



Renato Markele Ferreira Cândido

FILTROS DE PARTÍCULAS  
APLICADOS A SISTEMAS MAX PLUS

Campinas  
2013





UNIVERSIDADE ESTADUAL DE CAMPINAS  
FACULDADE DE ENGENHARIA ELÉTRICA E DE COMPUTAÇÃO

Renato Markele Ferreira Cândido

FILTROS DE PARTÍCULAS  
APLICADOS A SISTEMAS MAX PLUS

Orientador: Prof. Dr. Rafael Santos Mendes

Dissertação de Mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica da Faculdade de Engenharia Elétrica e de Computação da Universidade Estadual de Campinas para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Automação.

ESTE EXEMPLAR CORRESPONDE À VERSÃO FINAL DA DISSERTAÇÃO DEFENDIDA PELO ALUNO RENATO MARKELE FERREIRA CÂNDIDO E ORIENTADA PELO PROF. DR. RAFAEL SANTOS MENDES.

---

Campinas  
2013

Ficha catalográfica  
Universidade Estadual de Campinas  
Biblioteca da Área de Engenharia e Arquitetura  
Rose Meire da Silva - CRB 8/5974

C161f Cândia, Renato Markele Ferreira, 1988-  
Filtros de partículas aplicados a sistemas max plus / Renato Markele Ferreira  
Cândia. – Campinas, SP : [s.n.], 2013.

Orientador: Rafael Santos Mendes.  
Dissertação (mestrado) – Universidade Estadual de Campinas, Faculdade de  
Engenharia Elétrica e de Computação.

1. Álgebra abstrata. 2. Processo estocástico. 3. Redes de petri. I. Mendes,  
Rafael Santos, 1957-. II. Universidade Estadual de Campinas. Faculdade de  
Engenharia Elétrica e de Computação. III. Título.

Informações para Biblioteca Digital

**Título em outro idioma:** Particle filters for max plus systems

**Palavras-chave em inglês:**

Abstract algebra

Stochastic process

Petri nets

**Área de concentração:** Automação

**Titulação:** Mestre em Engenharia Elétrica

**Banca examinadora:**

Rafael Santos Mendes [Orientador]

Ricardo Lüders

Romis Ribeiro de Faissol Attux

**Data de defesa:** 24-10-2013

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

## COMISSÃO JULGADORA - TESE DE MESTRADO

**Candidato:** Renato Markele Ferreira Cândido

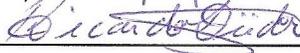
**Data da Defesa:** 24 de outubro de 2013

**Título da Tese:** "Filtros de Partículas Aplicados a Sistemas Max Plus"

Prof. Dr. Rafael Santos Mendes (Presidente):



Prof. Dr. Ricardo Luders:



Prof. Dr. Romis Ribeiro de Faissol Attux:





AOS MEUS PAIS, GERALDO EDÉSIO  
CÂNDIDO E MARIA DA LUZ FER-  
REIRA CÂNDIDO.



# Agradecimentos

Agradeço,

ao Prof. Rafael Santos Mendes, o apoio e oportunidade dada para o desenvolvimento desta pesquisa.

aos professores da FEEC: Celso de Almeida, Renato Baldini Filho, Akebo Yamakami, Wagner Caradori do Amaral e João Bosco Ribeiro do Val os ótimos cursos oferecidos.

a todos os funcionários da FEEC.

à agência CAPES o apoio financeiro concedido.

aos meus pais, cujo sacrifício me permitiu chegar até aqui, às minhas irmãs Cristiane e Camila e à minha namorada Isabela o companheirismo e suporte durante esta jornada.

a Deus por ter iluminado meu caminho até aqui.

aos meus colegas de moradia: Victor, Hugo, Lucas, Athos e Paulo.



Se uma inteligência, em determinado instante, pudesse conhecer todas as forças que governam o mundo natural e as posições de cada ser que o compõem; se, além disso, essa inteligência fosse suficientemente grande para submeter essas informações a análise, teria como abranger em uma única fórmula os movimentos dos maiores corpos do universo e os dos menores átomos. Para essa inteligência, nada seria incerto, e o futuro quanto o passado, se faria presente diante de seus olhos.

Pierre-Simon de Laplace



# Resumo

A principal contribuição desta dissertação é a proposta de algoritmos de filtragem por partículas em sistemas a eventos discretos nos quais predominam os problemas de sincronização. Esta classe de sistemas pode ser descrita por meio de equações lineares em uma álgebra não convencional usualmente conhecida como álgebra Max Plus. Os Filtros de Partículas são algoritmos Bayesianos sub-ótimos que realizam uma amostragem sequencial de Monte Carlo para construir uma aproximação discreta da densidade de probabilidade dos estados baseada em um conjunto de partículas com pesos associados. É apresentada uma revisão de sistemas a eventos discretos, de filtragem não linear e de filtros de partículas de um modo geral. Após apresentar esta base teórica, são propostos dois algoritmos de filtros de partículas aplicados a sistemas Max Plus. Em seguida algumas simulações foram apresentadas e os resultados apresentados mostraram a eficiência dos filtros desenvolvidos.

Palavras-chave: Filtragem Estocástica, Sistemas a Eventos Discretos, Álgebra Max-plus, Filtros de Partículas.



# Abstract

This thesis proposes, as its main contribution, particle filtering algorithms for discrete event systems in which synchronization phenomena are prevalent. This class of systems can be described by linear equation systems in a nonconventional algebra commonly known as Max Plus algebra. Particle Filters are suboptimal Bayesian algorithms that perform a sequential Monte Carlo sampling to construct a discrete approximation of the probability density of states based on a set of particles with associated weights. It is presented a review of discrete event systems, nonlinear filtering and particle filters. After presenting this theoretical background, two particle filtering algorithms applied to Max Plus systems are proposed. Finally some simulation results are presented, confirming the accuracy of the designed filters.

Keywords: Stochastic filtering, Discrete Event Systems, Max-Plus algebra, Particle Filters.



# Lista de Figuras

1.1	(a) Exemplo de rede de Petri e (b) Estado da Rede de Petri após o disparo da transição $t_1$ . . . . .	7
1.2	Grafo de Eventos Temporizado . . . . .	8
1.3	controlador em malha fechada . . . . .	21
1.4	Diagrama do observador . . . . .	21
2.1	Etapas do processo de filtragem por partículas . . . . .	28
2.2	Esquema do processo de reamostragem . . . . .	30
3.1	(a) Partículas geradas a partir de (3.5) e a região de verossimilhança não nula (b) Ampliação da região próxima ao suporte de $p(\mathbf{z}_k \mathbf{x}_k^i)$ . . . . .	38
3.2	Densidade $h(x)$ com componentes impulsivas e uniformes . . . . .	52
4.1	Sistema de Manufatura (flow-shop) . . . . .	64
4.2	Simulação 1 - Sequência de disparos e erro de estimação para os estados $x_2$ e $x_5$ . . . . .	67
4.3	Simulação 2 - Sequência de disparos e erro de estimação para os estados $x_2$ e $x_5$ . . . . .	69
4.4	Simulação 3 - Sequências de disparo dos estados $x_1$ a $x_4$ . . . . .	72
4.5	Simulação 3 - Sequências de disparo dos estados $x_5$ a $x_9$ . . . . .	73
4.6	Simulação 3 - Variância dos pesos e tamanho efetivo das amostras . . . . .	74
4.7	Simulação 4 - Sequências de disparo dos estados $x_1$ a $x_4$ . . . . .	76
4.8	Simulação 4 - Sequências de disparo dos estados $x_5$ a $x_9$ . . . . .	77
4.9	Um GET com duas entradas e uma saída . . . . .	78
4.10	Simulação 5 - Sequências de disparo das transições de entrada . . . . .	79
4.11	Simulação 5 - Sequências de disparo dos estados $x_2$ , $x_3$ e $x_4$ . . . . .	80
4.12	Simulação 6 - Sequências de disparo das transições de entrada . . . . .	81
4.13	Simulação 6 - Sequências de disparo dos estados $x_2$ , $x_3$ e $x_4$ . . . . .	82
4.14	Simulação 7 - Sequências de disparo dos estados $x_2$ , $x_3$ e $x_4$ . . . . .	84

4.15 Simulação 8 - Sequências de disparo dos estados  $x_2$ ,  $x_3$  e  $x_4$  . . . . . 86

# Lista de Tabelas

1.1	Exemplos de Diódes	12
4.1	Simulação 1 - Média e variância do erro quadrático médio após dez simulações	66
4.2	Simulação 2 - Média e variância do erro quadrático médio após dez simulações	69
4.3	Simulação 3 - Comparando os Filtros FP1 e FP2	72
4.4	Simulação 4 - Comparando os Filtros FP1 e FP2	76
4.5	Simulação 5 - Comparando os Filtros FP1 e FP2	79
4.6	Comparando os Filtros FP1 e FP2	82
4.7	Comparando os Filtros FP1 e FP2	83
4.8	Simulação 8 - Comparando os Filtros FP1 e FP2	85



# Sumário

<b>Introdução</b>	<b>1</b>
<b>1 Sistemas Max-Plus</b>	<b>5</b>
1.1 Modelagem de SED . . . . .	5
1.1.1 Redes de Petri . . . . .	5
1.1.2 Modelagem Max-Plus . . . . .	8
1.2 Álgebra Max-Plus . . . . .	10
1.2.1 Teoria geral de dióides . . . . .	10
1.2.2 Sistemas de Equações Lineares em $\overline{\mathbb{R}}_{max}$ . . . . .	15
1.2.3 Funções de Transferência . . . . .	18
1.3 Controle de Sistemas Max-Plus . . . . .	19
1.3.1 Problema de controle ótimo em malha aberta . . . . .	19
1.3.2 Pré-compensação com modelo de referência . . . . .	20
1.3.3 Problemas de controle em malha fechada . . . . .	20
1.3.4 Observador Max Plus . . . . .	21
<b>2 Filtragem Não Linear</b>	<b>23</b>
2.1 Definição do problema de filtragem não linear . . . . .	23
2.2 Filtros de Partículas . . . . .	25
<b>3 Filtro de Partículas Aplicado a Sistemas Max-Plus</b>	<b>35</b>
3.1 Filtro de Partículas com Densidade de Importância Sub-Ótima . . . . .	35
3.2 Filtro de Partículas com Densidade de Importância Ótima . . . . .	39
3.2.1 Geração de Variáveis Aleatórias: Aceitação-Rejeição . . . . .	43
3.2.2 Função Majorante . . . . .	47
3.2.3 Partículas Inconsistentes . . . . .	54

3.2.4	Filtragem de Partículas geradas pelo Método de Aceitação Rejeição . . .	58
3.2.5	Caso Exponencial . . . . .	59
3.3	Contribuições . . . . .	60
<b>4</b>	<b>Resultados</b>	<b>63</b>
4.1	Simulação 1 . . . . .	64
4.2	Simulação 2 . . . . .	68
4.3	Simulação 3 . . . . .	70
4.4	Simulação 4 . . . . .	75
4.5	Simulação 5 . . . . .	78
4.6	Simulação 6 . . . . .	80
4.7	Simulação 7 . . . . .	82
4.8	Simulação 8 . . . . .	84
<b>5</b>	<b>Conclusão</b>	<b>87</b>
5.1	Trabalhos Futuros . . . . .	89
	<b>Bibliografia</b>	<b>90</b>
<b>A</b>	<b>Cálculo da função de verossimilhança</b>	<b>95</b>
<b>B</b>	<b>Demonstração do Método de Aceitação Rejeição</b>	<b>97</b>

# Introdução

Os Sistemas a Eventos Discretos (SEDs) constituem uma classe de sistemas caracterizada por apresentar espaço de estados discreto e dinâmica dirigida única e exclusivamente pela ocorrência de eventos (Cassandras & Lafortune 2007). O estudo destes sistemas é de fundamental importância em diversas áreas, como redes de computadores e sistemas de tráfego, assim como no planejamento de atividades de caráter logístico (Silva, Mendes, Hardouin, Maia & Cottenceau 2011).

Em geral, quando se deseja modelar um sistema com espaço de estados contínuo a primeira abordagem a ser considerada é modelar o sistema a partir de equações diferenciais. O mesmo não ocorre com os sistemas com espaço de estados discreto, ou seja, não há paradigma formal para os modelos com espaço de estados discreto com dinâmica dirigida pela ocorrência de eventos. A escolha de um modelo para estes sistemas depende de qual subclasse destes sistemas deve ser modelada e de quais são os objetivos de modelagem. Pode-se citar, por exemplo, os autômatos finitos, que são utilizados para modelar sistemas com espaço de estados discreto e finito quando não se está preocupado com o momento em que o sistema entrou em um determinado estado o com quanto tempo o sistema permaneceu neste estado. Deste modo, na modelagem por autômatos finitos, assume-se que o comportamento do sistema é descrito em termos da sequência de eventos sem se preocupar com seus tempos de ocorrência (Cassandras & Lafortune 2007).

Outra ferramenta para a modelagem de SEDs são as Redes de Petri, que conseguem abordar uma subclasse mais ampla de sistemas a eventos discretos em comparação com os autômatos finitos. A modelagem por Redes de Petri não exige que o espaço de estados seja finito, e a temporização surge de maneira natural nas Redes de Petri Temporizadas. Outra motivação para a utilização de modelos baseados nas Redes de Petri é a grande quantidade

de técnicas de análise desenvolvidas para estes modelos (Cassandras & Lafortune 2007).

A dinâmica de grande parte dos SEDs está sujeita aos problemas de conflito e/ou sincronização. Um conflito ocorre quando há a competição por recursos; já a sincronização pode ser observada quando a ocorrência de um evento está sujeita à ocorrência de dois ou mais eventos diferentes. Os sistemas de manufatura oferecem exemplos destes dois tipos de problemas (Maia, Lüders, Mendes & Hardouin 2005). Um conflito ocorre num sistema de manufatura quando, por exemplo, duas peças devem ser processadas na mesma máquina, devendo haver uma decisão sobre qual delas será processada em primeiro lugar. O problema de sincronização fica evidente, por exemplo, quando a montagem de uma peça depende de duas ou mais partes, cujo processamento por sua vez pode obedecer às mesmas restrições. Nesse caso, obviamente, a parte com processamento mais tardio determinará o início da montagem da peça final.

Os sistemas a eventos discretos abordados neste trabalho são sistemas nos quais os problemas de sincronização são predominantes. Os Grafos de Eventos Temporizado (GETs) são uma subclasse de Redes de Petri adequada para modelar este tipo de sistema. Uma característica importante dos GETs é que eles podem ser descritos em termos de equações lineares em uma álgebra não convencional usualmente denominada *Álgebra Max Plus*. Esta abordagem se baseia nas estruturas algébricas conhecidas como dióides ou semi-anéis idempotentes. Diversas instâncias desta estrutura são úteis para a modelagem de SEDs, sendo o exemplo mais comum a álgebra Max Plus propriamente dita na qual as operações de soma e produto são definidas como sendo, respectivamente, as operações de maximização e de soma da álgebra usual. Neste contexto, os estados do sistema correspondem às datas de ocorrência dos diversos tipos de eventos apresentados em função de uma variável interna correspondente a um contador de eventos. O desenvolvimento desta estrutura algébrica e a formalização de sua utilização na teoria de SEDs possibilitou o surgimento de uma teoria de controle específica para estes sistemas (Hardouin, Maia, Cottenceau & Lhommeau 2010). Em (Cohen, Gaubert & Quadrat 1999), por exemplo, é proposta uma solução de controle ótimo, em malha aberta, para o problema em que se deseja obter o melhor controle possível de maneira a garantir que a saída de um SED siga a uma trajetória de referência. Em (Lhommeau, Hardouin, Ferrier & Ouerghi 2005), (Maia, Hardouin, Santos-Mendes & Cottenceau 2005) e (Menguy, Boimond, Hardouin & Ferrier 2000), também são propostas estratégias de controle em malha aberta. Em (Cottenceau, Hardouin, Boimond & Ferrier 2001) e (Maia, Hardouin, Santos-Mendes & Cottenceau 2005), trata-se do problema de obter um controlador, em malha fechada, que altere o comportamento entrada-saída segundo um modelo pré-determinado, denominado modelo de referência. Outras propostas de controladores para SEDs em ma-

lha fechada podem ser encontradas em (Cottenceau et al. 2001), (Lhommeau, Hardouin & Cottenceau 2003) e (Maia, Hardouin, Santos-Mendes & Cottenceau 2003). Em (De Schutter & van den Boom 2001, Necoara, De Schutter, van den Boom & Hellendoorn 2007), propõe-se um modelo de controle preditivo que minimiza um critério quadrático usando programação linear. Em (van den Boom & De Schutter 2004), esta mesma abordagem é estendida para um contexto estocástico.

Uma questão é de grande importância no desenvolvimento de controladores: Com que tipo de informação opera o controlador? Em aplicações onde se utiliza a realimentação de estados, observar ou estimar o estado torna-se essencial. De um modo geral, a observação ou estimação dos estados é de grande interesse, pois fornece informações sobre as propriedades internas do sistema que podem ser utilizadas em várias aplicações como detecção de falhas, diagnósticos, além de controladores com realimentação de estados. O problema de observabilidade em Redes de Petri Não Temporizadas é tratado em (Giua & Seatzu 2002); em (Loreto, Gaubert, Katz & Loiseau 2010), é apresentado um observador dinâmico para sistemas dinâmicos lineares Max Plus e em (Hardouin et al. 2010) é desenvolvido um observador ótimo para sistemas lineares Max Plus.

Neste trabalho, pretende-se tratar do problema de se obter estimativas para os estados de Sistemas a Eventos Discretos que podem ser modelados por GETs, num contexto ruidoso, no qual o estado não é diretamente medido, utilizando a abordagem Bayesiana. A ideia principal da abordagem Bayesiana para a estimação de estados dinâmicos é construir a densidade de probabilidade *a posteriori* dos estados baseando-se em todas as informações disponíveis, principalmente na sequência de medidas recebidas.

Entre os algoritmos ótimos para filtragem utilizando a abordagem Bayesiana, destaca-se o Filtro de Kalman. Este filtro é aplicável a sistemas lineares contínuos afetados por ruídos gaussianos aditivos. Para estes sistemas, o filtro de Kalman realiza uma estimação recursiva da densidade de probabilidade do estado do sistema, condicionada à sequência de medidas recebidas, a partir de uma estimativa para a média e para a variância desta densidade o que, para o caso Gaussiano, é suficiente para caracterizar completamente a densidade de probabilidade. Pode-se encontrar exemplos de aplicações práticas do Filtro de Kalman em diversos equipamentos como: GPS (Sistema de posicionamento global), Naves Espaciais e Satélites (Anderson & Moore 2005).

Existem outros filtros Bayesianos ótimos, como o Filtro de Massa Pontual (*Point-Mass Filter*), adequado a sistemas cujo espaço de estados é finito e discreto (Alspach & Sorenson 1972, Kramer & Sorenson 1988), e os filtros de Beneš (Beneš 1981) e Daum

(Daum 1986), aplicáveis a uma certa classe de sistemas não-lineares. No entanto, não se encontra na literatura nenhum filtro ótimo para sistemas a eventos discretos. Deste modo, foram considerados os *Filtros de Partículas*, que são filtros sub-ótimos que realizam uma amostragem sequencial de Monte Carlo para construir uma aproximação discreta da densidade de probabilidade dos estados baseada em um conjunto de partículas com pesos associados.

Basicamente, na filtragem por partículas, utiliza-se uma *densidade de importância* para construir um conjunto de partículas com pesos associados constituindo uma aproximação para a densidade de probabilidade dos estados do sistema condicionada à sequência de medidas recebidas. Uma das etapas mais importantes no projeto de um filtro de partículas é a escolha desta densidade de importância. Neste trabalho são propostas duas escolhas para a densidade de importância, sendo uma delas a escolha ótima, que considera tanto o estado anterior do sistema quanto a medida recebida, e a outra uma escolha sub-ótima definida pelo modelo de transição dos estados do sistema.

Este trabalho está organizado como se segue: No Capítulo 1 é apresentada uma revisão de *Redes de Petri e Sistemas Max-Plus*. No Capítulo 2 é feita uma breve discussão do problema de *Filtragem Não Linear* e em seguida a teoria de *Filtros de Partículas* é introduzida. No Capítulo 3 são desenvolvidos os algoritmos para filtragem por partículas aplicado a sistemas Max Plus. Alguns resultados são apresentados no Capítulo 4 e no Capítulo 5 são apresentadas algumas conclusões.

# Sistemas Max-Plus

Os Sistemas a Eventos Discretos (SEDs) são sistemas cujo espaço de estados é discreto sendo que a transição entre estados ocorre instantaneamente, de maneira assíncrona com o tempo e é dirigida pela ocorrência de eventos. A ocorrência de um evento pode ser resultado de alguma interferência externa, de algum fenômeno natural ou de uma série de condições que são subitamente satisfeitas (Cassandras & Lafortune 2007).

## 1.1 Modelagem de SED

### 1.1.1 Redes de Petri

As redes de Petri (RPs), desenvolvidas por C.A. Petri, constituem uma poderosa ferramenta para a modelagem de sistemas a eventos discretos. As redes de Petri conseguem representar, graficamente, grande parte das informações a respeito da dinâmica de um SED. Como será visto a seguir, essas redes conseguem explicitar, por exemplo, a função de transição do SED e as condições para que os eventos relacionados às transições possam ocorrer por meio de uma representação gráfica simples e intuitiva.

Pode-se definir uma Rede de Petri como um par  $(G, M)$  composto por um grafo direcionado bipartido  $G = (A, V)$  e uma marcação inicial  $M_0$ . O conjunto  $V$  representa os nós e é dividido em dois subconjuntos disjuntos  $P$  e  $T$ , cujos elementos recebem o nome de *lugares* e *transições* respectivamente. O conjunto  $A$  é um conjunto de arcos, sub-conjunto do conjunto

$(P \times T) \cup (T \times P)$ . Graficamente, cada lugar  $P_i, i = 1, \dots, |P|$  é representado por um círculo, cada transição  $T_j, j = 1, \dots, |T|$  é representada por uma barra e a marcação de cada lugar é indicada por pontos (ou fichas), ou seja,  $m_i$  fichas são colocadas em cada lugar  $P_i$  para indicar sua marcação. A marcação de uma Rede de Petri pode ser representada por um vetor  $x = [m_1, \dots, m_{|P|}]$  (Bacelli 1992).

