinte transições internas ( $x_1, x_2, x_3, \dots, x_{20}$ ), dez postos de trabalho ( $w_1, w_2, w_3, \dots, w_{10}$ ), onze estoques internos, e três saídas ( $y_1, y_2, y_3$ ) (Figura 4.10).

As equações recursivas:

$$x_1(k) = 2u_1(k) \oplus x_6(k-1);$$

$$x_2(k) = 1u_2(k) \oplus x_7(k-1);$$

$$x_3(k) = 4u_3(k) \oplus x_8(k-1);$$

$$x_4(k) = 1u_4(k) \oplus x_9(k-1);$$

$$x_5(k) = 2u_5(k) \oplus x_{10}(k-1);$$

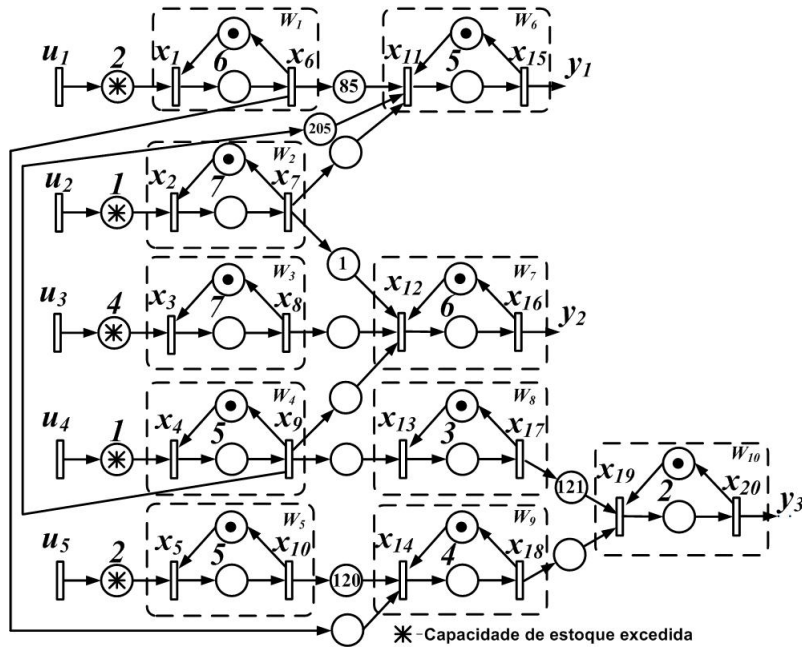


Figura 4.10: Sistema de manufatura MIMO.

$$\begin{aligned}
 x_6(k) &= 6x_1(k); \\
 x_7(k) &= 7x_2(k); \\
 x_8(k) &= 7x_3(k); \\
 x_9(k) &= 5x_4(k); \\
 x_{10}(k) &= 5x_5(k); \\
 x_{11}(k) &= x_6(k) \oplus x_9(k) \oplus x_7(k) \oplus x_{15}(k-1); \\
 x_{12}(k) &= x_7(k) \oplus x_8(k) \oplus x_9(k) \oplus x_{16}(k-1); \\
 x_{13}(k) &= x_9(k) \oplus x_{17}(k-1); \\
 x_{14}(k) &= x_{10}(k) \oplus x_6(k) \oplus x_{18}(k-1); \\
 x_{15}(k) &= 5x_{11}(k); \\
 x_{16}(k) &= 6x_{12}(k); \\
 x_{17}(k) &= 3x_{13}(k); \\
 x_{18}(k) &= 4x_{14}(k); \\
 x_{19}(k) &= x_{17}(k) \oplus x_{18}(k) \oplus x_{20}(k-1); \\
 x_{20}(k) &= 2x_{19}(k).
 \end{aligned}$$

A modelagem matricial:



Os tempos das saídas dos produtos, com maior taxa de produção dada por  $\beta = 7$ , são :

Tabela 4.1: Tempos de Saída sem o Controlador

Saída	Tempo(seg)									
$y_1$	13	19	25	31	37	43	49	55	61	67
$y_2$	17	24	31	38	45	52	59	66	73	80
$y_3$	14	20	26	32	38	44	50	56	62	68

Aplicando o método I, os seguintes resultados são obtidos (Figura 4.11):

$$[\varphi] = [\varphi]_1 \oplus [\varphi]_2 \oplus [\varphi]_3 = (C_1 \otimes A_0^* \otimes B_0) \phi y_r(0) \oplus (C_2 \otimes A_0^* \otimes B_0) \phi y_r(0) \oplus (C_3 \otimes A_0^* \otimes B_0) \phi y_r(0),$$

$$[\varphi] = [0 \ 0 \ 0 \ 2 \ 1],$$

$$m = \left\lceil \frac{y_r(0)}{\beta} \right\rceil = \left\lceil \frac{17}{7} \right\rceil = 3.$$

Após a implementação do controle, as primeiras saídas ocorrem em:  $y_1(0) = 13$ ,  $y_2(0) = 17$  e  $y_3(0) = 14$  com a taxa  $\beta = 7$ .

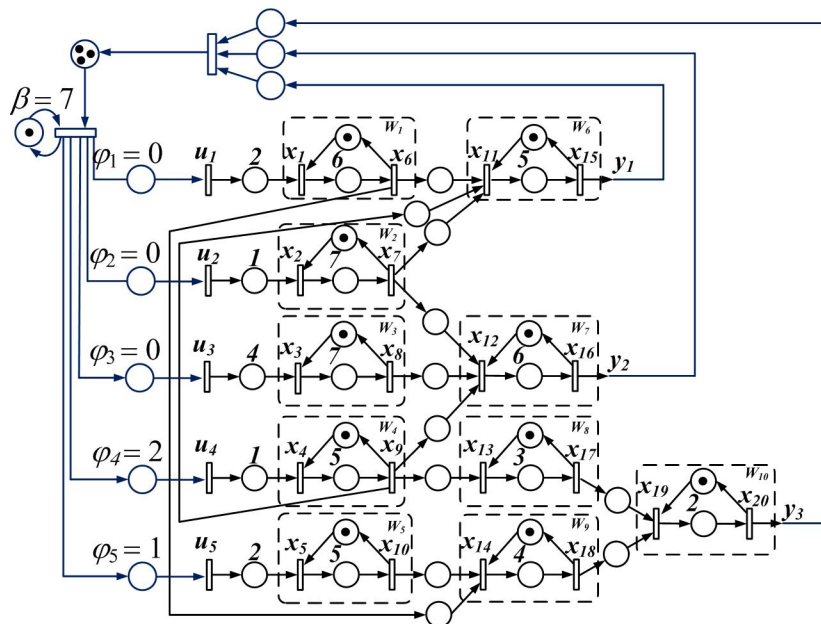


Figura 4.11: Controle após a implementação do método proposto.

#### 4.2.5 Considerações sobre os Resultados do Método I

Um método para controlar o fluxo de entrada de matéria-prima foi desenvolvido com base em dígrafos de datadores. Todas as provas das proposições foram desenvolvidas matematicamente na álgebra max-plus. Foi mostrado que não foi necessária a implementação de algoritmos complexos para encontrar os parâmetros do controlador proposto, diferentemente do que é feito na série de potências formal, utilizando matrizes de transferência.

O método foi aplicado em dois sistemas modelados por GET para demonstrar que o mesmo não é limitado, mas que pode ser utilizado numa grande variedade de sistemas que tratam com problemas de sincronização e atraso de tempo. Em ambos os exemplos, os tempos de atrasos nas entradas, obtidos pelo método proposto, foram melhores que os obtidos pelo método por modelo de referência. Isto demonstra que o método proposto é eficaz, assegurando o máximo atraso possível nas entradas de alimentação, sem comprometer a primeira saída dos produtos.

Foi demonstrado que, para sistemas MIMO, quando as saídas têm diferentes taxas de produção, os métodos garantem a estabilidade do sistema, mas limitam todas as saídas para uma única taxa de produção.

Como dito anteriormente, a principal contribuição do método I é a simplicidade de manipulação de matriz para o cálculo dos parâmetros do controlador, porque com o método proposto, não será necessário alterar as equações por datadores para série de potências formal, o que reduz consideravelmente o tempo de modelagem do sistema, além de não serem necessárias aplicações de algoritmos complexos. Observa-se, então, que é mais vantajoso trabalhar com datadores do que com matrizes de transferência, pela simplicidade na implementação do sistema de controle.

### 4.3 Método de Controle II - Sincronização do Disparo de Transições

Na implementação do método de sincronização do disparo de várias transições, as hipóteses, condições iniciais, estruturação do modelo, numeração da transições do modelo e as condições iniciais canônicas utilizadas no desenvolvimento deste método serão as mesmas consideradas na Seção 4.1, onde também foi estabelecido que a capacidade de processamento de cada máquina do sistema é de uma peça por vez.

Para facilitar o desenvolvimento matemático das equações, os vetores das matrizes utilizados nas equações serão descritos de forma simplificada para que os resultados finais não sejam tão extensos, de acordo com a notação a seguir.

#### 4.3.1 Notação Vetorial

Os vetores linha da matriz  $A$  serão representados da seguinte forma:

$$A = \begin{bmatrix} [A]_1 \\ [A]_2 \\ [A]_3 \\ \vdots \\ [A]_n \end{bmatrix} = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ a_{31} & a_{32} & \dots & a_{3n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{pmatrix},$$

sendo  $[A]_i = (a_{i1} \ a_{i2} \ a_{i3} \ \dots \ a_{in}) \in A^{n \times n}$ , para  $i = 1, 2, 3, \dots, n$ , são os  $n$  vetores linha da matriz  $A$ ,  $n$  também representa o número de transições internas do  $GET_{MB}$ .

Os vetores linha da matriz  $B$  serão representados da seguinte forma:

$$B = \begin{bmatrix} [B]_1 \\ [B]_2 \\ [B]_3 \\ \vdots \\ [B]_n \end{bmatrix} = \begin{bmatrix} b_{11} & b_{12} & \dots & b_{1l} \\ b_{21} & b_{22} & \dots & b_{2l} \\ b_{31} & b_{32} & \dots & b_{3l} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nl} \end{bmatrix},$$

sendo  $[B]_i = (b_{i1} \ b_{i2} \ b_{i3} \ \dots \ b_{il}) \in B^{n \times l}$ , para  $i = 1, 2, 3, \dots, n$ , são os  $n$  vetores linha da matriz  $B$ , e  $l$  é o número de entradas do  $GET_{MB}$ .

Então,  $x(k)$  pode ser descrito como:

$x(k)$  pode ser escrito como

$$x(k) = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \\ \vdots \\ x_n(k) \end{bmatrix} = A \otimes \begin{bmatrix} x_1(k-1) \\ x_2(k-1) \\ x_3(k-1) \\ \vdots \\ x_n(k-1) \end{bmatrix} \oplus B \otimes u(k),$$



Então, individualmente,  $x_1(k)$ ,  $x_2(k)$ ,  $\dots$ ,  $x_n(k)$  também podem ser escritos como

$$x_1(k) = [0 \ \varepsilon \ \dots \ \varepsilon] \otimes x(k) = [0 \ \varepsilon \ \dots \ \varepsilon] \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} = [0 \ \varepsilon \ \dots \ \varepsilon] \otimes (A \otimes x(k-1) \oplus B \otimes u(k)),$$

$$x_2(k) = [\varepsilon \ 0 \ \dots \ \varepsilon] \otimes x(k) = [\varepsilon \ 0 \ \dots \ \varepsilon] \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} = [\varepsilon \ 0 \ \dots \ \varepsilon] \otimes (A \otimes x(k-1) \oplus B \otimes u(k)),$$

$$x_n(k) = [\varepsilon \ \varepsilon \ \dots \ 0] \otimes x(k) = [\varepsilon \ \varepsilon \ \dots \ 0] \otimes \begin{bmatrix} x_1(k) \\ x_2(k) \\ \vdots \\ x_n(k) \end{bmatrix} = [\varepsilon \ \varepsilon \ \dots \ 0] \otimes (A \otimes x(k-1) \oplus B \otimes u(k)).$$

Assim, de forma simplificada, os elementos de  $x(k)$  podem ser representados por:

$$\begin{aligned} x_1(k) &= [A]_1 \otimes x(k-1) \oplus [B]_1 \otimes u(k), \\ x_2(k) &= [A]_2 \otimes x(k-1) \oplus [B]_2 \otimes u(k), \\ &\vdots \\ x_n(k) &= [A]_n \otimes x(k-1) \oplus [B]_n \otimes u(k), \end{aligned}$$

### 4.3.2 Método de Controle II: Descrição do Problema

O estudo sobre sincronização dos disparos de várias transições de um sistema modelado por GET, em um contexto *just-in-time*, como abordado nesta tese, ainda não apresenta resultados na literatura. Os trabalhos que possuem uma relação mais aproximada com a questão de sincronização são os trabalhos de David-Henriet et al. [DHHRC14] e [DHHRC13], nos quais, um sistema max-plus com sincronização parcial é um sistema a evento discreto persistente dividido em um subsistema principal e um subsistema secundário, de modo que o subsistema secundário tem que ajustar o seu comportamento ao subsistema principal, mas o subsistema principal não é afetado pelo subsistema secundário. Nessa sincronização parcial, os resultados avaliados são os desempenhos dos sistemas, avaliando os dados das entradas e saídas dos sistemas, diferente da proposta de sincronização que será abordada nesta tese.