A cada transição  $t_j$  pode-se associar um conjunto de lugares de entrada  $I(t_j)$  e um conjunto de lugares de saída  $O(t_j)$ .

**Definição 1.1.** Um lugar  $p_i$  é um lugar de entrada de  $t_j$  se existe algum arco de  $p_j$  para  $t_j$ , deste modo,  $I(t_j) = \{p_i : (p_i, t_j) \in A\}$

**Definição 1.2.** Um lugar  $p_i$  é um lugar de saída de  $t_j$  se existe algum arco de  $t_j$  para  $p_i$ , ou seja,  $O(t_j) = \{p_i : (t_j, p_i) \in A\}$ .

Os conjuntos de transições de saída  $O(p_i)$  e de entrada  $I(p_i)$  de um lugar são definidos de maneira análoga.

A dinâmica das RP é a seguinte:

- Uma transição  $t_j$  é dita *habilitada* se cada lugar pertencente ao conjunto  $I(t_j)$  possui pelo menos uma ficha disponível;
- O disparo de uma transição *habilitada* consome uma ficha de cada lugar pertencente ao conjunto  $I(t_j)$  e adiciona uma ficha a cada lugar pertencente ao conjunto  $O(t_j)$ .

Transições que não possuem lugares de entrada são denominadas entradas do sistema e o disparo destas transições é feito por decisões externas. Por outro lado, transições que não possuem lugares de saída são denominadas saídas do sistema; no contexto de filtragem assume-se que apenas as transições de saída são observáveis. As demais transições são denominadas estados do sistema.

**Exemplo 1.1** (Rede de Petri autônoma). *O conjunto dos lugares de entrada da transição  $t_1$  da rede de Petri ilustrada na Figura 1.1(a) é:  $I(t_2) = \{p_1\}$ . Para esta mesma transição, conjunto dos lugares de saída é:  $O(t_2) = \{p_2, p_3\}$ .*

*A marcação inicial desta rede é  $M_0 = [2, 0, 0, 1]$ . Observa-se que a transição  $t_1$  encontra-se habilitada, o disparo desta transição modifica a marcação da rede para  $M_0 = [1, 1, 1, 1]$ . A Figura 1.1(b) mostra o novo estado da rede após o disparo da transição  $t_1$ .*

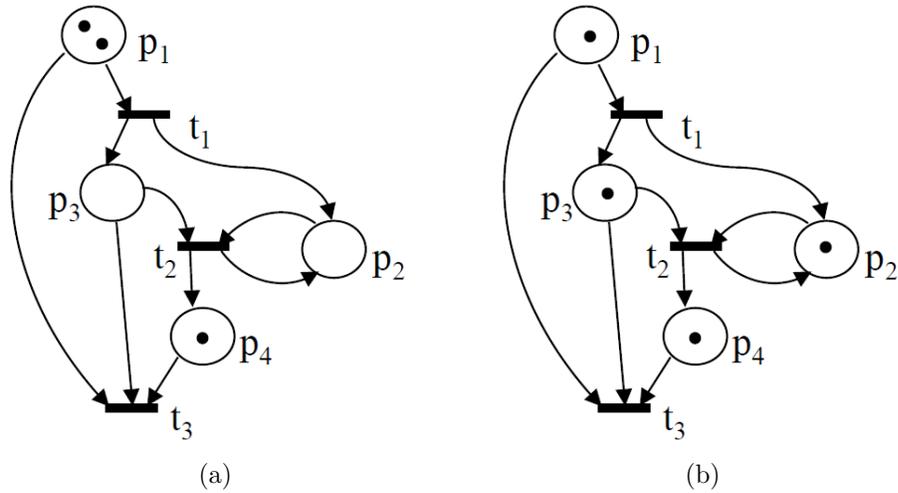


Figura 1.1: (a) Exemplo de rede de Petri e (b) Estado da Rede de Petri após o disparo da transição  $t_1$

As Redes de Petri não temporizadas oferecem informações sobre seqüências de eventos. No entanto, em situações em que se deseja saber, por exemplo, quando determinados eventos aconteceram, é necessário introduzir o tempo nas redes de Petri. A temporização pode ser feita de duas maneiras básicas: associando-se o tempo às transições (*tempo mínimo de disparo*) ou associando-se o tempo aos lugares (*tempo de permanência*). O *tempo mínimo de disparo* representa o tempo decorrido entre o início do disparo e o fim deste processo; este tempo pode ser usado para representar o tempo de produção de um determinado produto. O *tempo de permanência* representa o tempo mínimo que uma ficha deve permanecer em um lugar antes de ser tornar disponível para habilitar uma transição; este tempo pode ser usado para representar o tempo que uma ficha leva para ser transportada de uma transição inicial para o lugar de destino (Baccelli 1992).

Os Grafos de Eventos Temporizado (GET) são uma subclasse de Redes de Petri Temporizadas na qual todos os lugares apresentam exatamente uma transição de entrada e uma transição de saída. Deste modo, um GET não permite e nem é capaz modelar conflitos ou competições por recursos, já que as fichas presentes em um lugar são provenientes de apenas uma transição. Por outro lado, um GET é capaz de modelar problemas de sincronização, uma vez que uma transição pode ser precedida de vários lugares. A Figura 1.2 apresenta um GET *p-temporizado*, ou seja, um GET no qual o tempo é associado aos lugares.

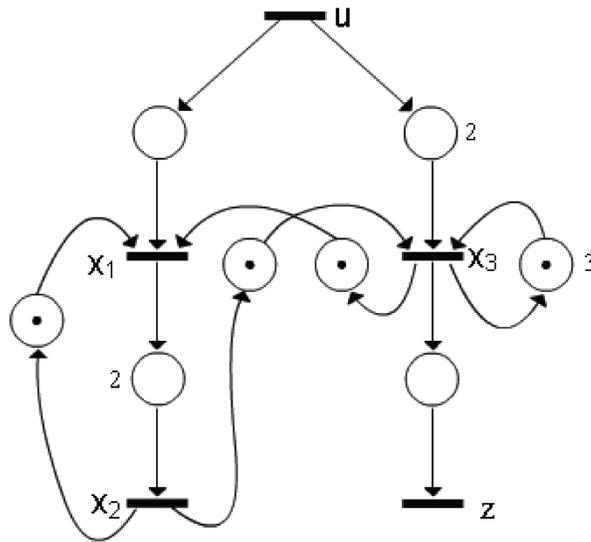


Figura 1.2: Grafo de Eventos Temporizado

### 1.1.2 Modelagem Max-Plus

Uma característica importante dos GETs é que eles podem ser descritos por sistemas de equações, similares às de sistemas dinâmicos lineares convencionais; mais especificamente, em termos de equações lineares recursivas em uma álgebra denominada **Álgebra Max Plus**.

A cada transição  $t_j$ , pode-se associar uma sequência não decrescente formada pelas variáveis  $x_j(k)$ ,  $k = 1, 2, \dots$ , denominadas **datadores**, contendo o  $k$  –ésimo instante de disparo da transição  $t_j$ . Assumindo-se que as sequências de disparo associadas às transições de entrada são conhecidas, é possível determinar as sequências de disparo de todas as transições do GET como mostra o exemplo a seguir.

**Exemplo 1.2.** Considerando o GET apresentado na Figura 1.2 as seguintes relações são verdadeiras:

$$\begin{aligned}
 x_1(k) &= \max\{x_2(k-1); x_3(k-1); u(k)\} \\
 x_2(k) &= 2 + x_1(k) \\
 x_3(k) &= \max\{x_2(k-1); 3 + x_3(k-1); 2 + u(k)\} \\
 z(k) &= x_3(k)
 \end{aligned}$$

Considerando agora o conjunto  $\mathbb{R} \cup \{-\infty\} \cup \{\infty\}$  e as operações:  $\text{sum} \equiv \oplus \equiv \max$  e

product  $\equiv \otimes \equiv +$ , obtém-se que:

$$\begin{aligned}x_1(k) &= x_2(k-1) \oplus x_3(k-1) \oplus u(k) \\x_2(k) &= 2 \otimes x_1(k) \\x_3(k) &= x_2(k-1) \oplus 3 \otimes x_3(k-1) \oplus 2 \otimes u(k) \\z(k) &= x_3(k)\end{aligned}$$

que, na forma matricial, fica:

$$X_k = A_0 \otimes X_k \oplus A_1 \otimes X_{k-1} \oplus B_0 \otimes U_k \quad (1.1a)$$

$$Z_k = C \otimes X_k \quad (1.1b)$$

onde:

$$\begin{aligned}A_0 &= \begin{bmatrix} -\infty & -\infty & -\infty \\ 2 & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix}; \quad A_1 = \begin{bmatrix} -\infty & 0 & 0 \\ -\infty & -\infty & -\infty \\ -\infty & 0 & 3 \end{bmatrix}; \quad B_0 = \begin{bmatrix} 0 \\ -\infty \\ 2 \end{bmatrix}; \\C &= \begin{bmatrix} -\infty & -\infty & 0 \end{bmatrix}; \quad X_k = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}; \quad U_k = \begin{bmatrix} u(k) \end{bmatrix} \quad e \quad Z_k = \begin{bmatrix} z(k) \end{bmatrix}\end{aligned}$$

Nota-se que (1.1a) ainda não está na forma recursiva pois a determinação de  $X_k$  depende do próprio  $X_k$  e, como não foi definida nenhuma operação inversa a  $\oplus$ , esta equação não pode ser resolvida pelas técnicas usuais. Nas próximas seções serão desenvolvidas as ferramentas necessárias para que se possa encontrar a forma recursiva de (1.1a), e no Exemplo 1.3 este procedimento será demonstrado.

Como será visto na próxima seção, o conjunto  $\mathbb{R} \cup \{-\infty\} \cup \{\infty\}$  munido das operações: soma  $\equiv \oplus \equiv \max$  e produto  $\equiv \otimes \equiv +$  é um **dióide** ao qual se dá o nome de **Max-Plus**, notado por  $\overline{\mathbb{R}}_{\max}$ .

## 1.2 Álgebra Max-Plus

Nesta seção, serão apresentados alguns conceitos da teoria de dióides (Cohen, Moller, Quadrat & Viot 1989, Baccelli 1992) e, em seguida, algumas ferramentas básicas para o estudo de sistemas de equações max-plus lineares.

### 1.2.1 Teoria geral de dióides

**Definição 1.3** (Dióides). *Um conjunto  $D$ , munido de duas operações internas soma  $\equiv \oplus$  e produto  $\equiv \otimes$  é um **dióide** ou **semi-anel idempotente** se os seguintes axiomas são verificados:*

**Axioma 1** (Associatividade).

$$\forall a, b, c \in D \begin{cases} (a \oplus b) \oplus c = a \oplus (b \oplus c) \\ (a \otimes b) \otimes c = a \otimes (b \otimes c) \end{cases}$$

**Axioma 2** (Comutatividade da adição).

$$\forall a, b \in D \quad a \oplus b = b \oplus a$$

**Axioma 3** (Distributividade).

$$\forall a, b, c \in D \begin{cases} (a \oplus b) \otimes c = (a \otimes c) \oplus (b \otimes c) \\ c \otimes (a \oplus b) = (c \otimes a) \oplus (c \otimes b) \end{cases}$$

**Axioma 4** (Existência do elemento nulo  $\epsilon$  e do elemento identidade  $e$ ).

$$\exists \epsilon \in D : \forall a \in D, a \oplus \epsilon = a$$

$$\exists e \in D : \forall a \in D, a \otimes e = a$$

**Axioma 5** (Absorção pelo elemento nulo).

$$\forall a \in D, a \otimes \epsilon = \epsilon \otimes a = \epsilon$$

**Axioma 6** (Idempotência da adição).

$$\forall a \in D, a \oplus a = a$$

**Definição 1.4.** Um dióide é comutativo se sua multiplicação for comutativa.

**Definição 1.5** (A estrutura algébrica  $\overline{\mathbb{R}}_{max}$ ). O conjunto  $\mathbb{R} \cup \{-\infty\} \cup \{\infty\}$  munido das operações: soma  $\equiv \oplus \equiv \max$  e produto  $\equiv \otimes \equiv +$  e com a convenção de que  $-\infty + \infty = -\infty$  é notado por  $\mathbb{R}_{max}$ .

**Teorema 1.1.** A estrutura  $\overline{\mathbb{R}}_{max}$  é um dióide comutativo.

**Prova:**

Para provar este teorema, será demonstrado que a estrutura  $\overline{\mathbb{R}}_{max}$  atende aos Axiomas 1-6 e que a sua multiplicação é comutativa.

1. Associatividade:

$$(a \oplus b) \oplus c = \max\{\max\{a, b\}, c\} = \max\{a, \max\{b, c\}\} = a \oplus (b \oplus c)$$

$$(a \otimes b) \otimes c = (a + b) + c = a + (b + c) = a \otimes (b \otimes c)$$

2. Comutatividade da adição e multiplicação:

$$(a \oplus b) = \max\{a, b\} = \max\{b, a\} = (b \oplus a)$$

$$(a \otimes b) = (a + b) = (b + a) = (b \otimes a)$$

3. Distributividade:

$$(a \oplus b) \otimes c = \max\{a, b\} + c = \max\{a + c, b + c\} = (a \otimes c) \oplus (b \otimes c)$$

$$c \otimes (a \oplus b) = c + \max\{a, b\} = \max\{c + a, c + b\} = (c \otimes a) \oplus (c \otimes b)$$

4. Existência do elemento nulo  $\epsilon$  e do elemento identidade  $e$ :

Definindo-se  $\epsilon \doteq -\infty$  e  $e \doteq 0$ , tem-se que:

$$a \oplus \epsilon = \max\{a, -\infty\} = a$$

$$a \otimes e = a + 0 = a$$

5. Absorção pelo elemento nulo:

$$a \otimes \epsilon = a + (-\infty) = -\infty = \epsilon$$

6. Idempotência da adição:

$$a \oplus a = \max\{a, a\} = a$$

■

Comparando as propriedades dos operadores  $\oplus$  e  $\otimes$  da estrutura algébrica  $\overline{\mathbb{R}}_{max}$  com os operadores  $+$  e  $\times$  da álgebra clássica, pode-se notar que  $\mathbb{R}_{max}$  não possui simetria na adição, ou seja, não vale a propriedade: se  $a \neq \epsilon$ , então  $\nexists b \in D : a \oplus b = \epsilon$ . Por outro lado, na estrutura  $\overline{\mathbb{R}}_{max}$ , existe a propriedade da idempotência da adição, o que não se observa na álgebra clássica.

Na Tabela 1.1 (Quadrat 1999) pode-se encontrar alguns exemplos dióides e suas aplicações.

Tabela 1.1: Exemplos de Dióides

$D$	$\oplus$	$\otimes$	$\epsilon$	$e$	Aplicação	Notação
$\mathbb{R} \cup \{+\infty\}$	min	+	$+\infty$	0	caminho mais curto	$\mathbb{R}_{min}$
$\mathbb{R} \cup \{-\infty\} \cup \{+\infty\}$	min	+	$+\infty$	0	caminho mais curto	$\overline{\mathbb{R}}_{min}$
$\mathbb{R} \cup \{-\infty\}$	max	+	$-\infty$	0	caminho mais longo	$\mathbb{R}_{max}$
$\mathbb{R}^+ \cup \{-\infty\}$	max	min	0	$+\infty$	capacidade máxima	$\mathbb{R}_{max,min}^+$
$[0, 1]$	max	$\times$	0	1		
$\mathbb{R}^+$	max	$\times$	0	1		$\mathbb{R}_{max,\times}^+$
$\{0, 1\}$	$\cup$	$\cap$	0	1	lógica	$\mathbb{B}$

**Teorema 1.2.** Em um dióide  $D$ , a seguinte equivalência é verificada:

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

Esta equivalência define uma relação de ordenamento parcial em  $D$ , notada por  $\geq$ , dada por:

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

Esta relação, é compatível com a soma e com a multiplicação à direita e à esquerda, ou seja:

$$a \geq b \implies \begin{cases} a \oplus c \geq b \oplus c, \\ a \otimes c \geq b \otimes c, \\ c \otimes a \geq c \otimes b, \end{cases}$$

A demonstração do Teorema 1.2 pode ser encontrada em (Cohen 2001).

**Definição 1.6** (Dióides Completos). *Um dióide é completo se e somente se ele for fechado para somas infinitas e o Axioma 3 se estende para somas infinitas.*

Num dióide completo, existe um limitante superior (notado por  $\infty$ ) dado pela soma de todos os seus elementos, que satisfaz:  $a \oplus \infty = \infty$ .

**Definição 1.7** (Dióides Arquimedianos). *Um dióide  $D$  é arquimediano se e somente se:*

$$\forall a e b \in D \exists c e d \in D : ac \geq b e da \geq b$$

Em (Cohen et al. 1989) demonstra-se que para dióides arquimedianos completos a seguinte expressão é válida:

$$\forall a \in D, a \neq \epsilon \quad \infty \otimes a = a \otimes \infty = \infty, \quad (1.2)$$

sendo que  $\epsilon \otimes \infty = \epsilon$ .

Para um dióide completo, com elemento nulo  $\epsilon$ , pode-se construir uma nova operação interna (notada por  $\wedge$ ) representando o limitante inferior dos operandos. Esta operação é associativa, comutativa, idempotente e tem  $\infty$  como elemento neutro :  $\forall a \quad \infty \wedge a = a$ . Esta operação apresenta ainda uma propriedade denominada "lei da absorção", dada por:

$$\forall a, b \in D, a \wedge (a \oplus b) = a \oplus (a \wedge b).$$

A seguinte equivalência também pode ser verificada:

$$a \geq b \iff a = a \oplus b \iff b = a \wedge b.$$

A multiplicação não é distributiva em relação ao operador  $\wedge$ . No entanto, devido ao Axioma 3, pode-se afirmar que a multiplicação é isotônica e, portanto, pode-se verificar que (Cohen et al. 1989):

$$(a \wedge b)c \leq (ac) \wedge (bc)$$

$$c(a \wedge b) \leq (ca) \wedge (cb)$$

O operador  $\wedge$  não é necessariamente distributivo em relação a  $\otimes$ . Pode-se, entretanto, verificar as seguintes relações:

$$\forall a, b e c \in D \quad \begin{cases} (a \wedge b) \otimes c \leq (a \otimes c) \wedge (b \otimes c) \\ (a \wedge b) \oplus c \leq (a \wedge c) \oplus (b \wedge c) \end{cases}$$

Se  $C = \{a, b, c, \dots\}$ ,  $x = a \oplus b \oplus c \oplus \dots$ ,  $y = a \otimes b \otimes c \otimes \dots$  e  $z = a \wedge b \wedge c \wedge \dots$ , a seguinte notação é utilizada:

$$x = \bigoplus_{w \in C} w, \quad y = \bigotimes_{w \in C} w \quad e \quad z = \bigwedge_{w \in C} w.$$

Pode-se definir matrizes  $n \times n$  cujos elementos pertençam a um dióide  $D$ . As operações de soma e produto para estas matrizes são definidas por:

$$[A \oplus B]_{ij} = [A]_{ij} \oplus [B]_{ij} \quad (\text{soma}) \quad (1.3)$$

$$[A \otimes B]_{ij} = \bigoplus_{k=1}^n [A]_{ik} \otimes [B]_{kj} \quad (\text{produto}) \quad (1.4)$$

O conjunto de matrizes munido das operações definidas por (1.3) e (1.4) formam um dióide notado por  $D^{n \times n}$  com as seguintes características:

- Se  $D$  é comutativo,  $D^{n \times n}$  não é necessariamente;
- Se  $D$  é arquimediano,  $D^{n \times n}$  não é necessariamente;
- Se  $D$  é completo,  $D^{n \times n}$  é completo.

Para um dióide  $D^{n \times n}$  pode-se definir ainda as seguintes relações:

$$A \geq B \Leftrightarrow a_{ij} \geq b_{ij} \quad \forall i, j$$

$$[A \wedge B]_{ij} = a_{ij} \wedge b_{ij}$$

## 1.2.2 Sistemas de Equações Lineares em $\overline{\mathbb{R}}_{max}$

Existem duas classes de sistemas lineares em  $\overline{\mathbb{R}}_{max}$  para os quais existem soluções satisfatórias (Baccelli 1992):

$$Ax = b \tag{1.5}$$

$$x = Ax \oplus b \tag{1.6}$$

### SOLUÇÃO DE $Ax = b$

**Definição 1.8.** *Uma Subsolução para (1.5) é dado por  $x$  que satisfaz:  $Ax \leq b$ .*

**Definição 1.9** (Residuação). *A residuação (à esquerda) de  $b$  por  $A$ , notada por  $A \bowtie b$ , é definida como a maior subsolução de (1.5).*

Analogamente, a residuação (à direita) de  $b$  por  $A$ , notada por  $b \phi A$ , é definida como a maior subsolução de  $xA = b$ .

**Teorema 1.3.** *A maior subsolução de (1.5), resultante da residuação à esquerda de  $b$  por  $A$ , é dada por:*

$$x_j = \bigwedge_i (A_{ij} \bowtie b_i),$$

onde  $A_{ij} \bowtie b_i$  representa  $b_i - A_{ij}$  na álgebra usual (Cohen et al. 1989).

**Prova:**

Tem-se que :

$$\begin{aligned}
 Ax \leq b &\iff \left\{ \bigoplus_j A_{ij}x_j \leq b_i, \quad \forall i \right\} \\
 &\iff \{x_j \leq b_i - A_{ij}, \quad \forall i, j\} \\
 &\iff \left\{ x_j \leq \bigwedge_i (A_{ij} \dot{\vee} b_i), \quad \forall j \right\}
 \end{aligned}$$

■

Deve-se notar que  $x = A \dot{\vee} b$  é a maior subsolução de (1.5), no entanto, nem sempre a igualdade é satisfeita.