A sincronização de atividades com a finalidade de eliminar os tempos de espera é de grande importância, pois possibilita o controle do início e fim de várias atividades críticas em um processo produtivo. Este tipo de controle é adequado na aplicação de processos especiais, tais como: resfriamento de peças, teste de aquecimento, secagem de material, acoplamento e outros, de forma que os instantes de tempo de término de duas ou mais atividades e o início de uma próxima, que depende

do fim das anteriores, pode significar a redução de perdas de material, caso todas as condições de sincronização sejam atendidas.

Um método de controle será proposto para a sincronização dessas várias transições. Vários exemplos são apresentados, ilustrando a sincronização de duas, três e quatro transições internas de um sistema produtivo. Para um melhor entendimento do que significa a sincronização e a redução de perdas de matéria-prima, considere o seguinte exemplo.

**Exemplo 4.3.1.** Uma parte de um sistema produtivo é formado por três máquinas  $M_1$ ,  $M_2$  e  $M_3$ , conforme ilustrado pela Figura 4.12. As máquinas  $M_1$  e  $M_2$  produzem partes (peças) de um produto que é concluído pela máquina  $M_3$ , essa conclusão é feita acoplando-se as peças produzidas por  $M_1$  e  $M_2$ . A máquina  $M_1$ , após produzir a peça número 1, também injeta uma cola de secagem rápida nessa peça. A máquina  $M_2$  produz a peça número 2, que será acoplada à peça número 1 na máquina  $M_3$ . A cola injetada na peça 1 tem um tempo de cura muito rápido, e muitas dessas peças podem ser perdidas, porque o processo pode não entregar as peças em um tempo mínimo para evitar a secagem da cola. O processo precisa ser ajustado para garantir que não haverá atraso no instante em que as peças são disponibilizadas para a máquina  $M_3$ . Para atender a essa condição, sem comprometer os estoques de entrada, é necessário implementar um controlador para atrasar os tempos de entrada de matéria-prima no sistema e garantir que a taxa de produção atenda a restrição imposta ao sistema, neste caso, garantir que as peças produzidas por  $M_1$  e  $M_2$  sejam disponibilizadas no mesmo instante de tempo para que a máquina  $M_3$  complete o processo produtivo.

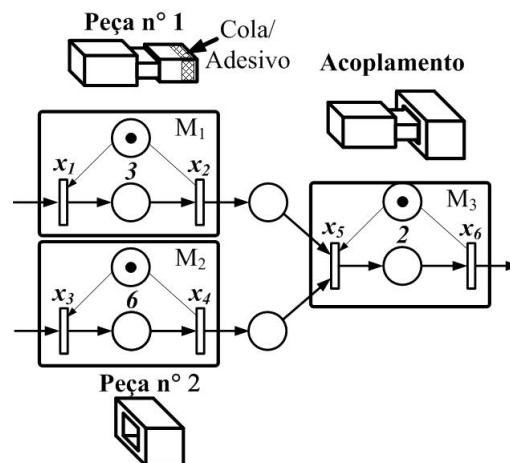


Figura 4.12: Sistema de Manufatura ilustrando uma simples sincronização.

Para realizar a sincronização, um controlador, baseado nos conceitos de estabilidade de GET [Com98], é proposto. A estrutura do controlador é ilustrada na Figura 4.13, sendo que  $[\varphi]$  é o vetor com os tempos de atraso das entradas de alimentação do sistema,  $\beta$  é a taxa de disparo do controlador, e  $m$  é o número de fichas do controlador.

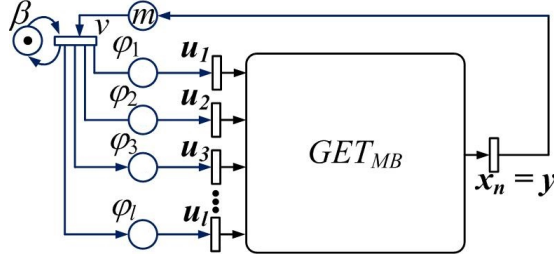


Figura 4.13: Sistema de realimentação para sincronização de transições.

O objetivo do controlador é sincronizar os disparos das transições selecionadas. Assim, o problema de controle pode ser formulado como um problema multiobjetivo. No qual, dado um sistema modelado por:

$$x(k) = A \otimes x(k-1) \oplus B \otimes u(k), \quad (4.19)$$

$$y(k) = C \otimes x(k). \quad (4.20)$$

A aplicação do controlador resultará num sistema que passa a ser representado pelas seguintes equações:

$$\underline{x}(k) = A \otimes x(k-1) \oplus B \otimes \underline{u}(k), \quad (4.21)$$

$$\underline{y}(k) = C \otimes \underline{x}(k), \quad (4.22)$$

$$\underline{u}(k) = [\varphi] \otimes v(k), \quad (4.23)$$

$$v(k) = \beta \otimes v(k-1) \oplus y(k-m), \quad (4.24)$$

sendo  $\underline{x}(k)$  os instantes de disparos das transições no sistema controlado,  $\underline{u}(k)$  os instantes de tempo dos disparos das entradas de alimentação no sistema controlado, e  $\underline{y}(k)$  os instantes de tempo das saídas dos produtos nos sistema controlado.

Tendo por finalidade assegurar o sincronismo de  $l$  transições, satisfazendo a condição de restrição expressa por:

$$\Delta_{i_1} \otimes x_{i_1}(0) = \Delta_{i_2} \otimes x_{i_2} = \dots = \Delta_{i_j} \otimes x_{i_j}(0) \quad j \in \{1, 2, 3, \dots, l\}. \quad (4.25)$$

sendo  $x_i = \{x_{i_1}, x_{i_2}, \dots, x_{i_j}\} \in \{x_1, x_2, x_3, \dots, x_n\}$  o conjunto das transições selecionadas para a sincronização,  $\Delta_i$  é o conjunto dos tempos de sincronização entre as transições, para dispararem adiantadas ou não umas em relação as outras. O valor e o sinal da relação  $Sinc = \Delta_{i_1} \setminus \Delta_{i_2}$  definem a forma de sincronização e o intervalo de tempo de disparo entre as transições  $x_{i_1}$  e  $x_{i_2}$ . Caso  $Sinc$  seja positivo, a transição  $x_{i_1}$  dispara adiantada em relação a  $x_{i_2}$  com a diferença de  $|Sinc|$  unidades de tempo. Caso  $Sinc$  seja negativo, a transição  $x_{i_1}$  dispara atrasada em relação a  $x_{i_2}$  com a diferença de  $|Sinc|$  unidades de tempo. Caso  $Sinc$  seja zero, as transições  $x_{i_1}$  e  $x_{i_2}$  disparam nos mesmos instantes de tempo.

Dessa forma, o problema de otimização multiobjetivo apresenta solução para a sincronização de várias transições, em um sistema modelado por *GET*, quando a hipótese e as proposições a seguir são adotadas.

*Hipótese 4.3.1.* Para realizar o controle de disparos sincronizados de transições em um  $GET_{MB}$  é necessário que as transições façam parte de caminhos concorrentes, ou seja, as transições devem pertencer a caminhos elementares com entradas diferentes. Se uma transição depende do disparo de uma transição predecessora ou sucessora, isso significa que essas transições pertencem ao mesmo caminho elementar de um sistema, então não é possível que elas disparem no mesmo instante de tempo e com a mesma quantidade de disparos. Podem ter a mesma quantidade de disparos, mas não no mesmo instante de tempo, e podem disparar no mesmo instante de tempo, mas não com a mesma quantidade de disparos.

**Exemplo 4.3.2.** Dado um  $GET_{MB}$  (Figura 4.14).

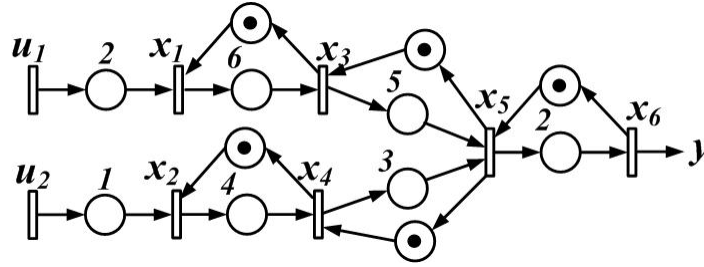


Figura 4.14: Sistema de Manufatura ilustrando uma sincronização de duas transições.

As equações de disparos das transições  $x_1$  e  $x_3$  são:

$$x_1(k) = x_3(k-1) \oplus u_1(k) \otimes 2, \quad (4.26)$$

$$x_3(k) = x_1(k) \otimes 6 \oplus x_5(k-1). \quad (4.27)$$

Observa-se pela figura e nas equações que ambas transições dependem uma da outra para que os disparos sucessivos possam ocorrer.  $x_1$  não dispara sem que  $x_3$  já tenha efetuado um disparo como condição inicial no tempo  $-\infty$ , e  $x_3$  não poderá disparar sem que primeiro ocorra o disparo de  $x_1$ . A única condição de disparo entre as transições é a própria equação de modelagem do sistema, não é possível fazer, por exemplo, que os instantes de disparos de  $x_1$  e  $x_3$  sejam  $x_1(0) = x_3(0)$ , ou seja, não existe outra forma de sincronização. Da mesma forma ocorre entre  $x_2$  e  $x_4$ ,  $x_3$  e  $x_6$ , e outras transições que pertencem aos mesmos caminhos elementares.

As transições possuem tempos mínimos para realizar o primeiro disparo, bem como tempos mínimos para uma nova habilitação, ou seja, possuem restrições de tempo para disparos sucessivos. Para que possam disparar novamente, as condições de restrições devem ser satisfeitas. Essas condições são, muitas das vezes, os disparos de outras transições. Esse tempo de espera para que a transição fique novamente habilitada pode ser muito maior que a taxa de produção do sistema,

então é necessário que seja calculado uma nova taxa de produção para o sistema, com o objetivo de evitar a instabilidade do processo. Para atender a algumas situações de restrições de controle, o sistema ficará mais lento, e caso o mesmo continue a operar na mesma taxa natural de produção, o resultado será um *overflow* nos estoques de entrada.

Para que o controlador assegure o disparo sincronizado no primeiro disparo é necessário que a proposição a seguir seja implementada.