Pode ser demonstrado que a operação de residuação obedece às seguintes relações:

$$a(a \dot{\vee} b) \leq b \quad (1.7)$$

$$a \dot{\vee} a \geq e \quad (1.8)$$

$$a(a \dot{\vee} a) = a \quad (1.9)$$

$$e \dot{\vee} a = a \quad (1.10)$$

$$\epsilon \dot{\vee} a = \infty \quad (1.11)$$

$$(a \dot{\vee} b)c \leq a \dot{\vee} (bc) \quad (1.12)$$

$$a \dot{\vee} (b \dot{\vee} c) = (ba) \dot{\vee} c \quad (1.13)$$

$$(a \dot{\vee} b) \oplus (a \dot{\vee} c) \leq a \dot{\vee} (b \oplus c) \quad (1.14)$$

$$(a \dot{\vee} b) \oplus (c \dot{\vee} b) \leq (a \wedge c) \dot{\vee} b \quad (1.15)$$

$$(a \dot{\vee} b) \wedge (c \dot{\vee} b) = (a \oplus c) \dot{\vee} b \quad (1.16)$$

$$(a \dot{\vee} b) \wedge (a \dot{\vee} c) = a \dot{\vee} (b \wedge c) \quad (1.17)$$

## SOLUÇÃO DE $x = Ax \oplus b$

**Teorema 1.4.** Se  $A^k \rightarrow \epsilon$  quando  $k \rightarrow \infty$ , a solução de  $x = Ax \oplus b$  é única, e dada por  $x = A^*b$ , sendo  $A^*$  definido como:

$$A^* \doteq A^0 \oplus A^1 \oplus \dots \oplus A^{n-1} \oplus A^n \oplus \dots, \quad (1.18)$$

**Prova:**

Se  $x$  é uma solução de  $x = Ax \oplus b$ , então  $x$  satisfaz:

$$x = A(Ax \oplus b) \oplus b = A^2x \oplus (A^1 \oplus A^0)b$$

$$x = A(A^2x \oplus Ab \oplus b) \oplus b = A^3x \oplus (A^2 \oplus A^1 \oplus A^0)b \cdots \quad x = A^kx \oplus (A^{k-1} \oplus A^{k-2} \oplus \dots \oplus A^0)b$$

No entanto, se  $A^k \rightarrow 0$  quando  $k \rightarrow \infty$ , utilizando um  $k$  suficientemente grande obtém-se que  $A^* = A^0 \oplus A^1 \oplus \dots \oplus A^{k-1}$ . Sendo assim,  $x = A^*b$  ■

Agora que as ferramentas necessárias foram apresentadas, pode-se colocar o sistema obtido no Exemplo 1.2 na sua forma recursiva. Este procedimento é demonstrado no exemplo a seguir:

**Exemplo 1.3.** No Exemplo 1.2 obteve-se o seguinte sistema de equações:

$$X_k = A_0 \otimes X_k \oplus A_1 \otimes X_{k-1} \oplus B_0 \otimes U_k \quad (1.19a)$$

$$Z_k = C \otimes X_k \quad (1.19b)$$

onde:

$$A_0 = \begin{bmatrix} -\infty & -\infty & -\infty \\ 2 & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix}; \quad A_1 = \begin{bmatrix} -\infty & 0 & 0 \\ -\infty & -\infty & -\infty \\ -\infty & 0 & 3 \end{bmatrix}; \quad B_0 = \begin{bmatrix} 0 \\ -\infty \\ 2 \end{bmatrix};$$

$$C = \begin{bmatrix} -\infty & -\infty & 0 \end{bmatrix}; \quad X_k = \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix}; \quad U_k = [u(k)] \quad e \quad Z_k = [z(k)]$$

Definindo-se  $b \doteq A_1 \otimes X_{k-1} \oplus B_0 \otimes U_k$ , 1.19a pode ser reescrita como:

$$X_k = A_0 X_k \oplus b$$

Como visto, a solução para esta equação é dada por:

$$\begin{aligned} X_k &= A_0^* b \\ &= A_0^* (A_1 \otimes X_{k-1} \oplus B_0 \otimes U_k) \\ &= A X_{k-1} \oplus B U_k \end{aligned}$$

onde:

$$A_0^* = \begin{bmatrix} 0 & -\infty & -\infty \\ -\infty & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix} \oplus \begin{bmatrix} -\infty & -\infty & -\infty \\ 2 & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix} \oplus \begin{bmatrix} -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \\ -\infty & -\infty & -\infty \end{bmatrix} = \begin{bmatrix} 0 & -\infty & -\infty \\ 2 & 0 & -\infty \\ -\infty & -\infty & 0 \end{bmatrix},$$

$$A = A_0^* \otimes A_1 = \begin{bmatrix} -\infty & 0 & 0 \\ -\infty & 2 & 2 \\ -\infty & 0 & 3 \end{bmatrix} \quad e \quad B = A_0^* \otimes B_0 = \begin{bmatrix} 0 \\ 2 \\ 2 \end{bmatrix}$$

Observa-se portanto que, adotando o procedimento utilizado no exemplo anterior, é possível descrever todas as transições de um GET por meio de equações lineares recursivas. Deste modo, o GET pode ser descrito por um sistema de equações lineares do tipo:

$$\mathbf{x}(k) = A\mathbf{x}(k-1) \oplus B\mathbf{u}(k) \quad (1.20)$$

$$\mathbf{z}(k) = C\mathbf{x}(k) \quad (1.21)$$

### 1.2.3 Funções de Transferência

Em sistemas max-plus, pode-se definir a transformada  $\gamma$ , similar à transformada  $z$  dos sistemas de tempo discreto, na qual  $\gamma$  opera como  $z^{-1}$  (Cohen et al. 1989, Baccelli 1992). Deste modo, a transformada  $\gamma$  de  $x$  é dada por:

$$x(\gamma) = \bigoplus_{i=1}^{\infty} x(i)\gamma^i \quad (1.22)$$

É imediato constatar que:

$$z(k) = x(k-1) \iff z(\gamma) = \gamma x(\gamma) \quad (1.23)$$

Sendo assim, aplicando a transformada  $\gamma$  no sistema descrito pelas Equações (1.20) e

(1.21) obtém-se:

$$\mathbf{x}(\gamma) = \gamma A \mathbf{x}(\gamma) \oplus B \mathbf{u}(\gamma) \quad (1.24)$$

$$\mathbf{z}(\gamma) = C \mathbf{x}(\gamma) \quad (1.25)$$

Definindo-se  $A(\gamma) \doteq \gamma A$ ,  $B(\gamma) \doteq B$  e  $C(\gamma) \doteq C$ , a Equação (1.24) pode ser reescrita como:

$$\mathbf{x}(\gamma) = A(\gamma) \mathbf{x}(\gamma) \oplus B(\gamma) \mathbf{u}(\gamma) \quad (1.26)$$

Relembrando que a solução de  $x = Ax \oplus b$  é  $x = A^*b$ , a solução de (1.26) é dada por:

$$\mathbf{x}(\gamma) = A^*(\gamma) B(\gamma) \mathbf{u}(\gamma) \quad (1.27)$$

Substituindo este resultado em (1.25), obtém-se:

$$\mathbf{z}(\gamma) = H(\gamma) \mathbf{u}(\gamma) \quad (1.28)$$

onde  $H(\gamma)$  é a função de transferência dada por:

$$H(\gamma) = C(\gamma) A^*(\gamma) B(\gamma) \quad (1.29)$$

## 1.3 Controle de Sistemas Max-Plus

### 1.3.1 Problema de controle ótimo em malha aberta

Considere o problema em que se deseja que a saída  $z$  do sistema seja tão próxima quanto possível de uma referência  $y$ , respeitando a condição  $z \leq y$ . Em (Cohen et al. 1989) demonstra-se que a entrada ótima que satisfaz esta restrição é dada por:

$$\mathbf{u}_{opt} = CA^*B \backslash y \quad (1.30)$$

Em um sistema de manufatura, por exemplo,  $y$  corresponde às demandas dos clientes,  $u$  a entrada de peças brutas no sistema e  $z$  a saída de peças processadas. Neste contexto,

o controle ótimo  $\mathbf{u}_{opt}$  é o que minimiza o estoque interno enquanto assegura que demanda é atendida.

### 1.3.2 Pré-compensação com modelo de referência

Considere agora o problema em que o comportamento desejado do sistema é dado por um modelo de referência  $G_{ref}$ . Seja  $H$  a função de transferência do sistema a ser controlado. A solução deste problema consiste na construção de um pré-compensador  $P$  para controlar as entradas de modo que a saída do sistema seja tão próxima quanto possível da que seria obtida aplicando estas mesmas entradas ao modelo de referência (Cottenceau 1999). Deste modo, a saída do sistema de referência é dada por  $z = G_{ref}v$ , onde  $v$  é o modelo de entrada. A saída do modelo a ser controlado é dada por  $y = Hu$ , onde  $u = Pv$  é a entrada filtrada, ou controlada, do sistema. O objetivo é, portanto, encontrar o maior valor de  $P$  que satisfaz:

$$HPv \leq G_{ref}v$$

Da teoria de residuação, encontra-se que o valor ótimo de  $P$  é dado por: (Ouerghi & Hardouin 2006)

$$P_{otm} = H \backslash G_{ref}. \quad (1.31)$$

### 1.3.3 Problemas de controle em malha fechada

Nos problemas de controle em malha fechada, as medidas da saída do sistemas são levadas em consideração na construção dos controladores. O objetivo dos controladores em malha fechada é modificar a dinâmica do sistema  $H$  usando um controlador  $F$  localizado entre a saída do sistema  $y$  e a entrada  $u$ , como pode ser observado na Figura 1.3. Estes controladores são escolhidos de maneira que o comportamento do sistema controlado seja o mais próximo possível de um modelo de referencia  $G_{ref}$  (Cottenceau 1999). Tem-se que, em malha fechada,

$$G_{mf}(F) = (HF)^*H.$$

Formalmente, o objetivo é calcular o maior controlador  $F$  tal que:

$$(HF)^*H \leq G_{ref}. \quad (1.32)$$

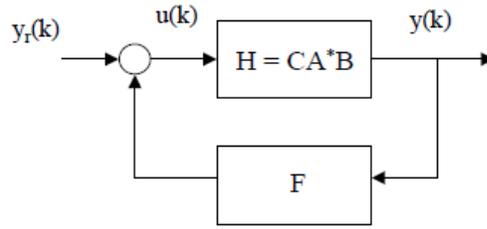


Figura 1.3: controlador em malha fechada

**Proposição 1.** (*Cottenceau et al. 2001*) Se  $G_{ref} = D^*H$  ou  $G_{ref} = HD^*$ , sendo  $D$  uma matriz racional de dimensões apropriadas, então existe um controlador máximo  $F_{max} = H \setminus G_{ref} \oslash H$ . Em particular, se  $G_{ref} = (HD)^*H$ , então  $G_{mf}(F_{max}) = G_{ref}$ .

### 1.3.4 Observador Max Plus

O objetivo do Observador Max Plus é utilizar a informação da saída de um sistema afetado por ruídos e aplicar em um simulador a fim de obter estimativas ( $\hat{x}$ ) para os estados do sistema ( $x$ ) por meio da matriz de observação  $L$ . Estas estimativas satisfazem  $\hat{x} \leq x$ . Na Figura 1.4, encontra-se um diagrama de blocos no qual a matriz  $L$  filtra a saída do sistema e fornece informações para o simulador.

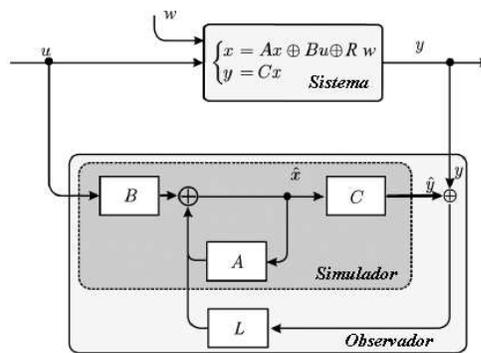


Figura 1.4: Diagrama do observador (Hardouin et al. 2010)

**Proposição 2** ((Hardouin et al. 2010)). A maior matriz de observação que satisfaz  $\hat{x} \leq x$  é dada por:

$$L_{opt} = (((A^*B) \oslash (CA^*B)) \wedge ((A^*R) \oslash (CA^*R))) \quad (1.33)$$

Nota-se o observador é baseado na teoria de residuação e é ótimo no sentido de que a matriz de observação  $L$  é a maior solução que satisfaz a restrição  $\hat{x} \leq x$ . Este fato permite que sejam obtidas estimativas menores ou iguais ao estado real. Deste modo, pode-se dizer que o Observador Max Plus fornece um limitante inferior para as estimativas dos estados.

## Filtragem Não Linear

### 2.1 Definição do problema de filtragem não linear

Na abordagem bayesiana para a estimação de estados de sistemas dinâmicos, deseja-se construir a densidade de probabilidade *a posteriori* dos estados aproveitando-se de todas as informações disponíveis, principalmente a sequência de medidas recebidas (Ristic, Arulampalam & Gordon 2004).

Para a definição do problema de filtragem não linear, considere o Modelo de Markov Oculto (do inglês: *Hidden Markov Model* - HMM) definido por:

$$\mathbf{x}_k = f_{k-1}(\mathbf{x}_{k-1}, \mathbf{w}_{k-1}), \quad (2.1)$$

$$\mathbf{z}_k = h_k(\mathbf{x}_k, \mathbf{v}_k), \quad (2.2)$$

onde  $\mathbf{x}_k \in \mathbb{R}^n$  e  $\mathbf{z}_k \in \mathbb{R}^p$  são os vetores de estados e de medidas, respectivamente, em um instante  $k$ ;  $f_{k-1} : \mathbb{R}^n \times \mathbb{R}^m \rightarrow \mathbb{R}^n$  é a função de transição do sistema;  $h_k : \mathbb{R}^n \times \mathbb{R}^r \rightarrow \mathbb{R}^p$  é a função de medida;  $\mathbf{w}_{k-1} \in \mathbb{R}^m$  e  $\mathbf{v}_k \in \mathbb{R}^r$  são sequências independentes que representam o ruído do processo e o ruído de medida, respectivamente.

Supõe-se que  $\mathbf{w}_{k-1}$  e  $\mathbf{v}_k$  possuem funções de densidade de probabilidade conhecidas. Supõe-se ainda que a densidade de probabilidade do estado inicial  $p(x_0) \equiv p(x_0|z_0)$  e as funções  $f_{k-1}$  e  $h_k$  sejam conhecidas.

Deseja-se portanto, construir a densidade de probabilidade dos estados  $\mathbf{x}_k$  baseando-se

em todas as informações disponíveis:  $p(\mathbf{x}_k|\mathbf{Z}_k)$ , onde  $\mathbf{Z}_k \equiv \{\mathbf{z}_i, i = 1, \dots, k\}$  é a sequência de todas as medidas disponíveis. A princípio, esta densidade pode ser obtida de maneira recursiva em dois estágios: predição e atualização (Gordon, Salmond & Smith 1993).

No estágio de previsão, assume-se que a densidade  $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$  seja conhecida no instante  $(k - 1)$ . Deste modo, é possível obter a densidade *a priori* dos estados  $\mathbf{x}_k$  no instante  $k$  usando a equação de Chapman-Kolmogorov (Arulampalam, Maskell, Gordon & Clapp 2002):

$$p(\mathbf{x}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})d\mathbf{x}_{k-1}. \quad (2.3)$$

O modelo probabilístico da evolução dos estados  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$  pode ser obtido por meio da equação (2.1).

No instante de tempo  $k$ , a medida  $\mathbf{z}_k$  torna-se disponível e pode ser utilizada para atualizar a densidade *a priori*  $p(\mathbf{x}_k|\mathbf{Z}_{k-1})$  por meio da regra de Bayes:

$$\begin{aligned} p(\mathbf{x}_k|\mathbf{Z}_k) &= p(\mathbf{x}_k|\mathbf{z}_k, \mathbf{Z}_{k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k, \mathbf{Z}_{k-1})p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}, \end{aligned} \quad (2.4)$$

onde

$$p(\mathbf{z}_k|\mathbf{Z}_{k-1}) = \int p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{Z}_{k-1})d\mathbf{x}_k \quad (2.5)$$

é a constante de normalização, a qual depende da função de verossimilhança  $p(\mathbf{z}_k|\mathbf{x}_k)$  que é definida a partir da equação (2.2) e da PDF de  $v_k$ .

As equações (2.3) e (2.4) constituem a base para a solução formal para o problema de estimação recursiva utilizando a abordagem Bayesiana (Gordon et al. 1993). Sendo assim, um algoritmo ótimo deve ser capaz de calcular a forma exata de  $p(\mathbf{x}_k|\mathbf{Z}_k)$  de maneira recursiva por meio de soluções analíticas para estas equações. As soluções ótimas conhecidas na literatura existem apenas para alguns casos restritos, dentre as quais destaca-se o Filtro de Kalman (Anderson & Moore 2005), que é aplicável somente a sistemas nos quais as funções  $f_k$  e  $h_k$  são lineares e  $\mathbf{w}_k$  e  $\mathbf{v}_k$  são ruídos gaussianos aditivos com matriz de covariância conhecida. Na maioria das situações práticas, as soluções analíticas para (2.4) e (2.5) são desconhecidas, o que inviabiliza a aplicação dos filtros não lineares ótimos a estes sistemas. Deste modo, soluções sub-ótimas devem ser consideradas.

## 2.2 Filtros de Partículas

Os Filtros de Partículas (FP) são algoritmos Bayesianos sub-ótimos para filtragem não linear. A maioria destes são baseados no método de amostragem sequencial de Monte Carlo (SMC) (Arulampalam et al. 2002). A ideia principal é obter estimativas para a densidade  $p(\mathbf{x}_k|\mathbf{Z}_k)$  a partir de um conjunto de  $N$  partículas  $\{\mathbf{x}_{k-1}^i, i = 1, \dots, N\}$ , que aproximam a densidade  $p(\mathbf{x}_{k-1}|\mathbf{Z}_{k-1})$  em  $k - 1$ . Deste modo, o filtro de partículas é um mecanismo de aproximação das relações (2.3) e (2.4) (Gordon et al. 1993).

A amostragem de importância é uma técnica na qual amostras obtidas a partir de uma densidade de importância  $q(\phi)$  são usadas para aproximar uma distribuição de interesse  $\pi(\phi)$  por meio de um processo de ponderação das amostras. A seguir, encontra-se uma breve descrição dessa técnica.

Sejam  $Y$  e  $X$  variáveis aleatórias multidimensionais relacionadas segundo a equação  $Y = f(X)$  sendo  $X \sim p(x)$ . A média de  $Y$  é dada por:

$$\bar{Y} = \int_{-\infty}^{\infty} f(x)p(x)dx \quad (2.6)$$

Supondo-se que seja possível obter  $N \gg 1$  amostras  $\{x^i, i = 1, \dots, N\}$  sorteadas segundo a densidade  $p(x)$ , uma estimativa para (2.6) é dada por:

$$\bar{Y}_N = \frac{1}{N} \sum_{i=1}^N f(x^i) \quad (2.7)$$

Admite-se agora uma situação na qual só é possível amostrar segundo uma densidade  $q(x)$  similar<sup>1</sup> a  $p(x)$ . Pode-se reescrever (2.6) da seguinte maneira:

$$\bar{Y} = \int_{-\infty}^{\infty} f(x) \frac{p(x)}{q(x)} q(x) dx \quad (2.8)$$

Deste modo, pode-se usar  $N$  amostras distribuídas segundo  $q(x)$  para estimar (2.8) por meio da seguinte equação:

$$\bar{Y}_n = \frac{1}{N} \sum_{i=1}^N f(x^i) \tilde{\omega}(x^i) \quad (2.9)$$

---

<sup>1</sup>Uma densidade de probabilidade  $q(x)$  é similar a  $p(x)$  se  $\forall x : p(x) > 0 \Rightarrow q(x) > 0$ .

onde

$$\tilde{\omega}(x^i) = \frac{p(x^i)}{q(x^i)} \quad (2.10)$$

Realizado-se a normalização dos pesos  $\tilde{\omega}(x^i)$  por meio de:

$$\omega(x^i) = \frac{\tilde{\omega}(x^i)}{\sum_{j=1}^N \tilde{\omega}(x^j)}$$

Tem-se que:

$$\bar{Y}_N = \sum_{i=1}^N f(x^i)\omega(x^i) \quad (2.11)$$

A partir do método de amostragem de importância, pode-se obter o algoritmo de amostragem sequencial de importância o qual é utilizado para realizar a filtragem não linear. Para desenvolver o algoritmo de amostragem sequencial de importância, considera-se que exista em  $k - 1$  um conjunto de partículas com pesos associados  $\{\mathbf{X}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N$ , onde  $\mathbf{X}_{k-1}^i = \{\mathbf{x}_j^i, j = 0, \dots, k - 1\}$  representa a trajetória da  $i$ -ésima partícula desde seu estado inicial  $\mathbf{x}_0^i$  até  $\mathbf{x}_{k-1}^i$ . Pode-se, portanto, construir a seguinte aproximação para a densidade conjunta *a posteriori* dos estados:

$$p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \cong \sum_{i=1}^N \omega_{k-1}^i \delta(\mathbf{X}_{k-1} - \mathbf{X}_{k-1}^i) \quad (2.12)$$

Se as amostras  $\mathbf{X}_{k-1}^i$  foram obtidas segundo uma densidade de importância  $q(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})$  então, de acordo com (2.10):

$$\omega_{k-1}^i \propto \frac{p(\mathbf{X}_{k-1}^i|\mathbf{Z}_{k-1})}{q(\mathbf{X}_{k-1}^i|\mathbf{Z}_{k-1})} \quad (2.13)$$

Admite-se que a densidade de importância  $q$  seja definida de tal maneira que:

$$q(\mathbf{X}_k|\mathbf{Z}_k) = q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)q(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}). \quad (2.14)$$

Deste modo, cada partícula  $\mathbf{X}_{k-1}^i$  pode ser expandida para  $\mathbf{X}_k^i$  usando o novo estado  $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}, \mathbf{z}_k)$ .

Para completar o procedimento de filtragem, deve-se atualizar cada peso  $\{\omega_{k-1}^i\}$  para

$\{\omega_k^i\}$  de modo que a seguinte aproximação seja verdadeira:

$$p(\mathbf{X}_k|\mathbf{Z}_k) \cong \sum_{i=1}^N \omega_k^i \delta(\mathbf{X}_k - \mathbf{X}_k^i). \quad (2.15)$$

onde:

$$\omega_k^i \propto \frac{p(\mathbf{X}_k^i|\mathbf{Z}_k)}{q(\mathbf{X}_k^i|\mathbf{Z}_k)} \quad (2.16)$$

Para obter-se a equação de atualização dos pesos deve-se expressar a densidade  $p(\mathbf{X}_k|\mathbf{Z}_k)$  em termos de  $p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})$ ,  $p(\mathbf{z}_k|\mathbf{x}_k)$  e  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$ .

Utilizando a regra de Bayes, tem-se que:

$$\begin{aligned} p(\mathbf{X}_k|\mathbf{Z}_k) &= p(\mathbf{X}_k|\mathbf{z}_k, \mathbf{Z}_{k-1}) \\ &= \frac{p(\mathbf{z}_k|\mathbf{X}_k, \mathbf{Z}_{k-1})p(\mathbf{X}_k|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \end{aligned} \quad (2.17)$$

Mas, por definição:

$$\begin{aligned} p(\mathbf{X}_k|\mathbf{Z}_{k-1}) &= p(\mathbf{x}_k, \mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \\ &= p(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Z}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \end{aligned} \quad (2.18)$$

Substituindo-se (2.18) em (2.17):

$$\begin{aligned} p(\mathbf{X}_k|\mathbf{Z}_k) &= \frac{p(\mathbf{z}_k|\mathbf{X}_k, \mathbf{Z}_{k-1})p(\mathbf{x}_k|\mathbf{X}_{k-1}, \mathbf{Z}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})} \\ &= \frac{p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})}{p(\mathbf{z}_k|\mathbf{Z}_{k-1})}p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \\ &\propto p(\mathbf{z}_k|\mathbf{x}_k)p(\mathbf{x}_k|\mathbf{x}_{k-1})p(\mathbf{X}_{k-1}|\mathbf{Z}_{k-1}) \end{aligned} \quad (2.19)$$

Finalmente, a equação de atualização dos pesos pode ser obtida substituindo-se (2.19) e (2.14) em (2.16):

$$\omega_k^i \propto \omega_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}, \quad (2.20)$$

Uma estimativa para  $\mathbf{x}_k$  pode ser calculada a partir de:

$$\hat{\mathbf{x}}_k = \sum_{i=1}^N \mathbf{x}_k^i \omega_k^i. \quad (2.21)$$

Em síntese, como pode ser observado na Figura 2.1, o filtro de partículas parte de um conjunto de  $N$  partículas com pesos associados  $\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N$ , formando uma aproximação para  $p(\mathbf{x}_{k-1}|\mathbf{z}_{k-1})$ , para obter um novo conjunto  $\{\mathbf{x}_k^i, \omega_k^i\}_{i=1}^N$ , que aproxima  $p(\mathbf{x}_k|\mathbf{z}_k)$ . Este processo é realizado em duas etapas: na etapa de propagação utiliza-se uma densidade de importância  $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$  para propagar cada partícula  $\mathbf{x}_{k-1}^i$  para  $\mathbf{x}_k^i$ ; na segunda etapa, os pesos  $\omega_{k-1}^i$  são atualizados para  $\omega_k^i$  a partir de (2.20).

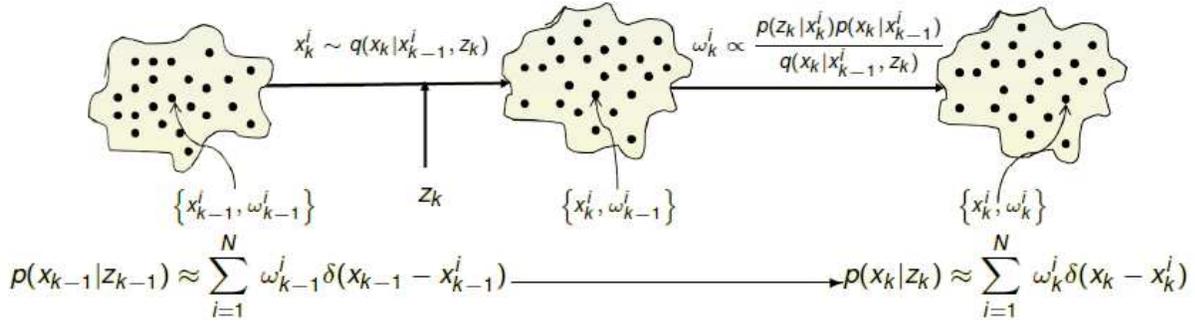


Figura 2.1: Etapas do processo de filtragem por partículas

Em (Doucet, Godsill & Andrieu 2000) demonstra-se que, para densidades de importância da forma (2.14), a variância dos pesos de importância cresce a cada iteração do algoritmo de filtragem. Este comportamento é prejudicial ao desempenho do filtro e leva ao fenômeno conhecido como *degeneração das partículas*, que ocorre quando após um certo número de iterações apenas algumas partículas têm pesos não desprezíveis. Na prática, este fenômeno causa uma situação na qual haverá um grande esforço computacional para atualizar partículas cuja contribuição para a aproximação de  $p(\mathbf{x}_k|\mathbf{z}_k)$  é desprezível.

O *tamanho efetivo da amostra* ( $N_{eff}$ ) pode ser usado como um indicador para o grau de degeneração do algoritmo (Kong, Liu & Wong 1994, Liu & Chen 1998). Este indicador é definido em termos da variância dos pesos de importância da seguinte maneira:

$$N_{eff} = \frac{N}{1 + N^2 var(\omega_k^i)}. \quad (2.22)$$

Uma aproximação de implementação computacional mais simples para (2.22) é dada por:

$$\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N \omega_k^i}. \quad (2.23)$$

Observa-se que  $1 \leq \hat{N}_{eff} \leq N$ , sendo que o limite superior  $N$  é atingido quando todas as partículas têm o mesmo peso  $1/N$ , e o limite inferior ocorre quando todas as partículas, exceto uma, têm peso nulo.

A degeneração ocorre devido ao fato de que o algoritmo de filtro de partículas por si só não apresenta qualquer mecanismo de *feedback* que leva em conta as observações para controlar a simulação (Gustafsson 2010). Uma alternativa para atenuar este fenômeno é a utilização de um processo de *reamostragem* no qual as partículas são clonadas com uma probabilidade proporcional ao seu peso. Deste modo, partículas com pesos baixos tendem a ser eliminadas e partículas com pesos relevantes tendem a ser clonadas. Este processo consiste em obter um novo conjunto de partículas  $\{\mathbf{x}_k^{i*}\}_{i=1}^N$  com pesos uniformes reamostrando-se  $N$  vezes a partir da representação discreta de  $p(\mathbf{x}_k|\mathbf{z}_k)$  dada por (2.24) de modo que  $P\{\mathbf{x}_k^{i*} = \mathbf{x}_k^j\} = \omega_k^j$  (Ristic et al. 2004).

$$p(\mathbf{x}_k|\mathbf{z}_k) = \sum_{i=1}^N \omega_k^i \delta(\mathbf{x}_k - \mathbf{x}_k^i) \quad (2.24)$$

Na Figura 2.2, encontra-se um esquema para o processo de reamostragem (van der Merwe, Doucet, de Freitas & Wan 2000). Nota-se que  $\mu_i \sim U[0, 1]$  é mapeada no índice  $j$ , correspondente à partícula  $\mathbf{x}_k^j$ . Nesta figura, torna-se evidente que as partículas com peso maior tem maiores chances de serem duplicadas.

Deste modo, o processo de reamostragem introduz no algoritmo de filtragem o mecanismo de *feedback* que utiliza a informação das observações.

Embora o processo de reamostragem diminua o efeito da degeneração, ele produz outros problemas práticos, como a perda da diversidade entre as partículas. Este problema é conhecido como *pauperização das amostras* e, em situações em que o ruído do sistema é pequeno, leva a uma situação em que todas as partículas irão colapsar para um único ponto em algumas iterações (Ristic et al. 2004). Sendo assim, é interessante que o processo de reamostragem seja executado apenas quando for realmente necessário. Existe um certo grau de liberdade quanto à escolha da estratégia de reamostragem, mas, geralmente, o processo de reamostragem é feito quando o grau de degeneração do algoritmo  $\hat{N}_{eff}$  encontra-se abaixo de

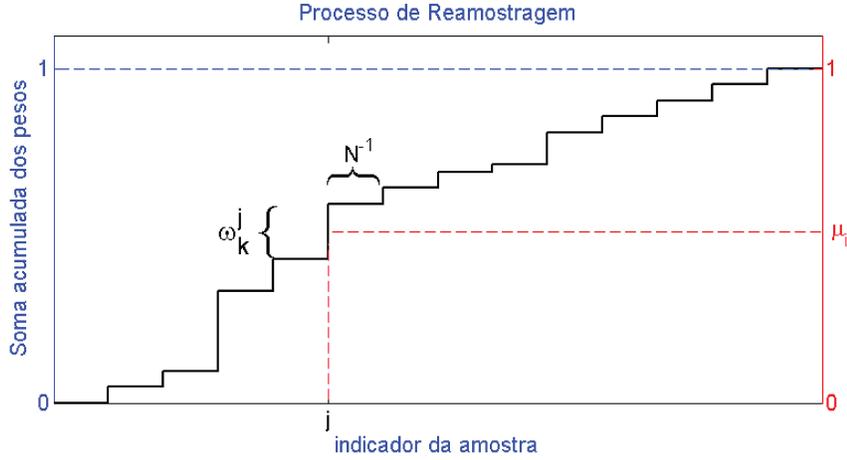


Figura 2.2: Esquema do processo de reamostragem

um determinado limiar  $N_{thr}$ . Nas referências indicadas a seguir pode-se encontrar um maior aprofundamento sobre o assunto: (Gilks & Berzuini 2001) e (Musso, Oudjane & Gland 2001).

No Algoritmo 2.1 descreve-se um pseudocódigo, proposto em (Kitagawa 1996), para realizar a reamostragem .

---

### Algoritmo 2.1: Reamostragem

---

**Entrada:**  $\{\mathbf{x}_k^i, \omega_k^i\}_{i=1}^N$   
**Saída:**  $\{\mathbf{x}_k^{j*}, \omega_k^j\}_{j=1}^N$

```

1 início
  // Construção da função de probabilidade acumulada  $CSW = [c_1, c_2, \dots, c_N]$ 
2    $c_1 \leftarrow \omega_k^1$ ;
3   para  $i = 2 : N$  faça  $c_i \leftarrow c_{i-1} + \omega_k^i$ ;
  // Mapeamento
4    $\mu_1 \leftarrow U[0, N^{-1}]$ ; // Sorteio do ponto de partida
5   para  $j = 1 : N$  faça
6      $\mu_j \leftarrow \mu_1 + N^{-1}(j - 1)$ ;
7     enquanto  $\mu_j > c_i$  faça  $i \leftarrow i + 1$ ;
8      $\mathbf{x}_k^{j*} \leftarrow \mathbf{x}_k^i$ ;
9      $\omega_k^j \leftarrow N^{-1}$ 
10  fim
11 fim

```

---

A escolha da densidade de importância  $q(\mathbf{x}_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$  influencia diretamente o grau de degeneração do algoritmo e, conseqüentemente, o seu desempenho. Conforme será demonstrado a seguir, a densidade de importância ótima condicionada por  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_k$  é dada por:

(Doucet et al. 2000)

$$\begin{aligned} q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{otm} &= p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) \\ &= \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)} \end{aligned} \quad (2.25)$$

onde  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$  é a constante de normalização dada por:

$$p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) d\mathbf{x}_k \quad (2.26)$$

Substituindo-se (2.25) em (2.20), encontra-se que a equação de atualização dos pesos é dada por:

$$\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \quad (2.27)$$

Segundo (Gustafsson 2010), a otimalidade da escolha da densidade de importância definida por (2.25) provém do fato de que, para esta escolha, a variância dos pesos de importância  $\omega_k^i$  condicionada por  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_k$  é nula. Deste modo, qualquer outra escolha para a densidade de importância iria acrescentar variância aos pesos. Isso, de acordo com a equação (2.22), iria diminuir o tamanho efetivo das amostras e, portanto, acentuar a degeneração do algoritmo. A seguir, é proposta uma demonstração, diferente da apresentada em (Doucet et al. 2000), para esta afirmação.

**Lema 2.1.** *A densidade de importância ótima, que minimiza a variância dos pesos de importância, condicionada por  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_k$  é dada por (2.25).*

**Prova:**

*Sabendo-se que  $\mathbf{x}_k$  possui densidade  $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ , o valor esperado dos pesos de importância  $\omega_k^i$ , definidos pela equação (2.20), condicionados a  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_k$  é dado por:*

$$\begin{aligned} E[\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k] &= \int \omega_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) d\mathbf{x}_k \\ &= \omega_{k-1}^i \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) d\mathbf{x}_k \end{aligned} \quad (2.28)$$

Substituindo (2.26) em (2.28):

$$E[\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k] = \omega_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \quad (2.29)$$

O segundo momento de  $\omega_k^i(\mathbf{x}_k)$  é:

$$\begin{aligned} E[(\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)^2] &= \int \left( \omega_{k-1}^i \frac{p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} \right)^2 q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) d\mathbf{x}_k \\ &= (\omega_{k-1}^i)^2 \int \frac{(p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i))^2}{q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} d\mathbf{x}_k \end{aligned} \quad (2.30)$$

E, a variância é:

$$\begin{aligned} var[\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k] &= E[(\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k)^2] - E[\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k]^2 \\ &= (\omega_{k-1}^i)^2 \left( \int \frac{(p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i))^2}{q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)} d\mathbf{x}_k - p^2(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \right) \end{aligned} \quad (2.31)$$

Substituindo (2.25) em (2.31):

$$\begin{aligned} var[\omega_k^i | \mathbf{x}_{k-1}^i, \mathbf{z}_k] &= (\omega_{k-1}^i)^2 \left( p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \int p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) d\mathbf{x}_k - p^2(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \right) \\ &= (\omega_{k-1}^i)^2 (p^2(\mathbf{z}_k | \mathbf{x}_{k-1}^i) - p^2(\mathbf{z}_k | \mathbf{x}_{k-1}^i)) \\ &= 0 \end{aligned} \quad (2.32)$$

O que é um resultado esperado, pois, dada uma partícula  $\mathbf{x}_{k-1}^i$  e a medida  $\mathbf{z}_k$ , o valor de  $\omega_k^i$  dado pela equação (2.27) é sempre o mesmo, não importando qual partícula foi obtida a partir de  $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{otm}$ . Deste modo, a variância de  $\omega_k^i$  condicionada a  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_k$  é sempre igual a 0 para o caso em que utiliza-se a densidade de importância  $q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{otm}$ . ■

Amostrar segundo (2.25) pode apresentar algumas desvantagens:

1. De um modo geral, é difícil amostrar segundo esta densidade;
2. É, em geral, difícil efetuar a etapa de atualização dos pesos quando se opta por esta densidade, uma vez que esta etapa depende do valor de  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$  calculado de acordo com a integral apresentada em (2.26).

No entanto, em alguns casos, é possível utilizar esta densidade de importância. Pode-se citar, por exemplo o caso em que  $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$  é Gaussiano (Doucet et al. 2000). Outra aplicação pode ser encontrada no Capítulo 3.2 deste trabalho, onde é apresentado um método a partir do qual é possível utilizar a densidade de importância ótima para uma certa classe de sistemas Max-Plus.

A partir dos passos desenvolvidos ao longo deste capítulo, pode-se desenvolver o pseudocódigo de um filtro de partículas genérico apresentado no Algoritmo 2.2 (Ristic et al. 2004).

---

**Algoritmo 2.2:** Filtro de Partículas Genérico

---

**Entrada:**  $\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N, \mathbf{z}_k$   
**Saída:**  $\{\mathbf{x}_k^{i*}, \omega_k^i\}_{i=1}^N$

- 1 **início**
- 2   **para**  $i = 1 : N$  **faça**
- 3     Propagar  $\mathbf{x}_k^i$  a partir da densidade de importância:  $\mathbf{x}_k^i \sim q(\mathbf{x}_k|\mathbf{x}_{k-1}^i, \mathbf{z}_k)$ ;
- 4     Atualizar os pesos utilizando (2.20):  $\omega_k^i \leftarrow \omega_{k-1}^i \frac{p(\mathbf{z}_k|\mathbf{x}_k^i)p(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i)}{q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k)}$ ;
- 5   **fim**
- 6   Calcular a soma dos pesos:  $t \leftarrow \sum_{i=1}^N \omega_k^i$ ;
- 7   **para**  $i = 1 : N$  **faça** normalizar os pesos:  $\omega_k^i \leftarrow t^{-1}\omega_k^i$  ;
- 8   Calcular  $\hat{N}_{eff}$  usando (2.23):  $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N \omega_k^i}$ ;
- 9   **se**  $\hat{N}_{eff} < N_{thr}$  **então** reamostrar utilizando o Algoritmo 2.1 ;
- 10 **fim**

---



# Filtro de Partículas Aplicado a Sistemas Max-Plus

## 3.1 Filtro de Partículas com Densidade de Importância Sub-Ótima

Nesta seção será desenvolvido um algoritmo de filtragem por partículas que utiliza a densidade de transição *a priori*  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$  como densidade de importância para obter estimativas para os estados de sistemas max-plus descritos por equações do tipo:

$$\mathbf{x}_k = A \otimes \mathbf{x}_{k-1} \oplus B \otimes \mathbf{u}_k \quad (3.1)$$

$$\mathbf{z}_k = C \otimes \mathbf{x}_k \quad (3.2)$$

onde  $\mathbf{x}_k \in \mathbb{R}^n$ ,  $\mathbf{u}_k \in \mathbb{R}^p$ ,  $\mathbf{z}_k \in \mathbb{R}^q$  e as matrizes  $A$ ,  $B$  e  $C$  possuem dimensões apropriadas.

Admite-se que os elementos das matrizes  $A$ ,  $B$  e  $C$  possam ser determinísticos ou variáveis aleatórias com distribuição uniforme ou exponencial representando, por exemplo, a incerteza quanto ao tempo de serviço de uma máquina.

A escolha de  $p(\mathbf{x}_k | \mathbf{x}_{k-1})$  como densidade de importância é uma das escolhas sub-ótimas mais populares (Ristic et al. 2004). Como será visto, as exigências para o uso desta densidade não são, em geral, muito complexas:

1. Devem-se conhecer as funções (3.1) e (3.2);
2. Deve-se ser capaz de utilizar (3.1) para obter amostras a partir de um conjunto de partículas disponíveis em  $k - 1$ ;
3. A função de verossimilhança  $p(\mathbf{z}_k|\mathbf{x}_k)$  deve ser conhecida.

Substituindo-se  $q(\mathbf{x}_k^i|\mathbf{x}_{k-1}^i, \mathbf{z}_k) = p(\mathbf{x}_k|\mathbf{x}_{k-1})$  em (2.20), encontra-se a seguinte equação de atualização dos pesos:

$$\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k|\mathbf{x}_k^i). \quad (3.3)$$

Nota-se que, no momento do cálculo de atualização do peso  $\omega_k^i$  o valor da medida  $\mathbf{z}_k$  e da partícula  $\mathbf{x}_k^i$  é conhecido. Deste modo, a atualização dos pesos pode ser feita por:

$$\omega_k^i \propto \omega_{k-1}^i V(\mathbf{x}_k^i, \mathbf{z}_k), \quad (3.4)$$

onde  $V(\mathbf{x}_k^i, \mathbf{z}_k)$  é a função de verossimilhança desenvolvida no Apêndice A. Nota-se que, neste caso, a atualização dos pesos só pode ser feita após a etapa de propagação das partículas, diferentemente do caso em que se utiliza a densidade de importância ótima, no qual a atualização do peso da partícula  $\mathbf{x}_k^i$  depende somente de  $\mathbf{z}_k^i$  e  $\mathbf{x}_{k-1}^i$ .

A escolha de  $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$  como densidade de importância permite a utilização direta de (3.1) para propagar as partículas  $\{\mathbf{x}_{k-1}^i\}_{i=1}^N$  para  $\{\mathbf{x}_k^i\}_{i=1}^N$ .

A hipótese de que os elementos da matriz  $C$  possam ser determinísticos pode levar a uma situação na qual o suporte da função de verossimilhança  $p(\mathbf{z}_k|\mathbf{x}_k^i)$  corresponda a um evento de probabilidade nula. Deste modo, todas as partículas geradas a partir de  $p(\mathbf{x}_k|\mathbf{x}_{k-1})$  serão atualizadas com peso nulo, já que, estatisticamente, a probabilidade de alguma destas partículas pertencerem ao suporte de  $p(\mathbf{z}_k|\mathbf{x}_k^i)$  é nula (Cândido, Santos-Mendes, Hardouin & Maia 2013). O exemplo a seguir ilustra uma situação na qual este fato ocorre.

**Exemplo 3.1.** *Considere um sistema descrito pelas seguintes equações:*

$$\mathbf{x}_k = \begin{bmatrix} U[0.2, 0.6] & U[0.1, 0.5] \\ U[0, 1.5] & 0.2 \end{bmatrix} \otimes \mathbf{x}_{k-1} \quad (3.5)$$

$$\mathbf{z}_k = \begin{bmatrix} 1 & 0 \\ U[1, 3] & 2 \end{bmatrix} \otimes \mathbf{x}_k \quad (3.6)$$

onde:  $\mathbf{x}_k = [x_1 \ x_2]^T$ ,  $\mathbf{z}_k = [z_1 \ z_2]^T$  e  $U[a, b]$  representa uma variável aleatória com distribuição uniforme entre  $a$  e  $b$ .

Considerando-se que em  $k - 1$  tem-se um conjunto de partículas dado por  $\{\mathbf{x}_{k-1}^i = 0, \omega_{k-1}^i = 1/N\}_{i=1}^N$  e que em  $k$  foi adquirida a medida  $\mathbf{z}_k = [1.4, 3.2]^T$ , o suporte da função  $p(\mathbf{z}_k|\mathbf{x}_k^i)$  definida pela matriz  $C$  e pela equação (A.1) do Apêndice A é dado pela região formada por:  $\{x_1, x_2 \in \mathbb{R} : x_1 = 0.4 \text{ e } x_2 \leq 1.2\}$ . Esta região corresponde a um evento de probabilidade nula dado que as partículas  $\mathbf{x}_k^i$  foram obtidas através de (3.5). Na Figura 3.1(a) pode-se observar 100 partículas geradas a partir do conjunto  $\{\mathbf{x}_{k-1}^i = 0, \omega_{k-1}^i = 0.01\}_{i=1}^{100}$  assim como a região correspondente ao suporte de  $p(\mathbf{z}_k|\mathbf{x}_k^i)$ . Na Figura 3.1(b) a região formada pela linha tracejada na Figura 3.1(a) foi ampliada, observa-se que, como esperado, nenhuma partícula pertence ao suporte de  $p(\mathbf{z}_k|\mathbf{x}_k^i)$  deste modo, o peso de todas as partículas serão atualizados para zero.