**Proposição 4.3.1.** *Dado um  $GET_{MB}$ , um conjunto de transições para sincronização,  $x_i = \{x_{i_1}, x_{i_2}, \dots, x_{i_j}\} \in \{x_1, x_2, \dots, x_n\}$ , um conjunto de tempos de sincronização,  $\Delta_i \in \mathbb{Z}$ ,  $i \in \{1, 2, \dots, n\}$ . Os atrasos,  $[\varphi]$ , nas entradas de alimentação que garantem a sincronização de  $l$  transições, satisfazendo a condição de restrição dada por  $\Delta_{i_1} \otimes x_{i_1}(k) = \Delta_{i_2} \otimes x_{i_2}(k) = \dots = \Delta_{i_j} \otimes x_{i_j}(k)$ , sendo  $j = 1, 2, \dots, l$ , são determinados por:*

$$[\varphi] = \frac{(\Delta_{i_1} \otimes [B]_{i_1}) \oplus (\Delta_{i_2} \otimes [B]_{i_2}) \oplus \dots \oplus (\Delta_{i_l} \otimes [B]_{i_l})}{(\Delta_{i_1} \otimes [B]_{i_1} \otimes [0]) \oplus (\Delta_{i_2} \otimes [B]_{i_2} \otimes [0]) \oplus \dots \oplus (\Delta_{i_l} \otimes [B]_{i_l} \otimes [0])}. \quad (4.28)$$

ou representados matricialmente por:

$$[\varphi] = \begin{bmatrix} \Delta_{i_1} & \Delta_{i_2} & \dots & \Delta_{i_r} \end{bmatrix} \otimes \begin{bmatrix} [B]_{i_1} \\ [B]_{i_2} \\ \vdots \\ [B]_{i_r} \end{bmatrix} \otimes \begin{bmatrix} \Delta_{i_1} & \Delta_{i_2} & \dots & \Delta_{i_r} \end{bmatrix} \otimes \begin{bmatrix} [B]_{i_1} \\ [B]_{i_2} \\ \vdots \\ [B]_{i_r} \end{bmatrix} \otimes [0]. \quad (4.29)$$

sendo  $[0]$  um vetor de zeros de dimensões apropriadas.

*Demonstração.* Para o primeiro disparo, em  $k = 0$ , todas as máquinas do processo produtivo estão ociosas, prontas para receber matéria-prima. Assim, para realizar a sincronização dos instantes de tempos de disparos das transições, é necessário que os somatórios dos tempos de atraso entre cada entrada de alimentação até as transições selecionadas possuam o mesmo valor, já adicionados dos tempos de sincronização  $\Delta$ . Considerando que toda a matéria-prima está disponível nas entradas de alimentação no primeiro disparo, os tempo de disparos das entradas de alimentação para o cálculo dos tempo de atraso é dado por um vetor de zeros, ou seja,  $u(0) = [0]$ . Assim, o maior tempo de atraso para uma determinada transições  $r$  é dado por  $x_r(0) = B_r \otimes [0]$ . No sistema realimentado, após a introdução do controlador, o maior tempo de atraso para o primeiro disparo para a mesma transição é dado  $\underline{x}_r(0) = B_r \otimes [\varphi]$ . A condição de sincronização é que o maior dos tempos de atraso das transições selecionadas para sincronização no sistema sem realimentação seja igual a todos os tempos de atraso das transições no sistema realimentado, quando somados aos tempos do vetor de atrasos  $[\varphi]$ . O exemplo a seguir ilustra a metodologia empregada para sincronização no primeiro disparo.

**Exemplo 4.3.3.** Considere o mesmo  $GET_{MB}$  ilustrado pela Figura 4.14. O objetivo é sincronizar o disparo das transições  $x_3$  e  $x_4$ , de forma que as mesmas disparem nos mesmos instantes de tempo, ou seja,  $x_3(k) = x_4(k)$  e  $\Delta_3 = \Delta_4 = 0$

A modelagem matricial do sistema é dada por:

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & 3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon \end{bmatrix} \quad A_1 = \begin{bmatrix} 0 & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 0 & \varepsilon & 0 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 0 & 0 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & 0 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix} \quad B_0 = \begin{bmatrix} 2 & \varepsilon \\ \varepsilon & 1 \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}$$

$$y(k) = C \otimes x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} x(k)$$

Considerando-se que toda a matéria-prima está disponível nas entradas de produção no instante de tempo  $t = 0$ , o primeiro disparo das transições de entrada ocorrem no tempo zero, ou seja,  $u(0) = [0]$ , dessa forma, utilizando-se a forma vetorial da Equação (??) ou por inspeção, pode-se determinar que  $x_3(0) = 8$  e  $x_4(0) = 5$ .

Utilizando o controle de fluxo de entrada de matéria-prima, não é possível fazer com que o tempo de disparo de  $x_3$  seja menor que 8 unidades de tempo, esse é o mínimo tempo de disparo dessa transição. Então não é possível que esse tempo seja adiantado, pois essa é uma restrição da condição de modelagem do sistema. Dessa forma, a única possibilidade de fazer-se que  $x_3(0)$  seja igual a  $x_4(0)$  é aplicando atrasos nas entradas de alimentação do sistema, para o caso específico, em  $u_2$  (Figura 4.15), retardando-se o tempo de disparo de  $x_4$ . Assim, o menor tempo de disparo sincronizado,  $\underline{x}_3(0)$  e  $\underline{x}_4(0)$ , das transições no sistema controlado ocorre quando  $t = 8$ , dessa forma, por inspeção, é possível verificar-se que é necessário atrasar o instante de tempo de disparo da transição  $x_4$  em 3 unidades de tempo, implementado-se atrasos nas entradas de alimentação dados por:  $\varphi_1 = 0$  e  $\varphi_2 = 3$ .

A aplicação dos atrasos possibilitou a sincronização dos instantes de tempo de disparos das

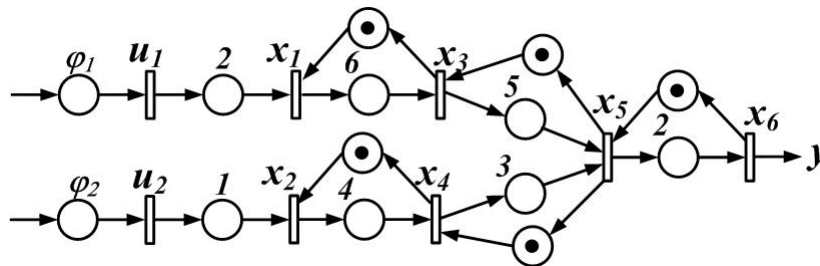


Figura 4.15: Atrasos aplicados no sistema para realizar a sincronização.

transições em  $k = 0$ .

Todos os comprimentos máximos dos caminhos de cada entrada de alimentação até as transições a serem controladas, somadas com os atrasos, devem possuir o mesmo comprimento. Esse comprimento pode ser chamado de tempo mínimo de sincronização para  $k = 0$ , ou caminho crítico, sendo denotado por  $\mathcal{C}_\Delta$ .  $\mathcal{C}_\Delta$  é o maior comprimento dos caminhos desde as entradas de alimentação até

as transições  $x_3$  e  $x_4$ . Os valores do vetor  $[\varphi]$  são os atrasos necessários para que os comprimentos dos máximos caminhos desde as entradas até as transições  $x_3$  e  $x_4$  possuam o mesmo tamanho. Essa é a condição para a solução deste controle para  $k = 0$ . As fichas devem chegar no mesmo instante de tempo nas transições a serem controladas, então elas devem percorrer caminhos de mesmo comprimento.

No caso geral, tem-se que o caminho crítico para as transições a serem sincronizadas é dado por:

$$\mathcal{C}_\Delta = \bigoplus_{j=1}^l \Delta_{i_j} \otimes [B]_{i_j} \otimes [0], \quad (4.30)$$

sendo que  $[0]$  denota um vetor de zeros com dimensão compatível.

A estrutura de controle proposta, assegura que em malha fechada  $\underline{u}(0) = [\varphi] \otimes v(0)$ , sendo  $v(0) = 0$ . Assim, para se assegurar o atraso de sincronização,  $[\varphi]$  é projetado de forma que todos os caminhos até as transições sejam compatíveis com os atrasos  $\Delta_i$ . Dessa forma:

$$\begin{bmatrix} \Delta_{i_1} \otimes \underline{x}_{i_1}(0) \\ \Delta_{i_2} \otimes \underline{x}_{i_2}(0) \\ \vdots \\ \Delta_{i_l} \otimes \underline{x}_{i_l}(0) \end{bmatrix} = \begin{bmatrix} \mathcal{C}_\Delta \\ \mathcal{C}_\Delta \\ \vdots \\ \mathcal{C}_\Delta \end{bmatrix} \quad (4.31)$$

Realizando-se a substituição dos vetores de  $\underline{x}_i$ , em  $k = 0$ , tem-se que:

$$\begin{bmatrix} \Delta_{i_1} \otimes B_{i_1} \otimes [\varphi] \\ \Delta_{i_2} \otimes B_{i_2} \otimes [\varphi] \\ \vdots \\ \Delta_{i_l} \otimes B_{i_l} \otimes [\varphi] \end{bmatrix} = \begin{bmatrix} \mathcal{C}_\Delta \\ \mathcal{C}_\Delta \\ \vdots \\ \mathcal{C}_\Delta \end{bmatrix} \quad (4.32)$$

Simplificando a expressão:

$$\begin{bmatrix} \Delta_{i_1} \otimes B_{i_1} \\ \Delta_{i_2} \otimes B_{i_2} \\ \vdots \\ \Delta_{i_l} \otimes B_{i_l} \end{bmatrix} \otimes [\varphi] = \begin{bmatrix} \mathcal{C}_\Delta \\ \mathcal{C}_\Delta \\ \vdots \\ \mathcal{C}_\Delta \end{bmatrix} \quad (4.33)$$

sendo que:  $\mathcal{C}_\Delta = \Delta_{i_1} \otimes B_{i_1} \otimes [0] \oplus \Delta_{i_2} \otimes B_{i_2} \otimes [0] \oplus \dots \oplus \Delta_{i_l} \otimes B_{i_l} \otimes [0]$ . Desenvolvendo-se a solução da forma matricial, tem-se que:

$$[\varphi] = \frac{(\Delta_{i_1} \otimes [B]_{i_1}) \oplus (\Delta_{i_2} \otimes [B]_{i_2}) \oplus \dots \oplus (\Delta_{i_l} \otimes [B]_{i_l})}{(\Delta_{i_1} \otimes [B]_{i_1} \otimes [0]) \oplus (\Delta_{i_2} \otimes [B]_{i_2} \otimes [0]) \oplus \dots \oplus (\Delta_{i_l} \otimes [B]_{i_l} \otimes [0])}. \quad (4.34)$$

□

*Observação 4.3.1.* Caso alguns dos tempos de atraso das entradas, elementos do vetor  $[\varphi]$ , resulte em um valor que tende para o infinito, isto significa que essas entradas não contribuem para o

disparo das transições selecionadas em  $k = 0$ . Entretanto, os disparos para  $k > 0$  dependem dessas entradas, dessa forma, os valores dos elementos que tendem para infinito devem ser substituídos pelo valor zero.

Garantida a sincronização para o primeiro disparo, é necessário agora garantir a sincronização para a sequência de disparos futuros. Quando a sincronização é implementada no sistema, a taxa de alimentação das entradas de produção é determinada pela taxa de disparo do controlador,  $\beta$ . Contudo, é necessário que essa taxa não seja menor que o maior tempo necessário para que as transições selecionadas possam estar novamente habilitadas para um novo disparo. Essa taxa é o tempo mínimo necessário para que as transições disparem sempre nos mesmos instantes de sincronização, em cada ciclo do processo produtivo. Essa taxa deve também garantir a estabilidade do sistema, ou seja:

$$\beta \succeq \lambda, \quad (4.35)$$

$$\beta = \beta \oplus \lambda. \quad (4.36)$$

Assim, para que o controlador garanta a sincronização nas sequências de disparos das transições selecionadas é necessário que a taxa do controlador atenda a seguinte proposição.