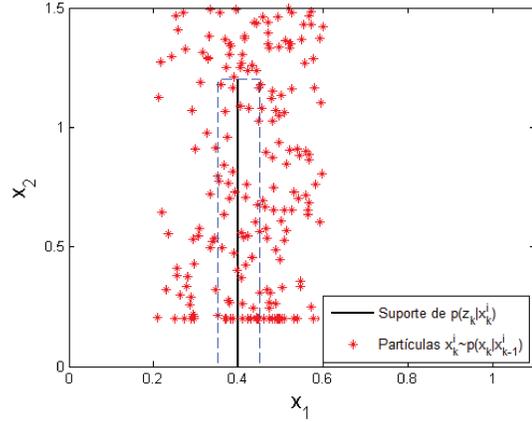
Se todas as partículas adquirirem peso nulo em  $k$ , tornar-se-á impossível obter qualquer estimativa para os estados não observados e, ainda, qualquer partícula gerada a partir do conjunto de partículas existente em  $k$  terá peso nulo. Portanto, uma etapa de *recondicionamento* das partículas deve ser realizada a fim de evitar este problema. Nesta etapa, define-se um limiar próximo à região correspondente ao suporte da função de verossimilhança e, por meio do menor caminho possível, as partículas que estiverem dentro desse limiar serão deslocadas de modo que pertençam à região de verossimilhança não nula. Considerando que, na Figura 3.1(b), a linha tracejada corresponde ao limiar, todas as partículas que se encontrarem dentro deste limite serão deslocadas para o suporte da função de verossimilhança.

No Algoritmo 3.1 descreve-se uma maneira sistemática para realizar a etapa de *recondicionamento* das partículas. Basicamente, este algoritmo calcula qual é a menor distância entre as partículas e a região de verossimilhança não nula: se esta distância for menor que o limiar estabelecido, a partícula é então deslocada de modo a pertencer à região factível (ou de verossimilhança não nula).

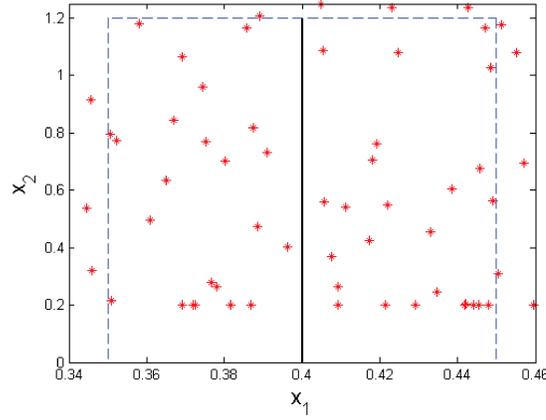
O valor do limiar  $K_{thr}$  deve ser pequeno o bastante para que a partícula deslocada  $\mathbf{x}_k^{i*}$  seja tal que a seguinte aproximação seja válida:

$$\{\mathbf{x}_k^{i*}, \omega_{k-1}^i\} \approx \{\mathbf{x}_k^i, \omega_{k-1}^i\}. \quad (3.7)$$

Deste modo, pode-se dizer os conjuntos  $\{\mathbf{x}_k^{i*}, \omega_{k-1}^i\}$  e  $\{\mathbf{x}_k^i, \omega_{k-1}^i\}$  apresentaram, aproximadamente, os mesmos comportamentos estatísticos dados  $\mathbf{x}_{k-1}^i$  e  $\mathbf{z}_{k-1}$ .



(a)



(b)

Figura 3.1: (a) Partículas geradas a partir de (3.5) e a região de verossimilhança não nula (b) Ampliação da região próxima ao suporte de  $p(\mathbf{z}_k | \mathbf{x}_k^i)$

Os estados  $x_j^{i*}$  que foram deslocados na etapa de recondicionamento não devem ser levados em conta no cálculo da função de verossimilhança, uma vez que isso envolveria cálculos com funções impulsivas, o que levaria a valores de verossimilhança infinitos ou até mesmo a indeterminações. Analisando a equação (A.1) nota-se que o cálculo da função de verossimilhança envolve produtórios. Deste modo, uma maneira de fazer com que os estados deslocados não interfiram no cálculo da verossimilhança é considerar  $p_{ij}(\tau) = 1$  se  $\tau = 0$  e  $p_{ij}(\tau) = 0$  se  $\tau \neq 0$  como a densidade de probabilidade e  $F_{ij}(\tau) = 1$  se  $\tau \geq 0$  e  $F_{ij}(\tau) = 0$  se  $\tau < 0$  como densidade de probabilidade acumulada para os elementos determinísticos  $c_{ij}$  da matriz  $C$ .

O Algoritmo 3.2 apresenta um pseudocódigo para o filtro de partículas para sistemas Max-Plus discutido ao longo deste capítulo.

---

**Algoritmo 3.1:** Recondicionamento

---

**Entrada:**  $\{\mathbf{x}_k^i\}_{i=1}^N, K_{thr}$   
**Saída:**  $\{\mathbf{x}_k^{i*}\}_{i=1}^N$   
//  $\mathbf{x}_k^i = [x_1^i \ x_2^i \ \dots \ x_n^i]^T$   
//  $\mathbf{x}_k^{i*} = [x_1^{i*} \ x_2^{i*} \ \dots \ x_n^{i*}]^T$

```
1 início
2   para  $j = 1 : n$  faça
3      $\Delta x_j \leftarrow M \gg 0$ ;
4     para  $r = 1 : q$  faça           //  $q$  é o número de linhas da matriz  $C$ 
5        $S_r \leftarrow \{j : c_{rj} \text{ é determinístico}\};$            // conjunto de índices
6       se  $c_{rj} \in S_r$  e  $z_r - (c_{rj} + x_j^i) \leq K_{thr}$  então
7         se  $|z_r - (c_{rj} + x_j^i)| \leq \Delta x_j$  então
8            $\Delta x_j \leftarrow |z_r - (c_{rj} + x_j^i)|$ ;
9            $x_j^{i*} \leftarrow z_r - c_{rj}$ ;
10          fim
11         fim
12       fim
13     fim
14 fim
```

---

## 3.2 Filtro de Partículas com Densidade de Importância Ótima

Um algoritmo para filtragem por partículas que utiliza a densidade de importância ótima pode ser desenvolvido para uma classe restrita de sistemas max-plus. Considere novamente os sistemas descritos por:

$$\mathbf{x}_k = A \otimes \mathbf{x}_{k-1} \oplus B \otimes \mathbf{u}_k \quad (3.8)$$

$$\mathbf{z}_k = C \otimes \mathbf{x}_k \quad (3.9)$$

onde  $\mathbf{x}_k \in \mathbb{R}^n$ ,  $\mathbf{u}_k \in \mathbb{R}^p$ ,  $\mathbf{z}_k \in \mathbb{R}^q$  e as matrizes  $A$ ,  $B$  e  $C$  possuem dimensões apropriadas.

Admite-se que os elementos das matrizes  $A$  e  $C$  possam ser determinísticos ou variáveis aleatórias com distribuição uniforme independentes entre si, sendo que em cada linha e em cada coluna da matriz  $C$  pode existir mais do que um elemento não nulo mas não mais que um valor determinístico. Observa-se que esta classe abrange os sistemas onde alguns estados são diretamente observados, mas não se restringe apenas a esses sistemas.

Na seção 2.2, demonstrou-se que a densidade de importância ótima é dada por:

---

**Algoritmo 3.2:** Filtro de Partículas para Sistemas Max-Plus
 

---

**Entrada:**  $\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N, \mathbf{z}_k$   
**Saída:**  $\{\mathbf{x}_k^{i*}, \omega_k^i\}_{i=1}^N$

```

1 início
2   para  $i = 1 : N$  faça
3     | Propagar  $\mathbf{x}_k^i$  utilizando (3.1):  $\mathbf{x}_k^i \leftarrow A \otimes \mathbf{x}_{k-1}^i \oplus B \otimes \mathbf{u}_k$ ;
4   fim
5   Recondicionar as partículas utilizando o Algoritmo 3.1;
6   para  $i = 1 : N$  faça
7     | Atualizar os pesos utilizando (3.4):  $\omega_k^i \leftarrow \omega_{k-1}^i V(\mathbf{x}_k^i, \mathbf{z}_k)$ ;
8   fim
9   Calcular a soma dos pesos:  $t \leftarrow \sum_{i=1}^N \omega_k^i$ ;
10  para  $i = 1 : N$  faça Normalizar os pesos:  $\omega_k^i \leftarrow t^{-1} \omega_k^i$ ;
11  Calcular  $\hat{N}_{eff}$  usando (2.23):  $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N \omega_k^i}$ ;
12  se  $\hat{N}_{eff} < N_{thr}$  então Reamostrar utilizando o Algoritmo 2.1 ;
13 fim
  
```

---

$$\begin{aligned}
 q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} &= p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) \\
 &= \frac{p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)}{p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)},
 \end{aligned} \tag{3.10}$$

onde

$$p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = \int_{-\infty}^{\infty} p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) d\mathbf{x}_k. \tag{3.11}$$

E, a equação de atualização dos pesos é dada por:

$$\omega_k^i \propto \omega_{k-1}^i p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) \tag{3.12}$$

Por conveniência, (3.10) pode ser reescrita como:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k) = \frac{V(\mathbf{x}_k; \mathbf{z}_k, \mathbf{x}_{k-1}^i)}{\Psi}, \tag{3.13}$$

onde:

$$\begin{aligned}
 V(\mathbf{x}_k; \mathbf{z}_k, \mathbf{x}_{k-1}^i) &= p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) \\
 &= p(\mathbf{z}_k | \mathbf{x}_k) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i)
 \end{aligned} \tag{3.14}$$

e  $\Psi$  é a constante de normalização dada por

$$\Psi = \int_{-\infty}^{\infty} p(\mathbf{z}_k | \mathbf{x}_k, \mathbf{x}_{k-1}^i) p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) d\mathbf{x}_k \quad (3.15)$$

Deste modo, (3.12) pode ser reescrita como:

$$\omega_k^i \propto \Psi \omega_{k-1}^i \quad (3.16)$$

De um modo geral, apenas alguns estados são relacionados com a saída por meio da matriz  $C$ , ou seja, apenas alguns estados são observados. Deste modo, pode-se particionar o vetor de estados da seguinte maneira:  $\mathbf{x} = [(\gamma^o)^T (\gamma^{no})^T]^T$ , onde  $\gamma^o$  é o vetor de estados observados e  $\gamma^{no}$  é o vetor de estados não observados. Devido à independência dos elementos das matrizes  $A$  e  $C$  pode-se observar que os vetores  $\gamma^o$  e  $\gamma^{no}$  são independentes. Portanto a seguinte fatoração é válida:

$$p(\mathbf{x}_k | \mathbf{x}_{k-1}^i) = p(\gamma^o | \mathbf{x}_{k-1}^i) p(\gamma^{no} | \mathbf{x}_{k-1}^i) \quad (3.17)$$

Por definição, somente os estados observados estão relacionados com a saída, portanto:

$$p(\mathbf{z}_k | \mathbf{x}_k) = p(\mathbf{z}_k | \gamma^o) \quad (3.18)$$

Sendo assim, (3.14) pode ser reescrita como se segue:

$$\begin{aligned} V(\mathbf{x}_k; \mathbf{z}_k, \mathbf{x}_{k-1}) &= p(\mathbf{z}_k | \gamma^o) p(\gamma^o | \mathbf{x}_{k-1}^i) p(\gamma^{no} | \mathbf{x}_{k-1}^i) \\ &= V_1(\gamma^o) V_2(\gamma^{no}) \end{aligned} \quad (3.19)$$

onde:

$$V_1(\gamma^o) = p(\mathbf{z}_k | \gamma^o) p(\gamma^o | \mathbf{x}_{k-1}^i) \quad (3.20)$$

$$V_2(\gamma^{no}) = p(\gamma^{no} | \mathbf{x}_{k-1}^i) \quad (3.21)$$

Em síntese,

$$q(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)_{opt} = \frac{V_1(\gamma^o) V_2(\gamma^{no})}{\Psi}. \quad (3.22)$$

Deseja-se, portanto, utilizar (3.19) para gerar partículas distribuídas segundo a densidade de importância ótima. Neste caso, nota-se que o vetor de estados  $\mathbf{x}_k$  pode ser construído gerando-se os vetores  $\gamma_o$  e  $\gamma_{no}$  de maneira independente a partir das equações (3.20) e (3.21) respectivamente.

O sistema descrito pelas equações (3.8) e (3.9) pode ser reescrito como:

$$\gamma^o = A_o \otimes \mathbf{x}_{k-1} \oplus B_o \otimes \mathbf{u}_k \quad (3.23)$$

$$\gamma^{no} = A_{no} \otimes \mathbf{x}_{k-1} \oplus B_{no} \otimes \mathbf{u}_k \quad (3.24)$$

$$\mathbf{z}_k = C_o \otimes \gamma^o \quad (3.25)$$

onde as matrizes  $A_o$  e  $B_o$  foram construídas removendo-se das matrizes  $A$  e  $B$  as linhas relacionadas aos estados não observados. Por outro lado, as matrizes  $A_{no}$  e  $B_{no}$  foram construídas removendo-se das matrizes  $A$  e  $B$  as linhas relacionadas aos estados observados. A matriz  $C_o$  foi construída removendo-se as colunas nulas da matriz  $C$ .

A partir da equação de verossimilhança definida pela equação (A.1), definida no Apêndice A, é possível obter formas analíticas para as equações (3.20) e (3.21):

$$V_1(\gamma^o) = \prod_{r=1}^q \left( \sum_{s=1}^{n1} d_{rs}(z_r - \gamma_s^o) \prod_{\substack{l=1, \\ l \neq j}}^{n1} N_{rl}(z_r - \gamma_l^o) \right) \prod_{i=1}^{n1} \left( \sum_{j=1}^n p_{ij}(\gamma_i^o - x'_j) \prod_{\substack{k=1, \\ k \neq j}}^n F_{ik}(\gamma_i^o - x'_k) \right) \quad (3.26)$$

onde  $\gamma_i^o$ ,  $z_i$  e  $x'_i$  representam o  $i$ -ésimo elemento dos vetores  $\gamma^o$ ,  $\mathbf{z}_k$  e  $\mathbf{x}_{k-1}$  respectivamente, e  $n$ ,  $n1$  e  $q$  são as dimensões dos vetores  $\mathbf{x}_{k-1}$ ,  $\gamma^o$  e  $\mathbf{z}_k$  respectivamente. As funções  $d_{rs}$  e  $N_{rl}$  representam, respectivamente, a densidade de probabilidade correspondente ao elemento  $c_{rs}^o$  da matriz  $C_o$  e a função de probabilidade acumulada do elemento  $c_{rl}^o$ . Analogamente, as funções  $p_{ij}$  e  $F_{ik}$  representam a função de densidade de probabilidade e a função de probabilidade acumulada dos elementos da matriz  $A_o$ .

$$V_2(\gamma^{no}) = \prod_{i=1}^{n2} \left( \sum_{j=1}^n p_{ij}(\gamma_i^{no} - x'_j) \prod_{\substack{k=1, \\ k \neq j}}^n F_{ik}(\gamma_i^{no} - x'_k) \right) \quad (3.27)$$

onde  $\gamma_i^{no}$  representa o  $i$ -ésimo elemento do vetor  $\gamma^{no}$  cuja dimensão é  $n2$ , e as funções  $p_{ij}$  e  $F_{ik}$  representam a função de densidade de probabilidade e a função de probabilidade dos elementos da matriz  $A_{no}$ .

A geração do vetor  $\gamma^{no}$  distribuído segundo (3.27) pode ser feita diretamente por meio da equação (3.24). Por outro lado, gerar o vetor  $\gamma^o$  utilizando (3.26) de forma direta pode apresentar dificuldades de implementação. De fato, de um modo geral, a função (3.26) representa uma superfície em  $\mathbb{R}^{n1}$  cujos elementos  $\gamma_1^o, \gamma_2^o, \dots, \gamma_{n1}^o$  são dependentes. Este fato dificulta a geração a partir desta equação.

Nas próximas subseções, será discutido um método para gerar amostras distribuídas segundo (3.22) utilizando uma função secundária a partir da qual seja mais fácil gerar as amostras desejadas.

### 3.2.1 Geração de Variáveis Aleatórias: Aceitação-Rejeição

O Método de Aceitação-Rejeição permite obter amostras distribuídas segundo uma densidade  $f(x)$  (densidade objetivo) a partir de uma função  $g(x)$  conhecida que satisfaça as seguintes condições (Law & Kelton 2000) :

1.  $g(x) \geq f(x), \forall x;$

2.  $\int_{-\infty}^{\infty} g(x)dx = c.$

A função  $g(x)$  é, portanto, uma função majorante de  $f(x)$  e, da condição 2, tem-se que  $h(x) = \frac{g(x)}{c}$  é uma densidade de probabilidade. Além das condições apresentadas para a função  $g(x)$ , é fundamental que seja possível amostrar segundo a densidade  $h(x)$ .

O Método de Aceitação-Rejeição é apresentado no Algoritmo 3.3.

---

**Algoritmo 3.3:** Método de Aceitação-Rejeição

---

**Dados:**  $f(x), g(x), h(x)$

**Saída:**  $\mathbf{X}$

```

1 início
2    $aceitou \leftarrow 0$ ;
3   enquanto  $aceitou = 0$  faça
4     Gerar uma v.a.  $\mathbf{Y}$  com densidade  $h(x)$ ;
5     Gerar um número aleatório  $\mathbf{U}$ , independente de  $\mathbf{Y}$  e com distribuição uniforme entre 0 e 1;
6     se  $\mathbf{U} \leq \frac{f(\mathbf{Y})}{g(\mathbf{Y})}$  então
7        $\mathbf{X} \leftarrow \mathbf{Y}$ ;
8        $aceitou \leftarrow 1$ ;
9     fim
10  fim
11 fim
```

---

No apêndice B, demonstra-se que a variável aleatória  $X$  gerada a partir deste algoritmo possui densidade de probabilidade igual a  $f(x)$ , ou seja,  $P[X \leq x] = P[Y \leq x | U \leq f(Y)/g(Y)]$ . Demonstra-se também que a probabilidade de aceitação da variável  $Y$  é inversamente proporcional a  $c$ , ou seja,  $P[\mathbf{U} \leq f(\mathbf{Y})/g(\mathbf{Y})] = 1/c$ . Esta medida pode ser usada para avaliar a eficiência do algoritmo: quanto maior for a probabilidade de aceitação, mais eficiente será o algoritmo.

Neste trabalho, o método de Aceitação-Rejeição será usado para obter partículas distribuídas segundo a densidade de importância definida pela equação (3.22). Portanto, deve-se buscar uma função majorante  $\Gamma(\mathbf{x}_k)$  a partir da qual seja possível de se obter amostras e que satisfaça às mesmas condições impostas à função  $g(x)$  apresentadas ao início desta subseção. Deste modo,  $\Gamma(\mathbf{x}_k)$  deve ser definida de maneira que:

$$\Gamma(\mathbf{x}_k) \geq \frac{V_1(\gamma^o)V_2(\gamma^{no})}{\Psi}, \quad \forall \mathbf{x}_k$$

Para satisfazer a esta condição, a função  $\Gamma(\mathbf{x}_k)$  pode ser definida como:

$$\Gamma(\mathbf{x}_k) = \frac{\bar{V}_1(\gamma^o)V_2(\gamma^{no})}{\Psi}, \quad (3.28)$$

onde  $\bar{V}_1(\gamma^o) \geq V_1(\gamma^o)$ ,  $\forall \gamma^o$  e  $\int_{-\infty}^{\infty} \bar{V}_1(\gamma^o)d\gamma^o = c'$ .

Nota-se que não é necessário definir uma função majorante para  $V_2(\gamma^{no})$ , pois, como mencionado anteriormente, o vetor  $\gamma^{no}$  pode ser gerado a partir da equação (3.24).

A integral da função  $\Gamma(\mathbf{x}_k)$  é dada por:

$$\begin{aligned} \int_{-\infty}^{\infty} \Gamma(\mathbf{x}_k) d\mathbf{x}_k &= \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \frac{\bar{V}_1(\gamma^o) V_2(\gamma^{no})}{\Psi} d\gamma^{no} d\gamma^o \\ &= \int_{-\infty}^{\infty} \frac{\bar{V}_1(\gamma^o)}{\Psi} \left( \int_{-\infty}^{\infty} V_2(\gamma^{no}) d\gamma^{no} \right) d\gamma^o \end{aligned}$$

mas, de acordo com a equação (3.21),  $V_2(\gamma^{no})$  é uma densidade de probabilidade. Portanto  $\int_{-\infty}^{\infty} V_2(\gamma^{no}) d\gamma^{no} = 1$ , Assim:

$$\begin{aligned} \int_{-\infty}^{\infty} \Gamma(\mathbf{x}_k) d\mathbf{x}_k &= \int_{-\infty}^{\infty} \frac{\bar{V}_1(\gamma^o)}{\Psi} d\gamma^o \\ &= \frac{c'}{\Psi} = c. \end{aligned}$$

Deste modo, a função  $\mathbf{h}(\mathbf{x}_k) = \frac{\Psi}{c'} \Gamma(\mathbf{x}_k) = \frac{\bar{V}_1(\gamma^o) V_2(\gamma^{no})}{c'}$  é uma densidade de probabilidade. Nota-se ainda, que a função  $\mathbf{h}_1(\mathbf{x}_k) = \frac{\bar{V}_1(\gamma^o)}{c'}$  também é uma densidade de probabilidade, uma vez que  $\int_{-\infty}^{\infty} \bar{V}_1(\gamma^o) d\gamma^o = c'$ .

Observa-se, na linha 6 do Algoritmo 3.3, que uma das etapas do método de aceitação/rejeição envolve uma divisão da densidade objetivo pela função majorante. No problema em questão, esta divisão seria dada por:

$$\frac{V_1(\gamma^o) V_2(\gamma^{no}) / \Psi}{\bar{V}_1(\gamma^o) V_2(\gamma^{no}) / \Psi} = \frac{V_1(\gamma^o)}{\bar{V}_1(\gamma^o)}. \quad (3.29)$$

Sendo assim, o vetor  $\gamma^{no}$  amostrado segundo a densidade  $V_2(\gamma^{no})$  não interfere na aceitação ou rejeição do algoritmo. Portanto, apenas o vetor  $\gamma^o$ , gerado a partir da função majorante  $\bar{V}_1(\gamma^o)$  deverá ser testado quanto à sua aceitação ou rejeição pelo algoritmo. O Algoritmo 3.4 representa um pseudo-código para o Método de Aceitação-Rejeição específico para este caso. Neste algoritmo é gerado apenas o vetor correspondente aos estados observados. O vetor de

estados não observados é obtido a partir da equação (3.24).

---

**Algoritmo 3.4:** Método de Aceitação-Rejeição para Amostragem Ótima em Sistemas Max Plus

---

**Dados:**  $V_1(\gamma^o)$ ,  $\bar{V}_1(\gamma^o)$ ,  $c'$

**Saída:**  $\gamma^o$

```

1 início
2    $aceitou \leftarrow 0$ ;
3   enquanto  $aceitou = 0$  faça
4     Gerar uma v.a.  $\mathbf{Y}$  com densidade  $\mathbf{h}_1(\mathbf{x}_k) = \frac{\bar{V}_1(\gamma^o)}{c'}$ ;
5     Gerar um número aleatório  $\mathbf{U}$ , independente de  $\mathbf{Y}$ , com distribuição uniforme entre 0 e 1;
6     se  $\mathbf{U} \leq \frac{V_1(\gamma^o)}{\bar{V}_1(\gamma^o)}$  então
7        $\gamma^o \leftarrow \mathbf{Y}$ ;
8        $aceitou \leftarrow 1$ ;
9     fim
10  fim
11 fim

```

---

A equação (3.29) mostra que, devido à escolha da função  $\Gamma(x)$  dada pela equação (3.28), o algoritmo apresentado é independente do valor da constante  $\Psi$ . Tal característica é interessante, uma vez que o cálculo direto desse valor a partir da integral apresentada em (3.15) nem sempre é possível. No entanto uma estimativa dessa integral pode ser obtida utilizando o resultado de que a probabilidade de aceitação do algoritmo é:

$$P[\mathbf{U} \leq \frac{V_1(\gamma^o)V_2(\gamma^{no})/\Psi}{\bar{V}_1(\gamma^o)V_2(\gamma^{no})/\Psi}] = P[\mathbf{U} \leq \frac{V_1(\gamma^o)}{\bar{V}_1(\gamma^o)}] = \Psi/c'$$

dado que  $\int \Gamma(x)dx = c'/\Psi$ . Deste modo, supondo que a constante  $c'$  é conhecida, uma estimativa para a constante  $\Psi$  pode ser dada por:

$$\Psi = c' \frac{\lambda}{\Lambda} \tag{3.30}$$

onde  $\Lambda$  é um número fixo de iterações do algoritmo e  $\lambda$  é o número de vezes em que a v.a  $\mathbf{Y}$  é aceita pelo algoritmo. Deste modo,  $\lambda/\Lambda$  é uma estimativa para a probabilidade de aceitação do algoritmo.

Portanto, para que este algoritmo possa ser utilizado, a função majorante  $\bar{V}_1(\gamma^o)$  e a constante  $c'$  devem ser conhecidas; além disso, deve-se ser capaz de obter amostras a partir dessa função. Na próxima subseção serão demonstrados os passos necessários para a

construção desta função majorante e, ao final, serão propostos os algoritmos para a construção da função majorante e para a amostragem a partir desta função.

### 3.2.2 Função Majorante

A função majorante  $\bar{V}_1(\gamma^o)$  é obtida a partir da função  $V_1(\gamma^o)$  definida pela equação (3.26). Nesta equação, sabe-se que  $N_{rl}(z_r - \gamma_l^o)$  e  $F_{ik}(\gamma_i^o - x'_k)$  são funções de probabilidade acumulada e, portanto,  $0 \leq N_{rl}(z_r - \gamma_l^o) \leq 1$  e  $0 \leq F_{ik}(\gamma_i^o - x'_k) \leq 1$ . Deste modo, a seguinte relação é válida:

$$V_1(\gamma^o) \leq \prod_{r=1}^q \left( \sum_{s=1}^{n1} d_{rs}(z_r - \gamma_s^o) \right) \prod_{i=1}^{n1} \left( \sum_{j=1}^n p_{ij}(\gamma_i^o - x'_j) \right) \quad (3.31)$$

Definindo os conjuntos  $S_r$  e  $J_i$  como o conjunto dos elementos da linha  $r$  da matriz  $C_o$  que são determinísticos e, analogamente,  $J_i$  como o conjunto dos elementos da linha  $i$  da matriz  $A_o$  que são determinísticos, pode-se reescrever o lado direito da desigualdade (3.31) como:

$$\prod_{r=1}^q \left( \sum_{s \in S_r} \delta(z_r - \gamma_s^o - c_{rs}^o) + \sum_{s \notin S_r} d_{rs}(z_r - \gamma_s^o) \right) \prod_{i=1}^{n1} \left( \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) + \sum_{j \notin J_i} p_{ij}(\gamma_i^o - x'_j) \right) \quad (3.32)$$

onde  $\delta(\cdot)$  representa a função impulso.

Sabendo-se que  $d_{rs}$  é uma densidade de probabilidade com distribuição uniforme entre  $\underline{d}_{rs}$  e  $\overline{d}_{rs}$  tem-se que:

$$\sum_{s \notin S_r} d_{rs}(z_r - \gamma_s^o) \leq \sum_{s \notin S_r} D_{rs} = D_r \quad (3.33)$$

onde

$$D_{rs} = \frac{1}{\overline{d}_{rs} - \underline{d}_{rs}}$$

Analogamente:

$$\sum_{j \notin J_i} p_{ij}(\gamma_i^o - x'_j) \leq \sum_{j \notin J_i} P_{ij} = P_i \quad (3.34)$$

onde

$$P_{ij} = \frac{1}{\overline{p}_{ij} - \underline{p}_{ij}}$$

Deste modo, tem-se que:

$$V_1(\gamma^o) \leq \prod_{r=1}^q \left( D_r + \sum_{s \in S_r} \delta(z_r - \gamma_s^o - c_{rs}^o) \right) \cdot \prod_{i=1}^{n1} \left( P_i + \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right) \quad (3.35)$$

Dada a restrição de que pode haver no máximo um elemento determinístico por linha da matriz  $C$  pode-se afirmar que  $|S_r| \leq 1$ . Este fato permite reescrever o primeiro produtório do lado direito da expressão (3.35) como:

$$\prod_{r=1}^q \left( D_r + \sum_{s \in S_r} \delta(z_r - \gamma_s^o - c_{rs}^o) \right) = \prod_{r=1}^q \prod_{i=1}^{n1} H_{ri} \quad (3.36)$$

onde:

$$H_{ri} = \begin{cases} D_r^{\frac{1}{n}}, & \text{se } |S_r| = 0 \\ D_r + \delta(z_r - \gamma_i^o - c_{ri}^o), & \text{se } |S_r| = 1 \text{ e } S_r = \{i\} \\ 1, & \text{se } |S_r| = 1 \text{ e } S_r = \{j\}, j \neq i \end{cases} \quad (3.37)$$

Deste modo, tem-se que:

$$V_1(\gamma^o) \leq \prod_{i=1}^{n1} \left[ \left( P_i + \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right) \cdot \prod_{r=1}^q H_{ri} \right] \quad (3.38)$$

Dadas as matrizes  $A$  e  $C$ , o vetor de medidas  $\mathbf{z}_k$  e a partícula  $\mathbf{x}_{k-1}^i$ , pode-se definir limites para  $\mathbf{x}_k^i$  fora dos quais as funções  $V_1(\gamma^o)$  e  $V_2(\gamma^{no})$  são garantidamente nulas. Lembrando que os elementos  $a_{ij}$  da matriz  $A$  são variáveis aleatórias uniformemente distribuídas entre  $\underline{a}_{ij}$  e  $\bar{a}_{ij}$ , considere a matriz  $\underline{A}$  constituída pelos elementos  $\underline{a}_{ij}$  e a matriz  $\bar{A}$  constituída pelos elementos  $\bar{a}_{ij}$ . De maneira análoga, pode-se definir as matrizes  $\underline{C}$  e  $\bar{C}$ .

Deste modo, a partir da equação (3.8), o limitante inferior para  $\mathbf{x}_k^i$  é dado por:

$$\underline{\mathbf{x}}_k^i = \underline{A} \otimes \mathbf{x}_{k-1}^i \quad (3.39)$$

onde  $\otimes$  representa a operação *produto* na álgebra max-plus.

Pode-se dizer ainda que:

$$\mathbf{x}_k^i \leq \bar{A} \otimes \mathbf{x}_{k-1}^i \quad (3.40)$$

Mas, a partir da equação (3.9) tem-se que:

$$\mathbf{x}_k^i \leq \underline{C} \wp \mathbf{z}_k \quad (3.41)$$

onde  $\wp$  representa a operação de *residuação* definida na subseção 1.2.2.

Portanto, o limitante superior para  $\mathbf{x}_k^i$  é dado por:

$$\bar{\mathbf{x}}_k^i = \mathbf{x}_{max1}^i \wedge \mathbf{x}_{max2}^i \quad (3.42)$$

onde  $\wedge$  é o operador de mínimo ponto a ponto, e:

$$\mathbf{x}_{max1}^i = \bar{A} \otimes \mathbf{x}_{k-1}^i \quad (3.43)$$

$$\mathbf{x}_{max2}^i = \underline{C} \wp \mathbf{z}_k \quad (3.44)$$

A partir destes limitantes, pode se definir:  $\underline{\gamma}^o$  e  $\bar{\gamma}^o$  como os limitantes inferior e superior para os estados observados e, também,  $\underline{\gamma}^{no}$  e  $\bar{\gamma}^{no}$  como os limitantes inferior e superior para os estados não observados.

Portanto, a função majorante procurada é dada por:

$$\bar{V}_1(\gamma^o) = \begin{cases} \prod_{i=1}^{n1} \left[ \left( P_i + \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right) \prod_{r=1}^q H_{ri} \right], & \text{se } \underline{\gamma}^o \leq \gamma^o \leq \bar{\gamma}^o \\ 0, & \text{caso contrário.} \end{cases} \quad (3.45)$$

Observa-se que esta função tem um formato geral do tipo:

$$\bar{V}_1(\gamma^o) = \rho_1(\gamma_1^o) \rho_2(\gamma_2^o) \cdots \rho_{n1}(\gamma_{n1}^o),$$

Sendo que cada função  $\rho_i(\gamma_i^o)$  pode assumir uma das duas formas a seguir:

**Caso 1** se  $S_r \neq \{i\} \forall r$ , ou seja, se o estado  $\gamma_i^o$  não se relaciona com nenhuma saída de maneira determinística,  $\rho_i(\gamma_i^o)$  é dado por:

$$\rho_i(\gamma_i^o) = K_i \left( P_i + \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right), \quad \underline{\gamma}_i^o \leq \gamma_i^o \leq \bar{\gamma}_i^o \quad (3.46)$$

**Caso 2** se  $\exists r \in \{1, \dots, q\} : S_r = \{i\}$ ,  $\rho_i(\gamma_i^o)$  é dado por:

$$\begin{aligned} \rho_i(\gamma_i^o) &= K_i \left( P_i + \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right) \left( D_r + \delta(z_r - \gamma_i^o - c_{ri}^o) \right) \\ &= K_i \left( D_r P_i + D_r \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) + P_i \delta(z_r - \gamma_i^o - c_{ri}^o) \right. \\ &\quad \left. + \delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) \right), \quad \underline{\gamma}_i^o \leq \gamma_i^o \leq \bar{\gamma}_i^o \quad (3.47) \end{aligned}$$

Deste modo, pode-se obter amostras para cada estado  $\gamma_i^o$  de maneira independente a partir densidade  $h_i(\gamma_i^o) = \rho_i(\gamma_i^o)/c'_i$ , onde  $c'_i = \int \rho_i(\gamma_i^o) d\gamma_i^o$ .

Se  $\rho_i(\gamma_i^o)$  é dado por (3.46), tem-se que:

$$c'_i = K_i \left( P_i(\bar{\gamma}_i^o - \underline{\gamma}_i^o) + |J_i| \right).$$

Mas, se  $\rho_i(\gamma_i^o)$  é dado por (3.47) deve-se notar que:

$$\delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) > 0 \Rightarrow c'_i \rightarrow \infty.$$

No entanto, será demonstrado a seguir que se:  $\delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) > 0$  então  $\bar{\gamma}_i^o = \underline{\gamma}_i^o = z_r - c_{ri}^o$ , o que determina completamente o valor do estado  $\gamma_i^o$ , portanto, a densidade  $h_i(\gamma_i^o)$  é dada, neste caso, por  $h_i(\gamma_i^o) = \delta(z_r - x_i - c_{ri})$  e  $c'_i = \delta(0)$ .

**Lema 3.1.** Se  $\delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) > 0$  então  $\bar{\gamma}_i^o = \underline{\gamma}_i^o = z_r - c_{ri}^o$ .

**Prova:**

*Sabe-se que:*

$$\delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}^o) > 0 \Rightarrow \exists j \in J_i : x'_j + a_{ij}^o = z_r - c_{ri}^o.$$

*Pela equação (3.39), pode-se afirmar que:*

$$\underline{\gamma}_i^o \geq x'_j + a_{ij}^o$$

e, pelas equações (3.42) e (3.44), tem-se que:

$$\bar{\gamma}_i^o \leq z_r - c_{ri}^o.$$

Mas,  $x'_j + a_{ij}^o = z_r - c_{ri}^o$  e  $\bar{\gamma}_i^o \geq \underline{\gamma}_i^o$ . Portanto:

$$\bar{\gamma}_i^o \leq z_r - c_{ri}^o \leq \underline{\gamma}_i^o \Rightarrow \bar{\gamma}_i^o = z_r - c_{ri}^o = \underline{\gamma}_i^o.$$

■

Finalmente, se  $\rho_i(\gamma_i^o)$  é dado por (3.47) e

$$\delta(z_r - \gamma_i^o - c_{ri}^o) \sum_{j \in J_i} \delta(\gamma_i^o - x'_j - a_{ij}) = 0,$$

o valor de  $c'_i$  é dado por:

$$c'_i = K_i \left( D_r P_i (\bar{\gamma}_i^o - \underline{\gamma}_i^o) + D_r |J_i| + P_i \right).$$

O exemplo 3.2 apresenta uma maneira de se obter amostras a partir de funções do tipo apresentado em (3.47).

**Exemplo 3.2.** *Seja a função:*

$$\rho(x) = 5 [6 + 2\delta(x - 1/2) + 2\delta(x - 1) + 3\delta(x - 3/2)], \quad 0 \leq x \leq 2.$$

*Tem-se, neste caso, que:*

$$c' = \int \rho(x) dx = 5[6(2 - 0) + 2 + 2 + 3] = 95,$$

e

$$h(x) = \frac{1}{19} (6 + 2\delta(x - 1/2) + 2\delta(x - 1) + 3\delta(x - 3/2)), \quad 0 \leq x \leq 2.$$

*Nota-se, portanto, que a probabilidade de  $x$  estar uniformemente distribuído entre 0 e 2 é de  $(2 - 0) * 6/19 = 12/19$ . Além disso, a probabilidade de  $x = 1$  é  $2/19$ , a probabilidade de  $x = 1/2$  é  $2/19$  e a probabilidade de  $x = 3/2$  é  $3/19$ . Na Figura ?? encontra-se o gráfico da densidade  $h(x)$ .*

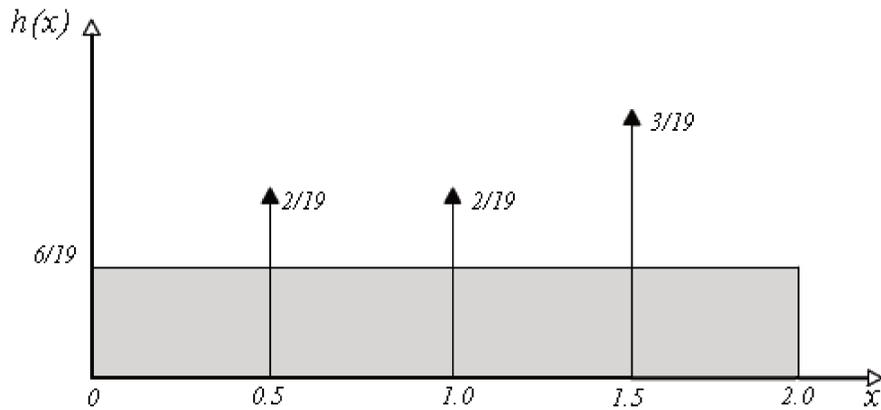


Figura 3.2: Densidade  $h(x)$  com componentes impulsivas e uniformes

Deste modo, pode-se associar um vetor  $W$  à densidade  $h(x)$  dado por:

$$W = [12/19 \quad 2/19 \quad 2/19 \quad 3/19].$$

Pode-se ainda, definir um vetor  $P$  contendo os deslocamentos das funções  $\delta$  que aparecem em  $\rho(x)$ :

$$P = [1/2 \quad 1 \quad 3/2].$$

A partir do vetor  $W$ , pode-se construir o seguinte vetor de probabilidade acumulada:

$$CSW = [12/19 \quad 14/19 \quad 16/19 \quad 1].$$

Considerando que  $U[a, b]$  representa uma densidade uniforme distribuída entre  $a$  e  $b$ , pode-se utilizar o seguinte algoritmo para gerar uma amostra com distribuição  $h(x)$ :

---

**Dados:**  $CSW, P$

**Saída:**  $x$

**1 início**

**2**    Obter um número aleatório  $u$  com distribuição entre 0 e 1:  $u \sim U[0, 1]$ ;

**3**     $i \leftarrow 1$ ; **enquanto**  $u > CSW(i)$  **faça**  $i \leftarrow i + 1$ ;

**4**    **se**  $i = 1$  **então**  $x \sim U[0, 2]$ ;

**5**    **senão**  $x \leftarrow P(i - 1)$ ;

**6 fim**

---

No Algoritmo 3.5, que será apresentado a seguir, propõe-se uma maneira de construir

a função majorante  $\bar{V}_1(\gamma^\circ)$ . A saída deste algoritmo são as estruturas  $\rho\{i\}$ ,  $i = 1, \dots, n$ , contendo os seguintes campos:

1.  $\rho\{i\}.\mathbf{const}$ : representa a constante  $K_i$  da equação 3.47;
2.  $\rho\{i\}.\mathbf{unif}$ : é um vetor cujo primeiro elemento contém a amplitude da componente uniforme da função  $\rho_i(x_i)$ , o segundo elemento contém o valor do limitante inferior  $\underline{\gamma}_i^\circ$  e no terceiro elemento encontra-se o valor do limitante superior  $\bar{\gamma}_i^\circ$ . Na equação 3.47 teria-se  $\rho\{i\}.\mathbf{unif} = \left[ D(r) \cdot P(i) \quad \underline{\gamma}_i^\circ \quad \bar{\gamma}_i^\circ \right]$ ;
3.  $\rho\{i\}.\mathbf{delta1}$ : é uma matriz na qual as componentes do tipo  $D \cdot \delta(\gamma_i^\circ - t)$  são armazenadas. Deste modo, se na função  $\rho_i(\gamma_i^\circ)$  aparecer uma soma de termos do tipo  $D_1 \cdot \delta(\gamma_i^\circ - t_1) + D_2 \cdot \delta(\gamma_i^\circ - t_2) + D_3 \cdot \delta(\gamma_i^\circ - t_3)$ , então:

$$\rho\{i\}.\mathbf{delta1} = \begin{bmatrix} D_1 & D_2 & D_3 \\ t_1 & t_2 & t_3 \end{bmatrix}^T.$$

4.  $\rho\{i\}.\mathbf{delta2}$ : é um vetor utilizado para armazenar os deslocamentos  $t$  de componentes do tipo  $\delta^2(\gamma^\circ - t)$ .

Como mencionado no final da subseção 3.2.1, para que o algoritmo de aceitação/rejeição possa ser utilizado, a função majorante  $\bar{V}_1(\gamma^\circ)$  e a constante  $c'$  devem ser conhecidas; além disso, deve-se ser capaz de obter amostras a partir dessa função. O Algoritmo 3.6 utiliza a função majorante construída a partir do Algoritmo 3.5 para gerar amostras distribuídas segundo  $\bar{V}_1(\gamma^\circ)$ . Além das amostras geradas, este algoritmo tem como saída o valor da constante  $c'$  e ainda o valor da função  $\bar{V}_1(\gamma^\circ)$  para cada amostra obtida. Como as funções  $V_1(\gamma^\circ)$  e  $\bar{V}_1(\gamma^\circ)$  envolvem componentes impulsivas, o valor destas funções em determinados pontos de seus respectivos domínios tenderá a infinito. Deste modo, no teste de aceitação/rejeição, pode aparecer uma divisão entre duas funções que tendem a infinito, necessitando-se assim do cálculo de um limite. Para evitar este cálculo de limite pode-se considerar a seguinte aproximação:  $\delta(0) \approx M$ . O valor de  $M$  deve ser grande o bastante para que esta aproximação seja verdadeira, por outro lado, o valor de  $M$  não pode ser tão grande a ponto de levar a um *overflow* durante o cálculo do valor das funções  $V_1(\gamma^\circ)$  e  $\bar{V}_1(\gamma^\circ)$ . Analisando a equação (3.45), o máximo valor da função  $\bar{V}_1(\gamma^\circ)$  é dado por  $\delta(0)^{n+q}$ , onde  $k > 0$ . Deste modo, se  $\Upsilon$  é o máximo valor finito que determinada máquina consiga representar, o valor de  $M$  pode ser definido como  $M = \Upsilon^{\frac{1}{n+q}}$ .