**Proposição 4.3.2.** *Dado um  $GET_{MB}$ , um conjunto de transições para sincronização,  $x_i = \{x_{i_1}, x_{i_2}, \dots, x_{i_j}\} \in \{x_1, x_2, \dots, x_n\}$ , um conjunto de tempos de sincronização,  $\Delta_i \in \mathbb{Z}$ ,  $i \in \{1, 2, \dots, n\}$ . A taxa de disparo do controlador,  $\beta$ , que garante a sincronização das  $l$  transições é dado por:*

$$\beta = \text{traço} \left[ \frac{(\Delta_{i_1} \otimes [AB]_{i_1}) \oplus (\Delta_{i_2} \otimes [AB]_{i_2}) \oplus (\Delta_{i_3} \otimes [AB]_{i_3}) \oplus \dots \oplus (\Delta_{i_j} \otimes [AB]_{i_r})}{(\Delta_{i_1} \otimes [B]_{i_1}) \oplus (\Delta_{i_2} \otimes [B]_{i_2}) \oplus (\Delta_{i_3} \otimes [B]_{i_3}) \oplus \dots \oplus (\Delta_{i_r} \otimes [B]_{i_j})} \right], \quad (4.37)$$

sendo

$$\beta \succeq \lambda, \quad (4.38)$$

e

$$\lambda = \bigoplus_{j=1}^n (\text{traço}(A^{\otimes j}))^{\otimes 1/j}. \quad (4.39)$$

*Demonstração.* Em um  $GET_{MB}$  a taxa natural de produção do sistema modelado é dada pelo maior tempo de atraso de um posto de trabalho/máquina, definida por  $\lambda$ , que é calculada pela Equação  $\lambda = \bigoplus_{j=1}^n (\text{traço}(A^{\otimes j}))^{\otimes 1/j}$  [BCOQ92].

O valor de  $\beta$  é mínimo tempo necessário para que todas as transições selecionadas estejam novamente habilitadas para um novo disparo. O primeiro disparo das transições,  $x(0)$ , no sistema controlado ocorreu em  $k = 0$ , sendo definidos por:

$$\underline{x}(0) = A \otimes \underline{x}(-1) \oplus B \otimes [\varphi] = B \otimes [\varphi]. \quad (4.40)$$

Baseado no Teorema 2.5.3, os elementos da matriz  $B$  fornecem os comprimentos dos caminhos elementares entre cada transição de entrada,  $u_1, u_2, \dots, u_l$ , do sistema em relação a cada transição



interna do sistema,  $x_1, x_2, \dots, x_n$ . Dessa forma, o elemento  $b_{rl}$  fornece o intervalo de tempo de disparo entre a transição  $u_l$  e a transição  $x_r$ .

O segundo disparo das transições,  $x(1)$ , é definido por:

$$\underline{x}(1) = A \otimes \underline{x}(0) \oplus B \otimes u(1) = A \otimes [B] \otimes [\varphi] \oplus B \otimes \beta \otimes [\varphi]. \quad (4.41)$$

Em circuito fechado, com o controlador, o fluxo de entrada do sistema é levado a uma taxa fixa  $\beta$ , assim:

$$\underline{x}(k) = \underline{x}(0) \otimes \underbrace{\beta \otimes \beta \otimes \dots \otimes \beta}_k. \quad (4.42)$$

De onde poder ser deduzido que:

$$\underline{x}(1) = \beta \otimes \underline{x}(0). \quad (4.43)$$

Desta forma, a Equação 4.41 pode ser escrita como:

$$\beta \otimes \underline{x}(0) = A \otimes B \otimes [\varphi] \oplus B \otimes \beta \otimes [\varphi]. \quad (4.44)$$

O que resulta em:

$$\beta \otimes B \otimes [\varphi] = A \otimes B \otimes [\varphi] \oplus \beta \otimes B \otimes [\varphi], \quad (4.45)$$

$$\beta \otimes B \otimes [\varphi] \succeq A \otimes B \otimes [\varphi]. \quad (4.46)$$

Esse valor de  $\beta$  é o mínimo tempo necessário para que todas as transições selecionadas estejam novamente habilitadas. Qualquer valor de taxa acima desse tempo assegura a sincronização. Então:

$$\beta_{\min} = \min\{\beta \mid \underline{x}(1) \preceq \beta \otimes \underline{x}(0)\}. \quad (4.47)$$

Considerando-se novamente o Teorema 2.5.3, Os elementos da matriz do produto matricial  $[A \otimes B]$  fornece o comprimento do maior caminho entre cada uma das entradas do sistema e as transições internas para o segundo disparo. Esses caminhos envolvem os laços das transições. A primeira coluna da matriz  $[A \otimes B]$  fornece os valores dos maiores caminhos entre a entrada  $u_1$  é cada uma das transições internas do sistema ( $x_1, x_2, \dots, x_n$ ). A segunda coluna da matriz  $A \otimes B$  fornece os valores dos maiores caminhos entre a entrada  $u_2$  é cada uma das transições internas do sistema ( $x_1, x_2, \dots, x_n$ ), e assim sucessivamente. Então, o objetivo é encontrar a solução mínima para:

$$\beta \otimes B \succeq A \otimes B. \quad (4.48)$$

Considerando-se somente as transições selecionadas para sincronização e acrescentando-se os tempos

de sincronização  $\Delta_i$  para cada transição, tem-se que:

$$\beta \otimes \begin{bmatrix} \Delta_{i_1} \otimes B_{i_1} \\ \Delta_{i_2} \otimes B_{i_2} \\ \vdots \\ \Delta_{i_l} \otimes B_{i_l} \end{bmatrix} \succeq \begin{bmatrix} \Delta_{i_1} \otimes [A \otimes B]_{i_1} \\ \Delta_{i_2} \otimes [A \otimes B]_{i_2} \\ \vdots \\ \Delta_{i_l} \otimes [A \otimes B]_{i_l} \end{bmatrix} \quad (4.49)$$

Aplicando-se a operação de residuação, para solução do sistema, obtém-se:

$$\beta = \begin{bmatrix} \Delta_{i_1} \otimes B_{i_1} \\ \Delta_{i_2} \otimes B_{i_2} \\ \vdots \\ \Delta_{i_l} \otimes B_{i_l} \end{bmatrix} \oslash \begin{bmatrix} \Delta_{i_1} \otimes [A \otimes B]_{i_1} \\ \Delta_{i_2} \otimes [A \otimes B]_{i_2} \\ \vdots \\ \Delta_{i_l} \otimes [A \otimes B]_{i_l} \end{bmatrix} \quad (4.50)$$

O resultado dessa operação de residuação é uma matriz com todos os valores máximos dos intervalos de tempo de cada disparo das transição selecionada em relação aos disparos de cada entrada de alimentação entre o primeiro e o segundo disparo. O maior desses valores é o máximo tempo de habilitação que as transições precisam para estarem novamente habilitadas. Dessa forma, o mínimo de  $\beta$  que satisfaz a Equação (4.47) e garante o disparo sincronizado nas sequências sucessivas de disparos do sistema é obtido por:

$$\beta = \text{traço} \left[ \frac{(\Delta_{i_1} \otimes [AB]_{i_1}) \oplus (\Delta_{i_2} \otimes [AB]_{i_2}) \oplus (\Delta_{i_3} \otimes [AB]_{i_3}) \oplus \dots \oplus (\Delta_{i_j} \otimes [AB]_{i_r})}{(\Delta_{i_1} \otimes [B]_{i_1}) \oplus (\Delta_{i_2} \otimes [B]_{i_2}) \oplus (\Delta_{i_3} \otimes [B]_{i_3}) \oplus \dots \oplus (\Delta_{i_r} \otimes [B]_{i_j})} \right], \quad (4.51)$$

A equação (4.51) compara individualmente qual é a maior diferença entre os caminhos com laços e caminhos sem laços entre cada entrada e as duas transições a serem controladas, e seleciona a maior dessas diferenças. Esse é o tempo que as duas transições precisam para estarem novamente habilitadas.

**Exemplo 4.3.4.** Considere novamente o  $GET_{MB}$  ilustrado pela Figura 4.14. Dando prosseguimento a sincronização de  $x_3$  e  $x_4$  para o segundo disparo, em  $k = 1$ . O objetivo é determinar-se a taxa de produção do sistema. Essa taxa é o maior tempo de habilitação necessário para que as transições possam disparar de forma sincronizada. A matrizes  $B$  e  $AB$  calculadas de acordo com a modelagem matricial do sistema, é dadas por:

$$B = A_0^* B_0 = \begin{bmatrix} 2 & \varepsilon \\ \varepsilon & 1 \\ 8 & \varepsilon \\ \varepsilon & 5 \\ 13 & 8 \\ 15 & 10 \end{bmatrix} \quad AB = \begin{bmatrix} 8 & \varepsilon \\ \varepsilon & 5 \\ 14 & 8 \\ 13 & 9 \\ 19 & 13 \\ 21 & 15 \end{bmatrix}$$

Assim, o primeiro disparo das transições em relação as entradas de alimentação, em  $k = 0$ , ocorrem nos instantes de tempo dados por:

$$[B]_3 = [8 \ \varepsilon], \quad (4.52)$$

$$[B]_4 = [\varepsilon \ 5]. \quad (4.53)$$

O valor  $\varepsilon$  significa que em  $k = 0$  as transições não dependem dos disparos das outras entradas de alimentação.

O segundo disparo das transições em relação as entradas de alimentação, em  $k = 1$ , ocorrem nos instantes de tempo dados por:

$$[AB]_3 = [14 \ 8], \quad (4.54)$$

$$[AB]_4 = [13 \ 9]. \quad (4.55)$$

A taxa é o maior intervalo de tempo entre os disparos das transições, consideradas conjuntamente em relação a cada entrada de alimentação. Os instantes de tempo de disparos de  $x_3$  e  $x_4$  em relação a  $u_1$  e  $u_2$  em  $k = 0$  são dados por  $[B]_3 \oplus [B]_4 = [8 \ 5]$ , e os instantes de tempo de disparos de  $x_3$  e  $x_4$  em relação a  $u_1$  e  $u_2$  em  $k = 1$  são dados por  $[AB]_3 \oplus [AB]_4 = [14 \ 9]$ . A maior das diferenças em relação a cada entrada de alimentação é:

$$\beta = (14 \not\phi 8) \oplus (9 \not\phi 5) = 6. \quad (4.56)$$

□

O procedimento para o cálculo do número de fichas do sistema de realimentação é o mesmo utilizado no método I. Dessa forma:

$$m = \lceil \frac{C \otimes B \otimes [0]}{\beta} \rceil \quad (4.57)$$

sendo  $[0]$  um vetor de zeros com dimensões compatíveis.

Essa condição já foi demonstrada para o método I, na demonstração da Proposição (4.2.2). Calcula-se o valor da saída mais rápida  $y(0)$  do sistema, em seguida é deduzida a expressão do valor de  $m$  quando  $k = m$ , garantindo que o controlador possua fichas até a saída do primeiro produto do sistema modelado por  $GET$ , considerando-se que a taxa de disparo do controlador para garantir a sincronização é dada por  $\beta$ .

### 4.3.3 Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Duas Transições

Considere um sistema de manufatura composto de oito postos de trabalho e três entradas de alimentação, ilustrado na Figura 4.16. O objetivo é calcular os parâmetros do controlador de forma a obedecer a condição de restrição  $x_4(k) = x_7(k)$ , sendo  $\Delta_4 = 0$  e  $\Delta_7 = 0$ , ou seja, as transições devem disparar nos mesmos instantes de tempo.

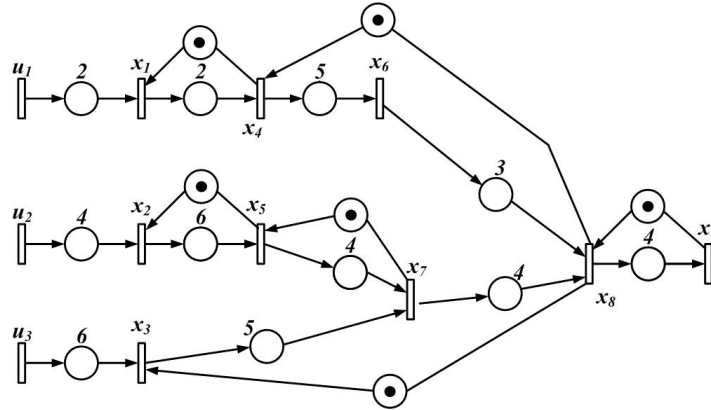


Figura 4.16: Sistema de Manufatura.