---

**Algoritmo 3.5:** Construção da Função Majorante
 

---

 Entrada:  $\mathbf{x}_{k-1}$ ,  $\underline{\gamma}^o$ ,  $\bar{\gamma}^o$ ,  $\mathbf{z}_k$ 

 Saída:  $\rho$ 

```

1 início
2    $n1 \leftarrow |\underline{\gamma}^o|$ ;
3    $q \leftarrow |\mathbf{z}_k|$ ;
4   para  $i = 1 : n1$  faça
5      $J\{i\} \leftarrow \{j : a_{ij}^o == \bar{a}_{ij}^o\}$ ;
6      $L\{i\} \leftarrow \{j : a_{ij}^o + x'_i \leq \bar{\gamma}_i^o \text{ e } \underline{\gamma}_i^o \leq \bar{a}_{ij}^o + x'_i\}$ ;
7      $K(i) \leftarrow 1$ ;
8      $P(i) \leftarrow \sum_{j \in L\{i\}, j \notin J\{i\}} \frac{1}{\bar{a}_{ij}^o - a_{ij}^o}$ ;
9   fim
10  para  $r = 1 : q$  faça
11     $S\{r\} \leftarrow \{s : c_{rs}^o == \bar{c}_{rs}^o\}$ ;
12     $D(r) \leftarrow \sum_{s \notin S\{r\}} \frac{1}{\bar{c}_{rs}^o - c_{rs}^o}$ ;
13    se  $|S\{r\}| = 0$  então
14      para  $i = 1 : n1$  faça  $K(i) \leftarrow K(i) \sqrt{D(r)}$ ;
15    fim
16  fim
17  para  $i = 1 : n1$  faça
18     $\rho\{i\}.const \leftarrow K(i)$ ;
19     $\rho\{i\}.unif \leftarrow [P(i), \underline{\gamma}_i^o, \bar{\gamma}_i^o]$ ;
20     $cont1 \leftarrow 0$ ;
21    para cada  $j \in J\{i\} \cap L\{i\}$  faça
22       $cont1 \leftarrow cont1 + 1$ ;
23       $\rho\{i\}.delta1(cont1, 1 : 2) \leftarrow [1, x'_j + a_{ij}^o]$ ;
24    fim
25    para  $r = 1 : q$  faça
26      se  $i \in S\{r\}$  então
27         $\rho\{i\}.unif \leftarrow [D(r) \cdot P(i), \underline{\gamma}_i^o, \bar{\gamma}_i^o]$ ;
28        para  $j = 1 : cont1$  faça  $\rho\{i\}.delta1(j, 1) \leftarrow D(r) \cdot \rho\{i\}.delta1(j, 1)$ ;
29        se  $\underline{\gamma}_i^o \leq z_r - c_{ri}^o \leq \bar{\gamma}_i^o$  então
30           $cont1 \leftarrow cont1 + 1$ ;
31           $\rho\{i\}.delta1(cont1, 1 : 2) \leftarrow [P(i), z_r - c_{ri}^o]$ ;
32           $cont2 \leftarrow 0$ ;
33          para cada  $j \in J\{i\} \cap L\{i\}$  faça
34            se  $(x'_j + a_{ij}^o) = (z_r - c_{ri}^o)$  então
35               $cont2 \leftarrow cont2 + 1$ ;
36               $\rho\{i\}.delta2(cont2) \leftarrow z_r - c_{ri}^o$ ;
37            fim
38          fim
39        fim
40      fim
41    fim
42  fim
43 fim

```

---

### 3.2.3 Partículas Inconsistentes

Dada uma medida  $\mathbf{z}_k$ , uma partícula  $\mathbf{x}_{k-1}^i$  é dita inconsistente quando, a partir dela, não é possível se obter  $\mathbf{z}_k$ , ou seja,  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = 0$ ; isto gera uma indeterminação na equação (3.13). A presença de valores determinísticos e variáveis aleatórias nas matrizes  $A$  e  $C$  pode

---

**Algoritmo 3.6:** Geração de Variáveis a partir da Função Majorante
 

---

```

Entrada:  $\rho$ 
Saída:  $Y, \bar{V}, c'$ 
1  , início
2  |  $c' \leftarrow 1;$ 
3  |  $\bar{V} \leftarrow 1;$ 
4  | para  $i = 1 : n$  faça
5  |   |  $\underline{\gamma}^o \leftarrow p\{i\}.unif(2);$ 
6  |   |  $\bar{\gamma}^o \leftarrow p\{i\}.unif(3);$ 
7  |   |  $nd1 \leftarrow$  número de linhas da matriz  $p\{i\}.delta1;$ 
8  |   | se  $\underline{\gamma}^o < \bar{\gamma}^o$  então
9  |   |   |  $CSW(1) \leftarrow p\{i\}.unif(1) \cdot (\bar{\gamma}^o - \underline{\gamma}^o);$ 
10 |   |   | para  $j = 1 : nd1$  faça  $CSW(j+1) \leftarrow CSW(j) + p\{i\}.delta1(j, 1);$ 
11 |   |   |  $c' \leftarrow c' \cdot CSW(nd1+1) \cdot \rho\{i\}.const;$ 
12 |   |   | para  $j = 1 : nd1+1$  faça  $CSW(j) \leftarrow CSW(j)/CSW(nd1+1);$ 
13 |   |   | Obter um número aleatório  $u$  uniformemente distribuído entre 0 e 1:  $u \leftarrow U[0, 1];$ 
14 |   |   |  $j \leftarrow 1;$ 
15 |   |   | enquanto  $u > CSW(j)$  faça  $j \leftarrow j + 1;$ 
16 |   |   | se  $j = 1$  então  $Y(i) \leftarrow U[\underline{\gamma}^o, \bar{\gamma}^o];$ 
17 |   |   | senão  $Y(i) \leftarrow p\{i\}.delta1(j-1, 2);$ 
18 |   |   | fim
19 |   |   | senão se  $\underline{\gamma}^o = \bar{\gamma}^o$  então
20 |   |   | |  $Y(i) \leftarrow \underline{\gamma}^o;$ 
21 |   |   | fim
22 |   |   |  $S \leftarrow p\{i\}.unif(1);$ 
23 |   |   | para  $j = 1 : nd1$  faça
24 |   |   | | se  $Y(i) = p\{i\}.delta1(j, 2)$  então  $S \leftarrow S + M \cdot p\{i\}.delta1(j, 1);$ 
25 |   |   | fim
26 |   |   |  $nd2 \leftarrow |p\{i\}.delta2|;$ 
27 |   |   | para  $j = 1 : nd2$  faça
28 |   |   | | se  $Y(i) = p\{i\}.delta2(j)$  então
29 |   |   | | |  $S \leftarrow S + M^2;$ 
30 |   |   | | |  $c' \leftarrow M;$ 
31 |   |   | | fim
32 |   |   | fim
33 |   |  $\bar{V} \leftarrow \bar{V} \cdot S \cdot \rho\{i\}.const;$ 
34 | fim
35 fim

```

---

proporcionar situações em que se tenha probabilidade 1 de que todas as partículas sejam inconsistentes. O exemplo a seguir ilustra uma situação na qual este problema ocorre:

**Exemplo 3.3.** *Seja um sistema descrito por:*

$$\mathbf{x}_k = \begin{bmatrix} \mu(2, 3) & \epsilon \\ 1 & \epsilon \end{bmatrix} \otimes \mathbf{x}_{k-1}$$

$$\mathbf{z}_k = \begin{bmatrix} \epsilon & 0 \end{bmatrix} \otimes \mathbf{x}_k$$

onde:  $\mathbf{x}_{k-1} = [x'_1 \ x'_2]^T$ ,  $\mathbf{x}_k = [x_1 \ x_2]^T$ ,  $\mathbf{z}_k = [z_1]$  e  $\mu(2, 3)$  representa uma variável aleatória com distribuição uniforme entre 2 e 3 cuja função de densidade de probabilidade será representada neste exemplo por  $\mu_{23}(x)$ .

Neste caso, tem-se que  $\gamma^o = x_2$  e  $\gamma^{no} = x_1$ . Deste modo:

$$\bar{V}_1(\gamma^o) = \delta(z_1 - x_2)\delta(x_2 - x'_1 - 1)$$

Nota-se que se  $z_1 \neq x'_1 + 1$  a função  $\bar{V}_1(\gamma^o)$  terá valor nulo para todo  $\gamma^o$  e, conseqüentemente,  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i) = 0$ . Mas  $x'_1$  é a componente não observada da partícula  $\mathbf{x}_{k-1}^i$  e, portanto, foi amostrada segundo a equação  $\mathbf{x}_{k-1}(1) = [\mu(2,3) \quad \epsilon] \otimes \mathbf{x}_{k-2}$ , ou seja,  $p(\mathbf{x}_{k-1}(1) | \mathbf{x}_{k-2}) = \mu_{2,3}(\mathbf{x}_{k-1}(1) - \mathbf{x}_{k-2}(2))$ .

Deste modo, a probabilidade de que em um conjunto de  $N$  partículas  $\{x_{k-1}^i, i = 1, \dots, N\}$  exista pelo menos uma na qual  $x'_1 = z_1 - 1$  é nula. Portanto, a partir de qualquer partícula pertencente a esse conjunto, é impossível obter um novo conjunto de partículas  $\{x_k^i, i = 1, \dots, N\}$  utilizando a densidade  $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$ .

A existência de partículas inconsistentes leva a uma condição de bloqueio no algoritmo de geração de partículas, uma vez que, dada uma partícula  $\mathbf{x}_{k-1}^i$  inconsistente com uma medida  $\mathbf{z}_k$ , a densidade  $p(\mathbf{x}_k | \mathbf{x}_{k-1}^i, \mathbf{z}_k)$  não pode ser definida. Deste modo, dada uma medida  $\mathbf{z}_k$ , a partícula  $\mathbf{x}_{k-1}^i$  deve ser corrigida a fim de eliminar possíveis inconsistências.

De um modo geral, a inconsistência de uma partícula  $\mathbf{x}_{k-1}^i$  pode ser de dois tipos:

**Tipo 1:** Existe pelo menos um elemento pertencente ao vetor de medidas  $\mathbf{z}_k$  que não é alcançável pelo limite superior  $\bar{\mathbf{x}}_k^i$  definido pela partícula  $\mathbf{x}_{k-1}^i$  e pela medida  $\mathbf{z}_k$  de acordo com a equação (3.42), ou seja,

$$\exists j \in \{1, \dots, q\} : \mathbf{z}_k(j) > \bar{C}(j, 1 : n) \otimes \bar{\mathbf{x}}_k^i.$$

Mas, pelas propriedades da operação de residuação, tem-se que:

$$\begin{aligned} \bar{C}(j, 1 : n) \otimes \bar{\mathbf{x}}_k^i < \mathbf{z}_k(j) &\iff \bar{\mathbf{x}}_k < \bar{C}(j, 1 : n) \oslash \mathbf{z}_k(j) = \mathbf{X}_{cr} \\ &\iff \bar{A} \otimes \mathbf{x}_{k-1}^i < \mathbf{X}_{cr} \\ &\iff \mathbf{x}_{k-1}^i < \bar{A} \oslash \mathbf{X}_{cr} \end{aligned}$$

Ou seja, uma inconsistência do Tipo 1 ocorre quando a partícula  $\mathbf{x}_{k-1}^i$  é estritamente menor que o limitante  $\bar{A} \oslash \mathbf{X}_{cr}$  definido pela medida  $\mathbf{z}_k(j)$ , pela  $j$ -ésima linha da matriz  $C$  e pela matriz  $A$ . Deste modo, para corrigir um inconsistência do Tipo 1 deve-se assegurar que a partícula  $\mathbf{x}_{k-1}^i$  **não** seja **estritamente** menor que este

limitante.

**Tipo 2:** Existe pelo menos um elemento pertencente ao vetor de medidas  $\mathbf{z}_k$  que é menor que o mínimo valor possível de se obter a partir da partícula  $\mathbf{x}_{k-1}^i$ , ou seja,

$$\exists j \in \{1, \dots, q\} : \mathbf{z}_k(j) < \underline{C}(j, 1 : n) \otimes \underline{\mathbf{x}}_k^i,$$

onde  $\underline{\mathbf{x}}_k^i$  é definido pela equação (3.39). Neste caso, como um resultado direto da equação (3.42), tem-se que:

$$\bar{\mathbf{x}}_k^i \leq \mathbf{x}_{max2}^i \Rightarrow \underline{C}(j, 1 : n) \otimes \bar{\mathbf{x}}_k^i \leq \underline{C}(j, 1 : n) \otimes \mathbf{x}_{max2}^i$$

Mas, pela equação (3.44):

$$\underline{C}(j, 1 : n) \otimes \mathbf{x}_{max2}^i \leq \mathbf{z}_k(j)$$

e, como consequência da inconsistência do tipo 2:

$$\mathbf{z}_k(j) < \underline{C}(j, 1 : n) \otimes \underline{\mathbf{x}}_k^i$$

Portanto:

$$\underline{C}(j, 1 : n) \otimes \bar{\mathbf{x}}_k^i < \underline{C}(j, 1 : n) \otimes \underline{\mathbf{x}}_k^i.$$

Sendo assim, uma inconsistência do tipo 2 leva a uma situação na qual existe pelo menos um  $l$  tal que o  $l$ -ésimo elemento do vetor  $\bar{\mathbf{x}}_k^i$  é menor que o  $l$ -ésimo elemento do vetor  $\underline{\mathbf{x}}_k^i$ , o que é um absurdo. Portanto, deve-se corrigir a partícula  $\bar{\mathbf{x}}_{k-1}^i$  a fim de eliminar essa inconsistência.

Deve-se notar ainda que, para que a inconsistência do Tipo 2 não ocorra, a seguinte relação deve ser verdadeira:

$$\underline{C} \otimes \underline{\mathbf{x}}_k^i \leq \mathbf{z}_k$$

ou

$$\begin{aligned} \underline{\mathbf{x}}_k^i \leq \underline{C} \backslash \mathbf{z}_k &\iff \underline{A} \otimes \mathbf{x}_{k-1}^i \leq \underline{C} \backslash \mathbf{z}_k \\ &\iff \mathbf{x}_{k-1}^i \leq \underline{A} \backslash (\underline{C} \backslash \mathbf{z}_k) \end{aligned} \quad (3.48)$$

Deste modo, para corrigir uma inconsistência do Tipo 2, deve-se assegurar que (3.48) seja satisfeita.

Os Algoritmos 3.7 e 3.8 podem ser usados para corrigir as inconsistências do Tipo 1 e 2, respectivamente. Basicamente, o objetivo destes algoritmos de correção de inconsistências é fazer com que os limitantes inferior e superior definidos pela partícula e pela medida sejam tais que:

$$\underline{C} \otimes \underline{\mathbf{x}}_k^i \leq \underline{C} \otimes \bar{\mathbf{x}}_k^i \leq \mathbf{z}_k \leq \bar{C} \otimes \bar{\mathbf{x}}_k^i.$$

---

### Algoritmo 3.7: Correção Inconsistência Tipo 1

---

**Entrada:**  $\mathbf{x}_{k-1}^i, \mathbf{z}_k$   
**Saída:**  $\mathbf{x}_{k-1}, \underline{\mathbf{x}}_k, \bar{\mathbf{x}}_k$

```

1 início
2    $\underline{\mathbf{x}}_k \leftarrow \underline{A} \otimes \mathbf{x}_{k-1}$ ;
3    $\mathbf{x}_{max2} \leftarrow \underline{C} \otimes \mathbf{z}_k$ ;
4    $\mathbf{x}_{max1} \leftarrow \bar{A} \otimes \mathbf{x}_{k-1}$ ;
5    $\bar{\mathbf{x}}_k \leftarrow \min\{\mathbf{x}_{max1}, \mathbf{x}_{max2}\}$ ;
6   para  $m = 1 : q$  faça
7     se  $\mathbf{z}(m) > \bar{C}(m, 1 : n) \otimes \bar{\mathbf{x}}_k^i$  então
8        $\mathbf{X}_{cr} \leftarrow \bar{C}(m, 1 : n) \otimes \mathbf{z}_k(m)$ ;
9        $\mathbf{x}_{k-1} \leftarrow \bar{A} \otimes \mathbf{X}_{cr}$ ;
10       $l \leftarrow \arg \{ \min \{ \underline{\mathbf{x}}_{k-1} - \mathbf{x}_{k-1} \} \}$ ;
11      fim
12   fim
13 fim
```

---



---

### Algoritmo 3.8: Correção Inconsistência Tipo 2

---

**Entrada:**  $\mathbf{x}_{k-1}, \mathbf{z}_k$   
**Saída:**  $\mathbf{x}_{k-1}, \underline{\mathbf{x}}_k, \bar{\mathbf{x}}_k$

```

1 início
2    $\underline{\mathbf{x}}_k \leftarrow \underline{A} \otimes \mathbf{x}_{k-1}^i$ ;
3    $\mathbf{x}_{max1} \leftarrow \bar{A} \otimes \mathbf{x}_{k-1}$ ;
4    $\mathbf{x}_{max2} \leftarrow \underline{C} \otimes \mathbf{z}_k$ ;
5    $\bar{\mathbf{x}}_k \leftarrow \min\{\mathbf{x}_{max1}, \mathbf{x}_{max2}\}$ ;
6    $\hat{\mathbf{x}} \leftarrow \underline{A} \otimes \underline{\mathbf{x}}_k$ ;
7   para  $j = 1 : n$  faça
8     se  $\mathbf{x}_{k-1}(j) > \hat{\mathbf{x}}(j)$  então  $\mathbf{x}_{k-1}(j) \leftarrow \hat{\mathbf{x}}(j)$ ;
9   fim
10   $\underline{\mathbf{x}}_k \leftarrow \underline{A} \otimes \mathbf{x}_{k-1}^i$ ;
11   $\mathbf{x}_{max1} \leftarrow \bar{A} \otimes \mathbf{x}_{k-1}$ ;
12   $\bar{\mathbf{x}}_k \leftarrow \min\{\mathbf{x}_{max1}, \mathbf{x}_{max2}\}$ ;
13 fim
```

---

## 3.2.4 Filtragem de Partículas geradas pelo Método de Aceitação Rejeição

A partir dos algoritmos propostos neste capítulo, pode-se desenvolver o Algoritmo 3.9, que é um algoritmo para filtragem de partículas no qual utiliza-se o método de aceitação

rejeição para gerar partículas distribuídas segundo a densidade de importância ótima.

---

**Algoritmo 3.9:** Filtro de Partículas com o Método de Aceitação Rejeição

---

**Entrada:**  $\{\mathbf{x}_{k-1}^i, \omega_{k-1}^i\}_{i=1}^N, \mathbf{z}_k, \Lambda$   
**Saída:**  $\{\mathbf{x}_k^i\}_{i=1}^N$

```

1 início
2   para  $i = 1 : N$  faça
3     Determinar os limitantes  $\underline{\mathbf{x}}_k^i$  e  $\bar{\mathbf{x}}_k^i$ , utilizando (3.39) e (3.42), respectivamente ;
4     Corrigir inconsistências utilizando os Algoritmos 3.7 e 3.8 ;
5     Definir quais são os estados observados  $\gamma^o$  e os não observados  $\gamma^{no}$ ;
6     Gerar os estados não observados  $\gamma^{no}$  utilizando a equação (3.24);
7     Construir a função majorante  $\bar{V}_1(\gamma^o)$  utilizando o algoritmo 3.5;
8      $\lambda \leftarrow 0$ ;
9     para  $j = 1 : \Lambda$  faça
10      Obter  $\mathbf{Y}$ ,  $c'$  e  $\bar{V}_1(Y)$  utilizando o Algoritmo 3.6 ;
11      Calcular o valor de  $V_1(Y)$  utilizando a equação (3.26) ;
12      Gerar um número aleatório  $\mathbf{U}$ , independente de  $\mathbf{Y}$  e com distribuição uniforme entre 0 e 1;
13      se  $\mathbf{U} \leq \frac{V_1(Y)}{\bar{V}_1(Y)}$  então
14         $\gamma^o \leftarrow \mathbf{Y}$ ;
15         $\lambda \leftarrow \lambda + 1$ ;
16      fim
17    fim
18    Construir o vetor de estados  $\mathbf{x}_k^i$  a partir de  $\gamma^o$  e  $\gamma^{no}$ ;
19    Estimar  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$ :  $\Psi \leftarrow c' \cdot \frac{\lambda}{\Lambda}$ ;
20    Atualizar os pesos:  $\omega_k^i \leftarrow \Psi \omega_{k-1}^i$ ;
21  fim
22  Calcular a soma dos pesos:  $t \leftarrow \sum_{i=1}^N \omega_k^i$ ;
23  para  $i = 1 : N$  faça  $\omega_k^i \leftarrow t^{-1} \omega_k^i$ ;
24  Calcular  $\hat{N}_{eff}$  usando (2.23):  $\hat{N}_{eff} = \frac{1}{\sum_{i=1}^N \omega_k^i}$ ;
25  se  $\hat{N}_{eff} < N_{thr}$  então Reamostrar utilizando o Algoritmo 2.1 ;
26 fim

```

---

### 3.2.5 Caso Exponencial

Embora o Algoritmo 3.9 tenha sido desenvolvido para o caso em que os elementos aleatórios das matrizes  $A$ ,  $B$  e  $C$  possuem distribuições uniformes, a partir de algumas considerações este algoritmo pode ser estendido para o caso em que as variáveis aleatórias são distribuídas exponencialmente.

Primeiramente, deve-se notar que para os estados não observados  $\gamma^{no}$  nenhuma modifi-

cação é necessária, ou seja, estes estados continuam sendo gerados a partir da equação (3.24). Para os estados observados  $\gamma^o$  a equação 3.39 fornece os limitantes inferiores  $\underline{\gamma}^o$  e a equação 3.42 fornece limitantes superiores  $\bar{\gamma}^o$  finitos se  $\mathbf{z}_k$  é finito, pois dado que a matriz  $C_o$  não possui colunas nulas a operação  $\underline{C}_o \setminus \mathbf{z}_k$  tem resultado finito.

De maneira análoga ao caso uniforme pode-se definir  $D_r$  e  $P_i$  (vide equações (3.33) e (3.34)) de maneira que a relação (3.35) seja verdadeira. As funções de densidade de probabilidade das variáveis aleatórias  $c_{rs}$  da matriz  $C_o$  são dadas por:

$$d_{rs}(\tau) = \begin{cases} 0 & \text{se } \tau \leq \underline{c}_{rs} \\ \lambda_{rs} e^{-\lambda_{rs}(\tau - \underline{c}_{rs})} & \text{se } \tau > \underline{c}_{rs} \end{cases}$$

Definindo-se  $D_{rs} = \lambda_{rs}$ , tem-se que  $D_{rs} \geq d_{rs}(\tau)$ .

Analogamente, as funções de densidade de probabilidade das variáveis aleatórias  $a_{ij}$  da matriz  $A_o$  são dadas por:

$$a_{ij}(\tau) = \begin{cases} 0 & \text{se } \tau \leq \underline{a}_{ij} \\ \lambda_{ij} e^{-\lambda_{ij}(\tau - \underline{a}_{ij})} & \text{se } \tau > \underline{a}_{ij} \end{cases}$$

Definindo-se  $P_{ij} = \lambda_{ij}$ , tem-se que  $D_{ij} \geq a_{ij}(\tau)$ .