Equações recursivas:

$$x_1(k) = 2 \otimes u_1(k) \oplus x_4(k-1);$$

$$x_2(k) = 4 \otimes u_2(k) \oplus x_5(k-1);$$

$$x_3(k) = 6 \otimes u_3(k) \oplus x_8(k-1);$$

$$x_4(k) = 2 \otimes x_1(k) \oplus x_8(k-1);$$

$$x_5(k) = 6 \otimes x_2(k) \oplus x_7(k-1);$$

$$x_6(k) = 5 \otimes x_4(k);$$

$$x_7(k) = 5 \otimes x_3(k) \oplus 4 \otimes x_5(k);$$

$$x_8(k) = 3 \otimes x_6(k) \oplus 4 \otimes x_7(k) \oplus x_9(k-1);$$

$$x_9(k) = 4 \otimes x_8(k).$$

A modelagem matricial é dada por:

$$A_0 = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ 2 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 6 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & 4 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon \end{bmatrix} \quad A_1 = \begin{bmatrix} e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e \end{bmatrix}$$

$$B_0 = \begin{bmatrix} 2 & \varepsilon & \varepsilon \\ \varepsilon & 4 & \varepsilon \\ \varepsilon & \varepsilon & 6 \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$y(k) = C \otimes x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} x(k)$$

## Resultados

Dado um  $GET_{MB}$ , um conjunto de matrizes  $A_0$ ,  $A_1$  e  $B_0$ , os índices das transições a serem sincronizadas  $i = \{4, 7\}$  e o conjunto de restrições  $\Delta_i = \{0, 0\}$ . Procedeu-se ao cálculo dos parâmetros do controlador  $\beta$ ,  $[\varphi]$  e  $m$ , para  $x_4(k) = x_7(k)$ .

(a) Taxa de Disparo ( $\beta$ );

$$\beta = \left\lceil \left[ \frac{(\Delta_4 \otimes [B]_4) \oplus (\Delta_7 \otimes [B]_7)}{(\Delta_4 \otimes [AB]_4) \oplus (\Delta_7 \otimes [AB]_7)} \right] \right\rceil,$$

$$\lambda = \bigoplus_{j=1}^n (\text{trace}(A^j))^{1/j} = 9,$$

$$\beta = \beta \oplus \lambda = 13 \oplus 9 = 13.$$

(b) vetor com tempos de atrasos das entradas  $[\varphi]^T$ ;

$$[\varphi] = \frac{(\Delta_4 \otimes [B]_4) \oplus (\Delta_7 \otimes [B]_7)}{(\Delta_4 \otimes [B]_4 \otimes [0]) \oplus (\Delta_7 \otimes [B]_7 \otimes [0])},$$

$$[\varphi] = [10 \ 0 \ 3].$$

(c) Número de fichas do controlador ( $m$ ).

$$m = \left\lceil \frac{C \otimes B \otimes [0]}{\beta} \right\rceil,$$

$$m = \left\lceil \frac{22}{13} \right\rceil = 2.$$

É importante notar que a taxa de produção é modificada de modo a satisfazer estas restrições de tempo.

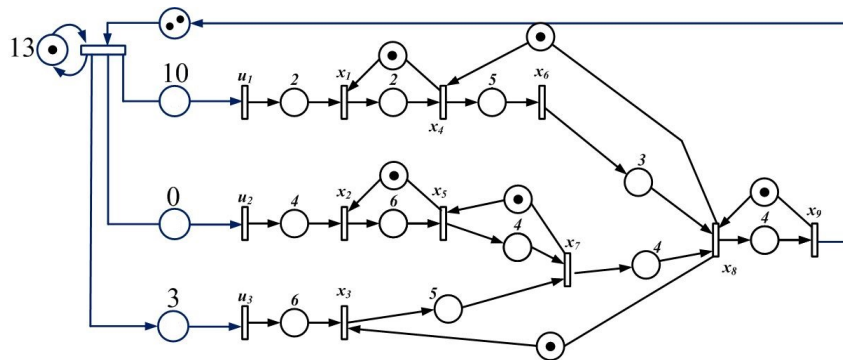


Figura 4.17: Resultado da aplicação do método de sincronização para duas transições.

Os instantes de disparo das transições são mostradas na Tabela (4.2) para a sincronização de duas transições.

Tabela 4.2: Tempos de disparos das transições para  $x_4(k) = x_7(k)$ .

Antes da Sincronização	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$x_4(k)$	4	18	27	36	45	54
$x_7(k)$	14	23	32	41	50	59
Após a Sincronização						
$x_4(k)$	14	27	40	53	66	79
$x_7(k)$	14	27	40	53	66	79

#### 4.3.4 Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Três Transições

Dado um  $GET_{MB}$  (Figura 4.16), um conjunto de matrizes  $A_0$ ,  $A_1$  e  $B_0$ , os índices das transições a serem sincronizadas  $i = \{3, 4, 5\}$  e o conjunto de restrições  $\Delta_i = \{3, 4, 0\}$ . Proceda-se ao cálculo dos parâmetros do controlador  $\beta$ ,  $[\varphi]$  e  $m$ , para  $3 \otimes x_3(k) = 4 \otimes x_4(k) = 0 \otimes x_5(k)$ .

##### Resultado

(a) Taxa de Disparo ( $\beta$ );

$$\beta = \left\lceil \left[ \frac{(\Delta_3 \otimes [B]_3) \oplus (\Delta_4 \otimes [B]_4) \oplus (\Delta_5 \otimes [B]_5)}{(\Delta_3 \otimes [AB]_3) \oplus (\Delta_4 \otimes [AB]_4) \oplus (\Delta_5 \otimes [AB]_5)} \right] \right\rceil,$$

$$\lambda = \bigoplus_{j=1}^n (\text{trace}(A^j))^{1/j} = 9,$$

$$\beta = \beta \oplus \lambda = 12 \oplus 9 = 12.$$

(b) vetor com tempos de atrasos das entradas  $[\varphi]^T$ ;

$$[\varphi] = \frac{(\Delta_3 \otimes [B]_3) \oplus (\Delta_4 \otimes [B]_4) \oplus (\Delta_5 \otimes [B]_5)}{(\Delta_3 \otimes [B]_3 \otimes [0]) \oplus (\Delta_4 \otimes [B]_4 \otimes [0]) \oplus (\Delta_5 \otimes [B]_5 \otimes [0])},$$

$$[\varphi] = [2 \quad 0 \quad 1].$$

(c) Número de fichas do controlador ( $m$ ).

$$m = \left\lceil \frac{C \otimes B \otimes [0]}{\beta} \right\rceil,$$

$$m = \left\lceil \frac{22}{13} \right\rceil = 2.$$

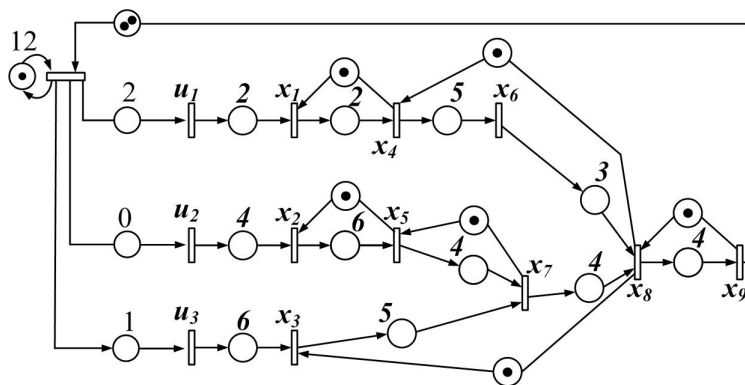


Figura 4.18: Resultado para sincronização de três transições,  $3 \otimes x_3(k) = 4 \otimes x_4(k) = x_5(k)$ .

Os instantes de disparo das transições são mostradas na Tabela (4.3) para a sincronização de três transições.

Tabela 4.3: Tempos de disparos das transições para  $3 \otimes x_3(k) = 4 \otimes x_4(k) = x_5(k)$ .

Antes da Sincronização	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$x_3(k)$	6	18	27	36	45	54
$x_4(k)$	4	18	27	36	45	54
$x_5(k)$	10	16	23	32	41	50
Após a Sincronização						
$x_3(k)$	7	19	31	43	55	67
$x_4(k)$	6	18	30	42	54	66
$x_5(k)$	10	22	34	46	58	70

#### 4.3.5 Exemplo Ilustrativo da Aplicação do Método II para a Sincronização de Quatro Transições

Dado um  $GET_{MB}$  (Figura 4.19), composto de oito postos de trabalho e quatro entradas de alimentação, um conjunto de matrizes  $A_0$ ,  $A_1$  e  $B_0$ , os índices das transições a serem sincronizadas  $i = \{5, 6, 7, 8\}$  e o conjunto de restrições  $\Delta_i = \{0, 0, 0, 0\}$ . Proceda-se ao cálculo dos parâmetros do controlador  $\beta$ ,  $[\varphi]$  e  $m$ , para  $x_5(k) = x_6(k) = x_7(k) = x_8(k)$ .





$$B_0 = \begin{bmatrix} 2 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & 3 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 1 & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & e \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon \end{bmatrix}$$

$$y(k) = C \otimes x(k) = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 \end{bmatrix} x(k)$$

### Resultados

Parâmetros do controlador para  $x_5(k) = x_6(k) = x_7(k) = x_8(k)$ ,  $\Delta_5 = \Delta_6 = \Delta_7 = \Delta_8 = 0$ .

(a) Taxa de Disparo ( $\beta$ );

$$\beta = \left\lceil \left[ \frac{(\Delta_5 \otimes [B]_5) \oplus (\Delta_6 \otimes [B]_6) \oplus (\Delta_7 \otimes [B]_7) \oplus (\Delta_8 \otimes [B]_8)}{(\Delta_5 \otimes [AB]_5) \oplus (\Delta_6 \otimes [AB]_6) \oplus (\Delta_7 \otimes [AB]_7) \oplus (\Delta_8 \otimes [AB]_8)} \right] \right\rceil,$$

$$\beta = 11,$$

$$\lambda = \bigoplus_{j=1}^n (\text{trace}(A^j))^{1/j} = 9,$$

$$\beta = \beta \oplus \lambda = 11 \oplus 9 = 11.$$

(b) vetor com tempos de atrasos das entradas  $[\varphi]_{1 \times l}$ ;

$$[\varphi] = \frac{(\Delta_5 \otimes [B]_5) \oplus (\Delta_6 \otimes [B]_6) \oplus (\Delta_7 \otimes [B]_7) \oplus (\Delta_8 \otimes [B]_8)}{(\Delta_5 \otimes [B]_5 \otimes [0]) \oplus (\Delta_6 \otimes [B]_6 \otimes [0]) \oplus (\Delta_7 \otimes [B]_7 \otimes [0]) \oplus (\Delta_8 \otimes [B]_8 \otimes [0])},$$

$$[\varphi] = [1 \ 2 \ 0 \ 6].$$

(c) Número de fichas do controlador ( $m$ ).

$$m = \left\lceil \frac{C \otimes B \otimes [0]}{\beta} \right\rceil,$$

$$m = \left\lceil \frac{12}{11} \right\rceil = 2.$$

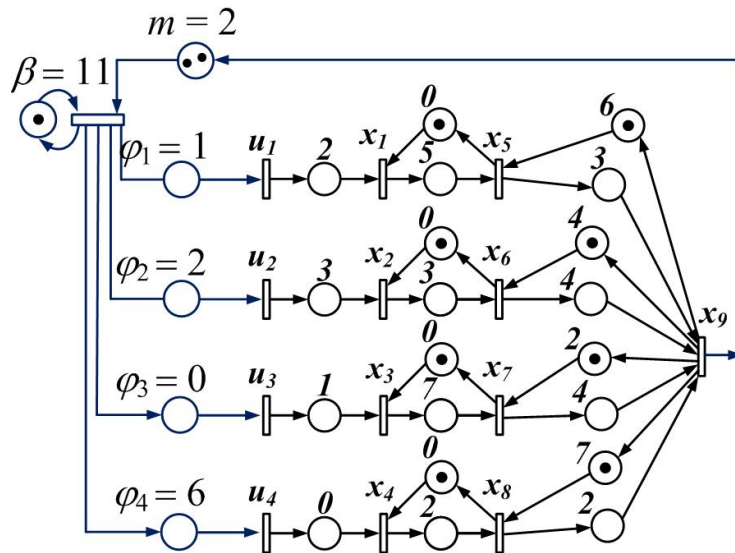


Figura 4.20: Resultado para a Sincronização de Quatro Transições Disparando no mesmo Instante de Tempo.

Tabela 4.4: Tempos do disparos das transições para  $x_5(k) = x_6(k) = x_7(k) = x_8(k)$ .

Antes da Sincronização	$k = 0$	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$
$x_5(k)$	7	18	27	36	45	54
$x_6(k)$	6	16	35	34	43	52
$x_7(k)$	8	15	23	32	41	50
$x_8(k)$	7	19	28	37	46	55
Após a Sincronização						
$x_5(k)$	8	19	36	41	52	63
$x_6(k)$	8	19	36	41	52	63
$x_7(k)$	8	19	36	41	52	63
$x_8(k)$	8	19	36	41	52	63

### 4.3.6 Outro Exemplo Ilustrativo para a Sincronização de Quatro Transições

Considerando o mesmo sistema de manufatura do exemplo anterior, composto de oito postos de trabalho e quatro entradas de alimentação ilustrado na Figura 4.19. A condição de restrição agora imposta será  $\Delta_i = \{3, 1, 0, 5\}$ , de forma que  $3 \otimes x_5(k) = 1 \otimes x_6(k) = 0 \otimes x_7(k) = 5 \otimes x_8(k)$ .

#### Resultados

Parâmetros do controlador para  $3 \otimes x_5(k) = 1 \otimes x_6(k) = 0 \otimes x_7(k) = 5 \otimes x_8(k)$ ,  $\Delta_5 = 3$ ,  $\Delta_6 = 1$ ,  $\Delta_7 = 0$  e  $\Delta_8 = 5$ .

(a) Taxa de Disparo ( $\beta$ );

$$\beta = \left\lceil \left[ \frac{(\Delta_5 \otimes [B]_5) \oplus (\Delta_6 \otimes [B]_6) \oplus (\Delta_7 \otimes [B]_7) \oplus (\Delta_8 \otimes [B]_8)}{(\Delta_5 \otimes [AB]_5) \oplus (\Delta_6 \otimes [AB]_6) \oplus (\Delta_7 \otimes [AB]_7) \oplus (\Delta_8 \otimes [AB]_8)} \right] \right\rceil,$$

$$\beta = 16,$$

$$\lambda = \bigoplus_{j=1}^n (\text{trace}(A^j))^{1/j} = 9,$$

$$\beta = \beta \oplus \lambda = 16 \oplus 9 = 16.$$

(b) vetor com tempos de atrasos das entradas  $[\varphi]_{1 \times l}$ ;

$$[\varphi] = \frac{(\Delta_5 \otimes [B]_5) \oplus (\Delta_6 \otimes [B]_6) \oplus (\Delta_7 \otimes [B]_7) \oplus (\Delta_8 \otimes [B]_8)}{(\Delta_5 \otimes [B]_5 \otimes [0]) \oplus (\Delta_6 \otimes [B]_6 \otimes [0]) \oplus (\Delta_7 \otimes [B]_7 \otimes [0]) \oplus (\Delta_8 \otimes [B]_8 \otimes [0])},$$

$$[\varphi] = [0 \quad 3 \quad 2 \quad 3].$$

(c) Número de fichas do controlador ( $m$ ).

$$m = \left\lceil \frac{C \otimes B \otimes [0]}{\beta} \right\rceil,$$

$$m = \left\lceil \frac{12}{16} \right\rceil = 1.$$

Os instantes de disparos das transições são mostrados na Tabela (4.4) para a sincronização de quatro transições.

#### 4.3.7 Considerações sobre o Método II

Nesta seção foi desenvolvida e tratada a questão da sincronização do disparo de várias transições, obtendo-se ótimos resultados quando o método de controle proposto foi aplicado. A equação geral para o cálculo dos parâmetros do controlador foi obtida para uma classe de sistemas modelados por GET, utilizando a álgebra max-plus.

O Método II foi aplicado para três casos de sincronização, e foi mostrado que para tornar possível a sincronização, a taxa de produção do sistema foi modificada, modificando ao mesmo tempo o desempenho inicial do sistema, proporcionando a redução das perdas de matéria-prima.

A definição dessas novas taxas de produção é de fundamental importância para um sistema produtivo, que precisa ter tempos de atraso para operações como de resfriamento, secagem de material, bem como união de componentes utilizando-se adesivos ou colas com rápido tempo de cura.

## Capítulo 5

# Conclusões

### 5.1 Considerações Finais e Perspectivas

Ao finalizar a apresentação dos métodos que foram analisadas e desenvolvidas nesta tese, conclui-se que o problema do controle foi rigorosamente formulado como um problema multi-objetivo de otimização, obtendo-se resultado satisfatório. Mostra-se também que o controlador assegura estabilidade do sistema e que o método é eficiente computacionalmente (complexidade polinomial). O método I tratado neste trabalho mostrou resultados satisfatórios para os problemas de controle propostos, ao mesmo tempo em que garante o melhor controle para calcular as entradas de matéria-prima em aplicações para sistemas produtivos. Mostrou-se matematicamente que o método I apresenta resultados iguais ou até melhores que o resultados apresentados em [CHBF01], [CHBF99], [MMLH05] e em [Mai03], que usam funções de transferência para calcular o melhor controle das entradas de alimentação.

A vantagem de usar o método I é que não será necessário modificar as equações de datadores para série de potências formal, o que reduz consideravelmente o tempo de modelagem do sistema. Apresentando-se muito mais vantajoso trabalhar com datadores do que com funções de transferência, por razões de simplicidade na implementação dos sistemas de controle. Os parâmetros do controlador são encontrados sem a necessidade de implementação de algoritmos complexos. Além disso, mostra-se que o método I apresenta desempenho superior à síntese de realimentação usada no modelo de referência com realimentação.

Para o método II, no problema de sincronização dos disparos das transições, foi possível estabelecer uma equação geral para tratar dessa questão, pois muitas vezes a qualidade, o tempo de espera e a sincronização em determinadas atividades industriais superam a necessidade de produção em massa. Em situações práticas, muitos produtos precisam de tempo de cura, acomodação, secagem e resfriamento para efetivamente serem empregados numa sequência de produção.

Os métodos de controle propostos, pela simplicidade de implementação, são vantajosos para as empresas, já que estas não podem ater-se a resolver equações matemáticas complexas para encontrar, por exemplo, os parâmetros de um controlador lógico programável que controla uma máquina de soldagem de componente eletrônicos, ou saber quanto de material precisa ser produzido pelas máquinas injetoras para atender as necessidades das linhas de produção.

No que diz respeito à implementação dos métodos, o grafo de eventos temporizados se apresenta

como uma ferramenta gráfica apropriada para modelagem e análise de problemas de sistemas de manufatura ou de fluxo de dados, e aplicado juntamente com a álgebra max-plus pode definir todo o comportamento do sistema cíclico de um ambiente industrial, por sua vez em conjunto com as técnicas já implantadas nas grandes indústrias, como JIT, Kanban, e outras, mostra-se eficiente para propor melhorias e oferece reduções dos custos operacionais, com qualidade e produtividade. Uma das bases de sustentação das políticas de qualidade é a absoluta eliminação do desperdício, investindo-se na política de gestão JIT e na automação dos processos.

A principal contribuição deste trabalho é proporcionar dois métodos rápidos e eficazes para o cálculo dos parâmetros de um controlador que garantam a estabilidade do sistema produtivo sem comprometer a produtividade do sistema. Este controlador pode também ser entendido como um mecanismo de agendamento de toda atividade industrial, desde a compra de material até o armazenamento final dos produtos dentro da fábrica. É possível, por este método, prever a quantidade de material que será necessário para o sistema produtivo operar sem necessidade de grandes volumes de materiais em estoque, atendendo as demandas do sistema.

Algumas perspectivas para a a extensão deste trabalho são listadas a seguir.

- Obtenção de outras condições de controle para implementação utilizando-se datadores.
- Análise de robustez do desempenho dos métodos quando são consideradas falhas inesperadas no sistema produtivo.
- Implementação dos métodos em sistemas com variáveis probabilísticas (ambiente estocástico), considerando-se que as temporizações são variáveis aleatórias.
- Adaptação do método II para sistemas em que a capacidade de operação das máquinas seja maior do que uma peça por vez.

## Apêndice A

### Controle por Modelo de Referência

O problema de controle de um GET é usualmente formulado tendo como objetivo obter disparos para as transições de entrada as mais tardias possíveis, sem entretanto violar especificações de disparos mínimos para as transições de saída. Esta formulação do problema ganha apelo intuitivo ao se considerar a estratégia JIT em sistemas de manufatura. Nela deseja-se operar a planta de modo que os estoques (de matéria-prima) em seu interior sejam mínimos, mas garantindo que sua produção atenda à demanda pré-especificada, ou seja, que as datas devidas para o produtos acabados sejam cumpridas. Um caso particular de importância é aquele em que se deseja, do ponto de vista da saída, que a planta se comporte como se não houvesse limitações de matéria-prima, trabalhando portanto com sua máxima capacidade de produção. Para que isso aconteça não é necessário, em geral, que os recursos estejam todos disponíveis ao se iniciar a produção, mas sim no momento exato em que puderem ser processados. Do ponto de vista de um GET isso significa que as trajetórias das transições de saída devem ser idênticas àquelas obtidas quando as transições de entrada são impulsivas, isto é, em  $t = 0$  as transições de entrada disparam infinitas vezes. Para que essas saídas sejam obtidas não é necessário entretanto (em geral) que as entradas sejam impulsivas. A questão é portanto, obter a trajetória mais tardia possível na entrada, sem que a trajetória da saída tenha seu desempenho alterado [MMLH05].

Considere agora um GET modelado utilizando-se as equações de datadores com suas respectivas transformadas- $\gamma$ , resultando em expressões do tipo dado pelas equações  $x(\gamma) = A(\gamma)x(\gamma) \oplus B(\gamma)u(\gamma)$  e  $y(\gamma) = C(\gamma)x(\gamma)$ . A matriz de transferência do sistema é dada por  $H = CA^*B$ , de modo que  $y = Hu$ . Seja também  $z \in \overline{\mathbb{Z}}_{max}[\gamma]$  uma trajetória que represente o desempenho aceitável para a transição de saída. É imediato constatar que dadas duas trajetórias  $a, b \in \overline{\mathbb{Z}}_{max}[\gamma]$ , se  $a \succeq b$  então a trajetória  $a$  é mais lenta que a trajetória  $b$ . O problema de obter a trajetória de controle ótima é então equivalente ao de obter a máxima trajetória  $u$  tal que  $z \succeq y = Hu$ . Como a função  $Hu$  é residuável, o resultado é  $u_{op}$  que é a máxima solução da inequação, e  $u_{op} = H \setminus z$ . Em particular, se se deseja que a planta opere em sua máxima taxa de produção, a trajetória  $z$  deve ser igual a  $Hu$  sendo  $u$  a entrada impulsiva. Conseqüentemente, para o caso de produção máxima,  $z = H$  e portanto  $u_{otimo} = H \setminus H$ . Deve-se observar que a relação  $H \setminus H \succeq e$  é sempre verificada de modo que a entrada ótima é, como esperado, mais lenta que a entrada impulsiva [MMLH05].

Uma abordagem usual em projeto de sistemas de controle é a alteração da matriz de trans-



ferência da planta através de seu acoplamento a um outro sistema, usualmente chamado de compensador. No caso de sistemas Max-plus, dois aspectos devem ser levados em conta, ao se considerar essa abordagem [MMLH05].

Em primeiro lugar deve-se determinar sobre que transições ocorrerá a ação de controle, isto é, quais são as transições "controláveis" no sentido de permitirem sua inibição por um dispositivo externo à planta. Em relação a esse aspecto, duas alternativas são consideradas. A primeira delas parte do pressuposto de que apenas a variável de entrada  $u(\gamma)$  é controlável (no sentido definido acima) cabendo portanto calculá-la a partir de alguma outra informação externa. Nos sistemas de manufatura, por exemplo, esta informação externa pode estar associada à disponibilidade de recursos no ambiente externo à planta. A variável  $v(\gamma)$  será associada a esta informação, sendo chamada de "sinal de referência". A segunda alternativa é considerar um conjunto arbitrário de transições da planta como sendo controláveis. Nesse caso, a transição de entrada ( $u(\gamma)$ ) permanece sendo aquela que sofre as influências do ambiente externo à planta [MMLH05].

Em segundo lugar deve-se determinar qual deverá ser o desempenho do sistema sob a ação do controlador. De modo geral, é possível calcular a função de transferência entre as saídas do sistema e as variáveis que sofrem influências externas ao sistema ( $u(\gamma)$  ou  $v(\gamma)$ ) levando-se em conta a presença do compensador. O desempenho do sistema será então especificado a partir da determinação de uma função de transferência de referência, notada por  $G_{ref}(\gamma)$  e também denominada "modelo de referência", com a mesma dimensão da função de transferência do sistema. Determinados estes dois aspectos, o compensador deve então ser calculado de modo a ser máximo (i.e. o controlador deve retardar ao máximo sua saída), mas garantindo que o comportamento do sistema sob a ação do controle permaneça limitado superiormente pelo modelo de referência. Em outras palavras, esse comportamento não pode ser mais lento que o imposto pelo modelo de referência [MMLH05].

Observa-se que maximizar o controlador não é exatamente o mesmo que maximizar o sinal de controle, pois ainda pode existir um sinal de controle maior, mas que eventualmente não pode ser obtido pelo controlador máximo. Entretanto, de modo geral, quanto "maior" o controlador, maior o atraso ele provoca entre sua entrada e sua saída. Em muitos casos pode-se provar que esta estratégia leva a resultados iguais aos obtidos pela otimização do sinal de controle [Mai03] [MMLH05].

Portanto o projeto de controladores é feito de modo a satisfazer a restrição imposta pelo modelo de referência maximizando os controladores e por isso retardando os disparos das transições sobre as quais se dá a ação de controle [MMLH05].

## A.1 Síntese de Controladores

Na seção anterior, o problema de controle por modelo de referência foi estabelecido na sua forma geral. Nessa seção, são apresentadas as estruturas de controle utilizadas. A pré-compensação e a realimentação de saída têm sido amplamente estudadas na literatura.

### A.1.1 Pré-compensação

Uma primeira estratégia para o controle por modelo de referência consiste na utilização de pré-compensação. Nesse caso, o problema de controle é encontrar um pré-compensador tal que a

entrada do sistema seja maximizada respeitando a restrição de que a função de transferência do sistema controlado satisfaça as limitações impostas pelo modelo de referência. Essa estratégia é ilustrada na Figura A.1 [CHBF99] [MMLH05].

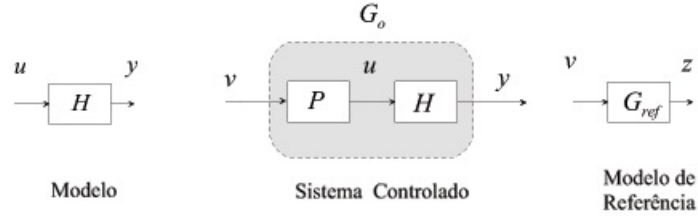


Figura A.1: Controle por pré-compensação [MMLH05].

Formalmente, pode se mostrar que a solução para o problema equivale a calcular o maior pré-compensador  $P$  tal que  $HP \preceq G_{ref}$ . Como a função  $f(x) = a \otimes x$  é residuável [BCOQ92], mostra-se que a maior solução dessa inequação, denotada por  $P_{op}$ , é dada por [MMLH05]:

$$P_{op} = H \setminus G_{ref}. \quad (A.1)$$

Essa estratégia de controle em malha aberta resulta na ação de controle  $u = P_{op}v$  que assegura sempre desempenho ótimo do sistema. Contudo, não se assegura a estabilidade robusta do sistema em relação a variações paramétricas, isto é, variações dos parâmetros temporais da planta podem comprometer a estabilidade do sistema. De fato, considere um sistema Max-plus linear tal que  $H = 1(7\gamma)^*$ . Esse sistema modela uma máquina que opera em ciclos com uma taxa de produção igual a uma peça a cada 7 unidades de tempo e com um atraso de transporte para disponibilização final igual a 1 unidade de tempo. Esse modelo é mostrado em linhas pontilhadas na Figura A.2(a). Considere também que a restrição de demanda é dada por um modelo de referência tal que  $G_{ref} = H$ , ou seja, o objetivo é preservar a função de transferência em malha aberta maximizando a entrada do sistema. Dessa forma, dentro de um contexto de gestão de recursos JIT, preserva-se a máxima taxa de produção do sistema retardando ao máximo possível a entrada de matéria prima a ser processada. Conforme visto anteriormente, o problema se resume a encontrar o maior controlador  $P$  tal que  $HP \preceq G_{ref}$ . A solução para esse problema é dada pela Teoria de Residuação  $P = H \setminus H = (7\gamma)^*$  [MMLH05].

É importante observar que qualquer outro modelo de referência pode ser utilizado. Por exemplo, adotando-se  $G_{ref} = 1(8\gamma)^*$ , o pré-compensador será dado por  $P = (8\gamma)^*$  [MMLH05].

Voltando ao caso em que  $G_{ref} = H$  (e  $P = (7\gamma)^*$ ), pode-se observar, a partir da Figura A.2, que o sistema é sensível a variações paramétricas, o que pode ocasionar problemas de instabilidade. No presente exemplo, se a temporização do ciclo aumentar (por exemplo por algum problema de manutenção da máquina) o pré-compensador se torna mais rápido do que o sistema e o número de

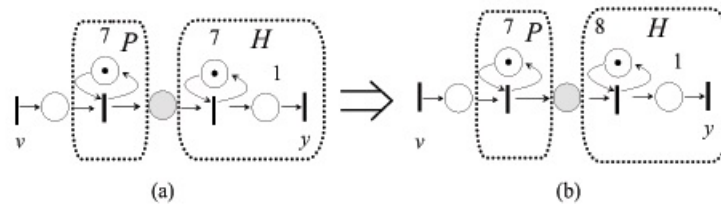


Figura A.2: Problema de estabilidade na pré-compensação [MMLH05].

recursos no lugar situado entre  $P$  e  $H$  cresce ilimitadamente [MMLH05].

### A.1.2 Realimentação de saída

Sabe-se da Teoria de Controle de Sistemas Lineares Contínuos que a realimentação torna o sistema controlado menos sensível a variações paramétricas da planta, contudo, o sistema em malha fechada pode se tornar instável. Nos SED Max-plus lineares, a estratégia de controle via realimentação sempre favorece a estabilidade, conforme mostrado por Maia (2003). Nesse sentido, para se contornar o problema de instabilidade, assegurando um sistema mais robusto (menos sensível a variações paramétricas), recomenda-se a utilização da estratégia de realimentação para o controle por modelo de referência. O exemplo abaixo mostra como a realimentação pode favorecer a estabilidade do sistema [MMLH05] [CHBF01].

Considere o mesmo sistema do exemplo anterior, ou seja,  $H = 1(7\gamma)^*$  com realimentação dada por  $F = 6\gamma(7\gamma)^*$ . O sistema em malha fechada é mostrado na Figura A.3(a). Mostra-se, conforme será visto a seguir, que essa realimentação preserva a função de transferência  $H$  do sistema. É interessante observar que a estabilidade do sistema em malha fechada é independente das variações paramétricas, pois a admissão de matéria prima é condicionada à saída de produtos do sistema. De fato, vê-se na Figura A.3(b) que o aumento da temporização do ciclo da máquina não acarreta explosão do número de recursos no sistema (na verdade eles permanecem constantes) [MMLH05].

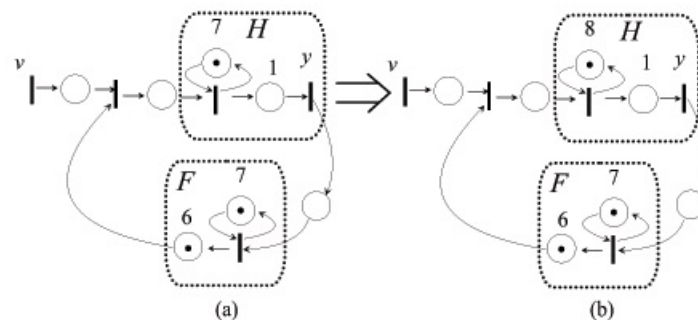


Figura A.3: Estabilidade robusta da realimentação [MMLH05].

A Figura A.4 ilustra de maneira geral a estrutura de controle com realimentação da saída. Na Figura A.4,  $H$  é função de transferência da planta,  $F$  é o controlador e  $u$  e  $y$  são respectivamente a

entrada e a saída do sistema;  $G_{ref}$  é o modelo de referência e  $v$  e  $z$  são respectivamente a entrada de referência (disponibilidade de matéria prima) e  $z$ , a saída do modelo de referência [MMLH05].

A partir da análise Figura A.4, obtém-se as seguintes equações [MMLH05]:

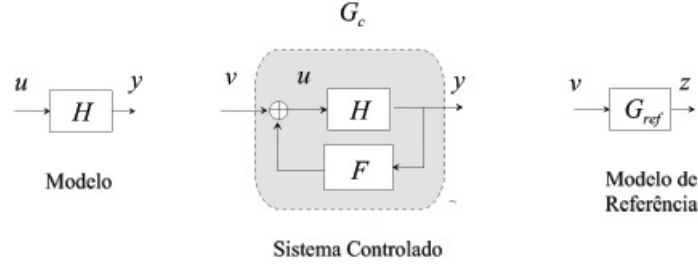


Figura A.4: Controle por realimentação [MMLH05].

$$u = v \oplus Fy, \quad (\text{A.2})$$

$$y = Hu = HFy \oplus Hv. \quad (\text{A.3})$$

A solução da Equação A.3 é [MMLH05]:

$$y = (HF)^*Hv. \quad (\text{A.4})$$

Dessa forma,  $u = (e \oplus F(HF)^*H)v$ . A Equação  $a(ba)^* = (ab)^*a$  [BCOQ92] garante que  $(HF)^*H = H(FH)^*$ . Portanto  $u = (e \oplus FH(FH)^*)v = (e \oplus FH \oplus (FH)^2 \oplus \dots)v = (FH)^*v$ . Consequentemente:

$$u = (FH)^*v. \quad (\text{A.5})$$

O problema de controle para a estratégia de realimentação consiste então em maximizar  $u = (FH)^*v$  tal que a função de transferência em malha fechada  $H(FH)^*$  seja menor ou igual ao modelo de referência  $G_{ref}$ . Em decorrência da isotonia dos operadores de adição e de multiplicação do dióide, esse problema equivale a encontrar a maior realimentação  $F$  tal que  $H(FH)^*$  seja menor ou igual a referência  $G_{ref}$ .

Formalmente, o que se procura é a maior solução para a inequação  $H(FH)^* \preceq G_{ref}$ . Esse problema não tem solução máxima para qualquer modelo de referência, pois a função  $G(X) = H(XH)^*$  não é sempre residuável. Contudo, a solubilidade pode ser assegurada se restrições paramétricas são impostas ao modelo de referência. demonstraram que se  $G_{ref}$  é da forma  $A^*H$  ou da forma  $HB^*$  (sendo  $A$  e  $B$  matrizes de transferência quaisquer de dimensões apropriadas), então, o melhor controlador de realimentação<sup>1</sup> é dado por  $F_r$ , que atrasa ao máximo a entrada de fichas no sistema, calculado por [CHBF99]:

$$F_r = Pr_+(H \setminus G_{ref} \setminus H). \quad (\text{A.6})$$

<sup>1</sup>Programas/soluções de aplicações utilizando-se modelo de referência são encontrados em <http://perso-laris.univ-angers.fr/hardouin/outils.html>



## Referências Bibliográficas

- [ADLM12] S. Amari, I. Demongodin, J. J. Loiseau, and C. Martinez, *Max-plus control design for temporal constraints meeting in timed event graphs*, IEEE Transactions on Automatic Control **57** (2012), 462–467.
- [Att98] S. S. Attié, *Automação hidráulica e pneumática empregando a teoria de sistemas a eventos discretos*, Master’s thesis, Universidade Federal de Santa Catarina, Florianópolis, Brasil, 1998.
- [Ban99] J. Banks, *Introduction to simulation*, Proceedings of the 31st Conference on Winter Simulation: Simulation - a bridge to the future (New York, NY, USA), WSC ’99, ACM, 1999, pp. 7–13.
- [BCOQ92] F. Baccelli, G. Cohen, G. J. Olsder, and J. P. Quadrat, *Synchronization and linearity*, Wiley, 1992.
- [BF96] J. L. Boimond and J. L. Ferrier, *Internal model control and max-algebra: controller design*, IEEE Transactions on Automatic Control **41** (1996), no. 3, 457–461.
- [BJ72] T. S. Blyth and M. F. Janowitz, *Residuation theory*, Oxford: Pergamon Press, 1972.
- [Bla98] J. T. Black, *O projeto da fábrica do futuro*, Bookman, 1998.
- [BM91] G. Basile and G. Marro, *Controlled and conditioned invariants in linear system theory*, Prentice Hall, Italy, 1991.
- [BRH12] T. Brunsch, J. Raisch, and L. Hardouin, *Modeling and control of high-throughput screening systems*, Control Eng. Practice **20** (2012), no. 1, 14–23.
- [CCN10] S. L. Campbell, J. P. Chancelier, and R. Nikoukhah, *Modeling and simulation in scilab/scicos with scicoslab 4.4*, second ed., Springer Science, 2010.
- [CG79] R. A. Cuninghame-Green, *Minimax algebra lecture notes in economics and mathematical systems*, Springer, 1979.
- [CG93] L. H. Corrêa and I. Gianesi, *Just-in-time, mrp ii e opt : um enfoque estratégico*, Atlas, São Paulo, 1993.
- [CGB03] R. A. Cuninghame-Green and P. Butkovic, *The equation  $A \otimes x = B \otimes y$  over  $(\max, +)$* , Theor. Comput. Sci. **293** (2003), 3–12.
- [CHBF99] B. Cottenceau, L. Hardouin, J.-L. Boimond, and J.-L. Ferrier, *Synthesis of greatest linear feedback for timed-event graphs in dioid*, IEEE Transactions on Automatic Control **44** (1999), no. 6, 1258–1262.

- [CHBF01] B. Cottenceau, L. Hardouin, J.-L. Boimond, and J.-L. Ferrier, *Model reference control for timed event graphs in dioids*, *Automatica* **37** (2001), no. 9, 1451–1458.
- [CL99] C. G. Cassandras and S. Lafortune, *Introduction to discrete event systems*, Kluwer Academic Publishers, 1999.
- [CLH14] B. Cottenceau, S. Lahaye, and L. Hardouin, *Modeling of time-varying (max,+) systems by means of weighted timed event graphs*, 12th IFAC International Workshop on Discrete Event Systems, WODES 2014 (2014).
- [CLHB03] B. Cottenceau, M. Lhommeau, L. Hardouin, and J.-L. Boimond, *On timed event graph stabilization by output feedback in dioid*, *KYBERNETIKA* **39** (2003), no. 2, 165–176.
- [Coh01] G. Cohen, *Análisis y control de sistemas de eventos discretos: de redes de petri temporizadas al álgebra*, Universidad Nacional de Rosario - Facultad de Ciencias Exactas, Ingeniería y Agimensura - Cuadernos Del Instituto de Matematica Beppo Levi, Rosario - República Argentina, 2001.
- [Com98] C. Commault, *Feedback stabilization of some event graph models*, *IEEE Transactions on Automatic Control* **43** (1998), no. 10, 1419–1423.
- [Cur01] J. E. R. Cury, *Teoria de controle supervisorio de sistemas a eventos discretos*, Tech. report, UFSC - Depto de Automação e Sistemas, Canela-RS, Novembro 2001.
- [DHHRC13] X. David-Henriet, L. Hardouin, J. Raisch, and B. Cottenceau, *Optimal control for timed event graphs under partial synchronization*, Conference on Decision and Control, CDC 2013 (2013).
- [DHHRC14] ———, *Modeling and control for max-plus systems with partial synchronization*, 12th IFAC International Workshop on Discrete Event Systems, WODES 2014 (2014).
- [DML12] J. R. S. Dias, C. A. Maia, and V. F. Lucena, *Método para controle de fluxo de entrada em sistemas max-plus lineares garantindo uma dada taxa de produção*, Anais do XIX Congresso Brasileiro de Automática, CBA 2012. (2012), 1157–1164.
- [DML13] ———, *Control of input stream of manufacturing systems modeled by timed event graphs and max-plus algebra*, 6th IFAC Conference on Management and Control of Production and Logistics **6** (2013), 478–485.
- [FGS04] A. Di Febbraro, D. Giglio, and N. Sacco, *Urban traffic control structure based on hybrid petri nets*, *IEEE Transactions on Intelligent Transportation Systems* **5** (2004), no. 4, 224–237.
- [Gar07] T. R. Garcia, *Modelagem e controle da coordenação do tráfego urbano através de formalismos para sistemas a eventos discretos e híbridos*, Ph.D. thesis, Universidade Federal de Santa Catarina, 2007.
- [Gau97] S. Gaubert, *Methods and applications of (max,+) linear algebra*, STACS 1997, Lecture Notes in Computer Science 500 (1997), 261–282.
- [Ghi95] P. Ghinato, *Sistema toyota de produção - mais do que simplesmente just in time*, *Revista Produção* **5** (1995), no. 2, 169–190.

- [Gon01] E. V. Gonçalves, *Arranjo físico da fábrica: Um modelo para o processo de projeto e um algoritmo genérico para a formação de células de fabricação.*, Ph.D. thesis, Escola de Engenharia de São Carlos, 2001.
- [Gov07] R. M. P. Goverde, *Railway timetable stability analysis using max-plus system theory*, Transportation Research Part B: Methodological **41** (2007), no. 2, 179–201.
- [Her97] S. S. Heragu, *Facilities design*, Miscellaneous/Catalogs Series, no. 1, PWS Pub. Co., 1997.
- [HOvdW06] B. Heidergott, G. J. Olsder, and J. van der Woude, *Max plus at work*, Princeton Séries in Applied mathematics, 2006.
- [HS07] R. G. Halburd and N. J. Southall, *Tropical nevanlinna theory and ultra -discrete equations*, Loughborough University, 2007.
- [Çin75] E. Çinlir, *Introduction to stochastic processes*, Prentice-Hall, Englewood Cliffs, N.J., USA., 1975.
- [Kat07] R. D. Katz, *Max-plus (a,b)-invariant spaces and control of timed discrete-event systems*, IEEE Transactions on Automatic Control **52** (2007), no. 2, 229–241.
- [KH91] B. H. Krogh and L. E. Holloway, *Synthesis of feedback control logic for discrete manufacturing systems.*, Automatica **27** (1991), no. 4, 641–651, NewsletterInfo: 40.
- [KL12] C. Kim and T.E. Lee, *Feedback control design for cluster tools with wafer residency time constraints*, IEEE International Conference on Systems, Man, and Cybernetics (2012).
- [Kle56] S. C. Kleene, *Representation of events in nerve nets and finite automata*, Automata Studies (1956).
- [Kle75] L. Kleinrock, *Theory queueing systems*, Wiley-Interscience, 1975.
- [LC04] G. F. List and M. Cetin, *Modeling traffic signal control using petri nets*, IEEE Transactions on Intelligent Transportation Systems **5** (2004), no. 3, 177–187.
- [LHCJ04] M. Lhommeau, L. Hardouin, B. Cottenceau, and L. Jaulin, *Interval analysis and dioid: application to robust controller design for timed event graphs*, Automatica **40** (2004), 1923–1930.
- [MAH11] C. A. Maia, C. R. Andrade, and L. Hardouin, *On the control of max-plus linear system subject to state restriction*, Automatica **47** (2011), no. 5, 988–992.
- [Mai03] C. A. Maia, *Identificação e controle de sistemas a eventos discretos na álgebra (max, +)*, Ph.D. thesis, UNICAMP, 2003.
- [MBHF00] E. Menguy, J. L. Boimond, L. Hardouin, and J. L. Ferrier, *Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input*, IEEE Transactions on Automatic Control **45** (2000), no. 11, 2155–2159.
- [MHMC03] C. A. Maia, L. Hardouin, R. S. Mendes, and B. Cottenceau, *Optimal closed-loop control of timed event graphs in dioids*, IEEE Transactions on Automatic Control **48** (2003), no. 12, 2284–2287.



- [MHSMC05] C. A. Maia, L. Hardouin, R. Santos-Mendes, and B. Cottenceau, *On the model reference control for max-plus linear systems*, Proceedings of the 44th IEEE Conference on Decision and Control, and the European Control Conference 2005 (2005).
- [MMLH05] C. A. Maia, R. S. Mendes, R. Luders, and L. Hardouin, *Estratégias de controle por modelo de referência de sistemas a eventos discretos max-plus lineares*, Revista Controle e Automação (2005).
- [Mur89] T. Murata, *Petri nets: Properties, analysis and applications*, Proc. of the IEEE **77** (1989), no. 4, 541–580.
- [MYLA94] J. Moody, K. Yamalidou, M. Lemmon, and P. Antsaklis, *Feedback control of petri nets based on place invariants*, Proceedings of the 33rd IEEE Conference on Decision and Control, vol. 3, dec 1994, pp. 3104–3109.
- [OS08] A. L. P. OLIVEIRA and M. A. P. SILVEIRA, *Uma abordagem metodológica para o desenvolvimento de sistemas automatizados e integrados de manufatura*, Production - SciELO Brasil **7** (2008), no. 3, 8–25.
- [RW89] P. J. G. Ramadge and W. M. Wonham, *The control of discrete event systems*, Proceedings of the IEEE **77** (1989), no. 1, 81–98.
- [SE14] A. Seleim and H. ElMaraghy, *Max-plus modeling of manufacturing flow lines*, Proceedings of the 47th CIRP Conference on Manufacturing Systems **17** (2014), 71–75.
- [SvdB00] B. De Schutter and T. van den Boom, *Model predictive control for max-plus-linear systems*, Proceedings of the American Control Conference, vol. 6, 2000, pp. 4046–4050.
- [Tri03] G. C. Trivelato, *Técnicas de modelagem e simulação de sistemas dinâmicos*, INPE (2003).
- [vEO98] R.J. van Egmond and G.J. Olsder, *The (max,+) algebra applied to synchronization of traffic light processes*, WODES (1998), 451–456.
- [Wan98] J. Wang, *Timed petri nets*, Kluwer Academic Publishers, 1998.
- [ZSX04] Q. Zhu, W. Sheng, and N. Xi, *Max-plus algebra model for on-line task scheduling of a reconfigurable manufacturing work-cell*, Proc. IEEE/RSJ International Conf. on Intelligent Robots and Systems, sept-oct 2004, pp. (2): 1245–1250.