Deste modo, se  $D_r = \sum_{s \notin S_r} D_{rs}$  e  $P_i = \sum_{j \notin J_i} P_{ij}$ , a função majorante para o caso exponencial também é dada por (3.45). Deve-se notar que, como os limitantes superiores e inferiores desta função são finitos, sua integral é finita. Sendo assim esta função atende às restrições apresentadas da subseção 3.2.1 e pode, portanto, ser utilizada como função majorante.

### 3.3 Contribuições

As principais contribuições dessa dissertação foram apresentadas neste capítulo. A seguir encontra-se uma descrição destas contribuições.

Na seção 3.1, desenvolveu-se um algoritmo de filtragem de partículas para sistemas

Max Plus que utiliza uma densidade de importância sub-ótima. Além das etapas necessárias para um filtro de partículas genérico, observou-se a necessidade de uma etapa adicional de recondicionamento das partículas para evitar a situação em que todos os pesos tornam-se nulos após serem atualizados. O desenvolvimento do algoritmo de recondicionamento das partículas (Algoritmo 3.1) foi fundamental para a implementação do algoritmo de filtragem de partículas, com densidade de importância sub-ótima, para sistemas Max Plus (Algoritmo 3.2).

Na seção 3.2, desenvolveu-se um algoritmo de filtragem de partículas para sistemas Max Plus que utiliza a densidade de importância ótima definida por (2.25). Para gerar partículas distribuídas segundo a densidade de importância ótima, utilizou-se o método de Aceitação Rejeição. O método de Aceitação Rejeição é um método tradicional de geração de variáveis que permite o uso de uma função majorante para gerar amostras distribuídas segundo uma densidade de interesse. Este método é indicado quando é difícil gerar amostras utilizando a densidade de interesse diretamente, mas é possível definir uma função majorante, que satisfaz às condições apresentadas na subseção 3.2.1, a partir da qual seja mais fácil obter amostras. Na subseção 3.2.2, definiu-se uma função majorante, dada por 3.45, para a densidade de importância ótima a partir da qual é possível obter amostras para os estados de maneira independente. Na subseção 3.2.3, apresentou-se o problema de inconsistência entre partículas e medidas, e a importância da correção das partículas inconsistentes. A definição da função majorante e o desenvolvimento dos algoritmos de correção de inconsistências (Algoritmos 3.7 e 3.8) possibilitou a implementação do algoritmo de filtragem de partículas, com densidade de importância ótima, para sistemas Max Plus (Algoritmo 3.9).



## Resultados

Neste capítulo, os filtros desenvolvidos nas seções 3.1 (FP1) e 3.2 (FP2) serão utilizados para estimar os estados de sistemas Max Plus lineares descritos por um sistema de equações do tipo:

$$\mathbf{x}_k = A \otimes \mathbf{x}_{k-1} \oplus B \otimes \mathbf{u}_k \quad (4.1)$$

$$\mathbf{z}_k = C \otimes \mathbf{x}_k \quad (4.2)$$

Alguns elementos das matrizes  $A$ ,  $B$  e  $C$  podem ser variáveis aleatórias com distribuição uniforme ou exponencial. Ao longo deste capítulo os elementos não determinísticos destas matrizes serão notados por  $[a, b]$  cujo significado depende de qual é a distribuição de probabilidade deste elemento:

- **Distribuição uniforme:** o elemento é uniformemente distribuído entre  $a$  e  $b$ ;
- **Distribuição exponencial:** o elemento possui distribuição exponencial, a partir de um valor  $a$ , com média  $b$ .

As funções de densidade de probabilidade para as distribuições uniforme e exponenciais estão disponíveis no Apêndice A

## 4.1 Simulação 1

Nesta simulação, será considerado o sistema de manufatura (flow-shop), apresentado na Figura 4.1, proposto em (Loreto et al. 2010). Este sistema é composto por três máquinas, denotadas por M1, M2 e M3, responsáveis por produzir três tipos de peças, denotadas por P1, P2 e P3. Assume-se que cada máquina processa cada peça somente uma vez. Assume-se ainda que cada peça segue a mesma seqüência de máquinas: M1, M2 e, finalmente, M3, e que a seqüência de peças em cada máquina é sempre a mesma: P1, P2 e P3. Uma peça é transportada entre as máquinas por meio de uma plataforma. Quando uma peça é processada pelas três máquinas, ela é removida da plataforma, que retorna para o ponto inicial para transportar uma nova peça. Nota-se que o tempo de permanência mínima das fichas em alguns lugares não é determinístico, mas segue uma determinada distribuição de probabilidade, por exemplo, o tempo de permanência associado ao lugar entre as transições  $x_1$  e  $x_2$  pertence ao intervalo  $[1, 7]$ , que representa uma certa incerteza quanto ao tempo de *set up* da máquina M1 ao mudar o processamento da peça P1 para P2. Por simplicidade, considera-se que no estado inicial existe um ficha em cada lugar. Fisicamente, isto significa que cada máquina pode processar até três peças simultaneamente.

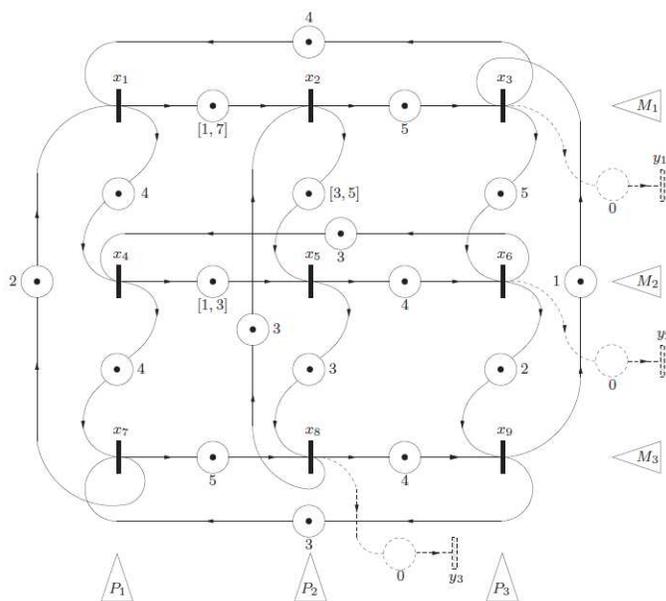


Figura 4.1: Sistema de Manufatura (flow-shop)

As matrizes  $A$  e  $C$  deste sistema autônomo são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ [1, 7] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & [3, 5] & \varepsilon & [1, 3] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & 5 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$$

Nesta simulação, será considerado que os elementos não determinísticos das matrizes  $A$  e  $C$  possuem distribuição uniforme.

Os filtros de partículas desenvolvidos nos Capítulos 3.1 (FP1) e 3.2 (FP2) foram aplicados a este sistema, e os resultados encontrados foram comparados com os resultados do observador de estados para sistemas Max Plus proposto em (Hardouin et al. 2010) (vide Subseção 1.3.4).

Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 1.2$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 15$ .

Tem-se que  $N$  é o número de partículas,  $K_{thr}$  é o limiar utilizado para a etapa de recondiçãoamento do filtro FP1, apresentada no Algoritmo 3.1, e  $\Lambda$  é o parâmetro utilizado para estimar a probabilidade de aceitação do algoritmo de aceitação rejeição de acordo com a equação (3.30).

Na Tabela 4.1, encontram-se estimativas para a média e variância do erro médio quadrático resultantes das estimativas geradas pelos filtros FP1 e FP2 e pelo observador de estados (OBS). Estas estimativas foram calculadas com base nos dados de dez simulações diferentes para cada filtro, e também para o observador, nas quais manteve-se as mesmas sequências de disparos para os estados reais. Nota-se que apenas os estados  $x_2$  e  $x_5$  apresentaram erros não nulos. De fato, analisando a matriz  $A$ , só existem elementos não determinísticos nas linhas correspondentes a estes estados. O erro de estimação não se propaga para os outros estados

a cada iteração porque os estados  $x_3$ ,  $x_6$  e  $x_8$ , que aparecem imediatamente após os estados  $x_2$  e  $x_5$  (Figura 4.1), são diretamente observados.

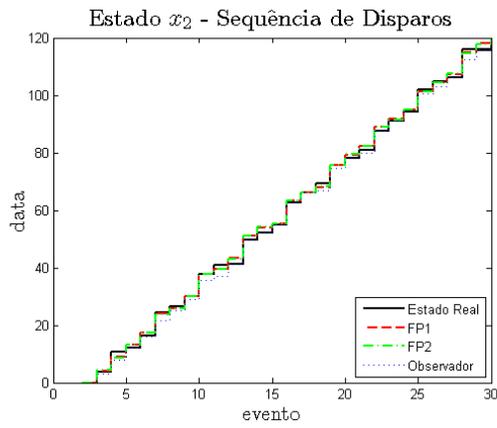
Tabela 4.1: Simulação 1 - Média e variância do erro quadrático médio após dez simulações

Média				Variância			
Est.	FP1	FP2	Obs.	Est.	FP1	FP2	Obs.
$x_1$	0.0000	0.0000	0.0000	$x_1$	0.0000	0.0000	0.0000
$x_2$	1.2530	1.2457	2.0828	$x_2$	0.0004	0.0001	0.0000
$x_3$	0.0000	0.0000	0.0000	$x_3$	0.0000	0.0000	0.0000
$x_4$	0.0000	0.0000	0.0000	$x_4$	0.0000	0.0000	0.0000
$x_5$	0.6677	0.5515	2.8368	$x_5$	0.0003	0.0000	0.0000
$x_6$	0.0000	0.0000	0.0000	$x_6$	0.0000	0.0000	0.0000
$x_7$	0.0000	0.0000	0.0000	$x_7$	0.0000	0.0000	0.0000
$x_8$	0.0000	0.0000	0.0000	$x_8$	0.0000	0.0000	0.0000
$x_9$	0.0000	0.0000	0.0000	$x_9$	0.0000	0.0000	0.0000

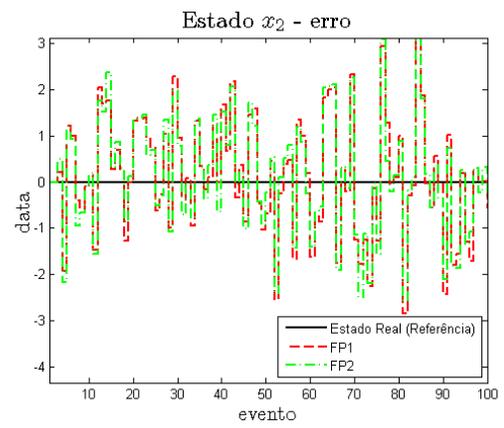
Nas Figuras 4.2(a) e 4.2(c), encontram-se as sequências de disparos para os estados  $x_2$  e  $x_5$ , respectivamente, estimadas pelos filtros de partículas FP1 e FP2 e pelo observador, assim como as sequências de disparos reais para estes estados. Nas Figuras 4.2(b) e 4.2(d), encontram-se os erros de estimação cometidos pelos filtros FP1 e FP2 e pelo observador, para os estados  $x_2$  e  $x_5$ , respectivamente.

Como pode ser verificado na Tabela 4.1 e na Figura 4.2, os resultados encontrados pelos filtros FP1 e FP2 foram muito próximos, sendo que o desempenho do filtro FP2 foi ligeiramente melhor que o do filtro FP1. O filtro FP2 apresentou menor variância do erro quadrático médio que o filtro FP1, e, como esperado, o observador apresentou variância do erro nula, pois não leva em conta as informações estocásticas para a estimação dos estados. Em contrapartida, o observador apresentou maiores médias para o erro quadrático médio. Pela Figura 4.2, pode-se notar que as estimativas para as sequências de disparo geradas pelos filtros estão sempre acima das estimativas encontradas pelo observador, que fornece uma estimativa para o limitante inferior da sequência de disparos dos estados.

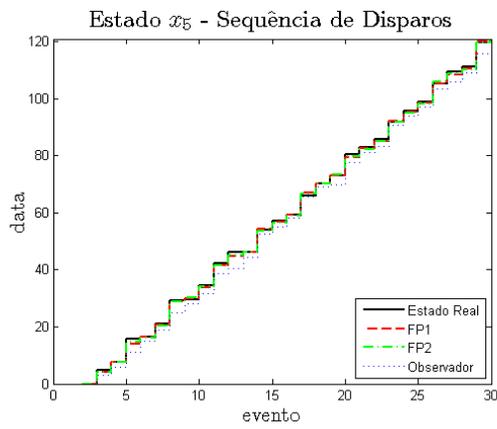
Nota-se ainda, pela Figura 4.2, que, enquanto as curvas de sequências de disparos são monotonicamente crescentes, as curvas apresentadas para os erros absolutos tendem a oscilar em torno de uma média constante. Deste modo, uma estimativa para erros relativos, calculados em relação aos estados do sistema após cada evento, seria degenerada, pois os erros calculados para os eventos mais tardios tenderiam a se anular.



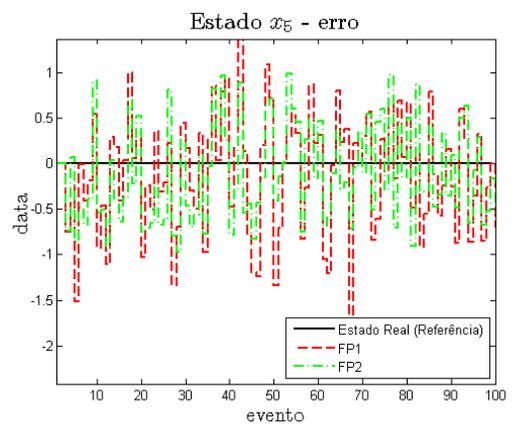
(a)



(b)



(c)



(d)

Figura 4.2: Simulação 1 - Sequência de disparos e erro de estimação para os estados  $x_2$  e  $x_5$

## 4.2 Simulação 2

Nesta simulação, será utilizado o mesmo sistema da simulação anterior, porém será considerado que os elementos aleatórios do sistema possuem distribuição exponencial. Deste modo, as matrizes  $A$  e  $C$  do sistema são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & \varepsilon \\ [1, 4] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & [3, 4] & \varepsilon & [1, 2] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & 5 & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & e & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & e & \varepsilon \end{bmatrix}$$

Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 1.8$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

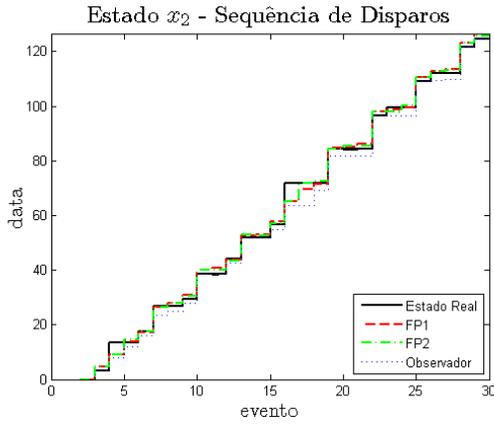
Na Tabela 4.2, encontram-se estimativas para a média e variância do erro médio quadrático resultantes das estimativas geradas pelos filtros FP1 e FP2 e pelo observador de estados (OBS). Assim como na Simulação 1, estas estimativas foram calculadas com base nos dados de dez simulações diferentes para cada filtro, e também para o observador, nas quais manteve-se as mesmas sequências de disparos para os estados reais.

Nas Figuras 4.3(a) e 4.3(c), encontram-se as sequências de disparos para os estados  $x_2$  e  $x_5$ , respectivamente, estimadas pelos filtros de partículas FP1 e FP2 e pelo observador, assim como as sequências de disparos reais para estes estados. Nas Figuras 4.3(b) e 4.3(d), encontram-se os erros de estimação, cometidos pelos filtros FP1 e FP2 e pelo observador, para os estados  $x_2$  e  $x_5$ , respectivamente.

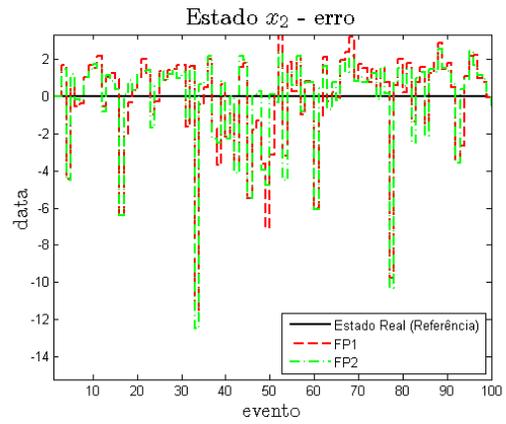
Na Tabela 4.2, pode-se observar que as médias dos erros quadráticos médios encontrados pelos filtros FP1 e FP2 foram menores que as médias encontrados pelo observador. No entanto, a média encontrada pelo filtro FP2 para o estado  $x_5$  foi consideravelmente menor que

Tabela 4.2: Simulação 2 - Média e variância do erro quadrático médio após dez simulações

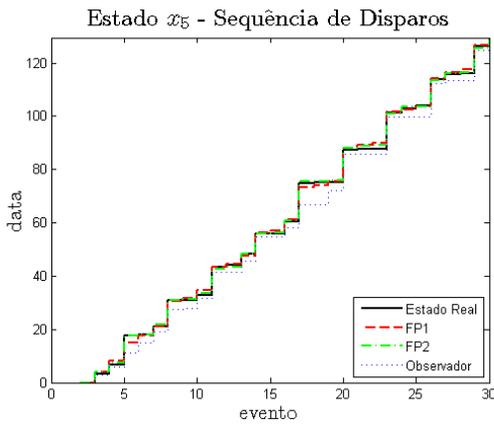
Média				Variância			
Est.	FP1	FP2	Obs.	Est.	FP1	FP2	Obs.
$x_1$	0.0000	0.0000	0.0000	$x_1$	0.0000	0.0000	0.0000
$x_2$	2.6111	2.5102	3.8496	$x_2$	0.0008	0.0004	0.0000
$x_3$	0.0000	0.0000	0.0000	$x_3$	0.0000	0.0000	0.0000
$x_4$	0.0000	0.0000	0.0000	$x_4$	0.0000	0.0000	0.0000
$x_5$	1.5543	0.8242	4.4396	$x_5$	0.0030	0.0001	0.0000
$x_6$	0.0000	0.0000	0.0000	$x_6$	0.0000	0.0000	0.0000
$x_7$	0.0000	0.0000	0.0000	$x_7$	0.0000	0.0000	0.0000
$x_8$	0.0000	0.0000	0.0000	$x_8$	0.0000	0.0000	0.0000
$x_9$	0.0000	0.0000	0.0000	$x_9$	0.0000	0.0000	0.0000



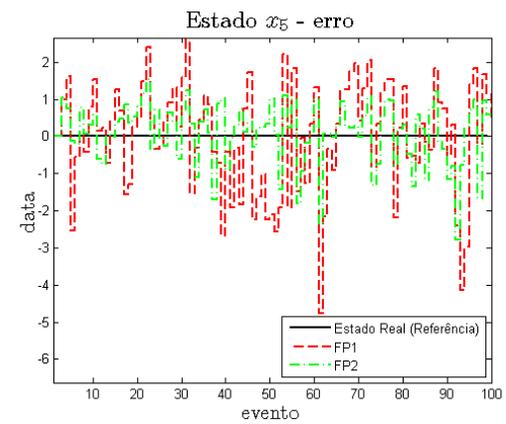
(a)



(b)



(c)



(d)

Figura 4.3: Simulação 2 - Sequência de disparos e erro de estimação para os estados  $x_2$  e  $x_5$

a média encontrado pelo filtro FP1 para este mesmo estado. Observa-se ainda que as médias encontradas nesta simulação, onde considerou-se as distribuições exponenciais, foram maiores do que as médias encontradas na Simulação 1, onde foram consideradas as distribuições uniformes. Novamente, o filtro FP2 apresentou menor variância do erro quadrático médio que o filtro FP1 e o observador apresentou variância nula.

### 4.3 Simulação 3

O sistema considerado nesta simulação é uma variante do sistema utilizado na Simulação 1 na qual os estados  $x_3$ ,  $x_6$  e  $x_8$  não são mais diretamente observados. A saída deste sistema é resultado de uma combinação dos estados  $x_3$ ,  $x_5$ ,  $x_6$  e  $x_8$  sendo que, a relação da saída com estes estados não é totalmente determinística. Foram adicionadas ainda as seguintes restrições quanto ao início de processamento das máquinas:

1. O  $k$ -ésimo início de processamento da peça P1 pela máquina M1 só pode ocorrer  $t_1$  unidades de tempo após o  $(k - 1)$ -ésimo início de processamento da peça P2 pela máquina M2. Sendo  $t_1$  uma variável aleatória uniformemente distribuída entre 3 e 7;
2. O  $k$ -ésimo início de processamento da peça P3 pela máquina M3 só pode ocorrer  $t_2$  unidades de tempo após o  $(k - 1)$ -ésimo início de processamento da peça P3 pela máquina M3. Sendo  $t_2$  uma variável aleatória uniformemente distribuída entre 3 e 7.

Estas restrições são modeladas por alterações nos elementos  $a_{15}$  e  $a_{91}$  na matriz  $A$  utilizada na Simulação 1. As restrições relativas às observações é modelada por uma reescritura da matriz  $C$ .

Deste modo, as matrizes  $A$  e  $C$  que descrevem este sistema são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 4 & \varepsilon & [3, 7] & \varepsilon & 2 & \varepsilon & \varepsilon \\ [1, 7] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & [3, 5] & \varepsilon & [1, 3] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & 5 & \varepsilon & \varepsilon \\ [3, 7] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \varepsilon & \varepsilon & 0 & \varepsilon & [0, 2] & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & [2, 6] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \end{bmatrix}$$

Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 2.4$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

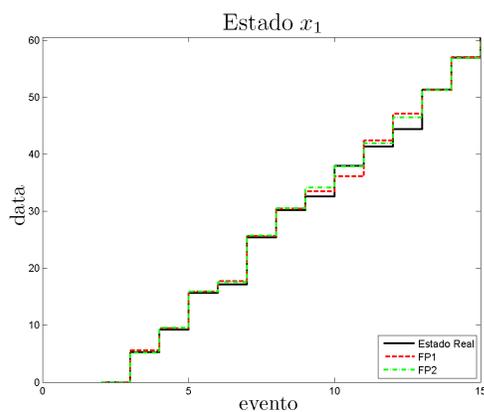
As Figuras 4.4 e 4.5 apresentam as sequências de disparos estimadas pelos filtros de partículas FP1 e FP2 para todos os estados do sistema assim como suas sequências de disparos reais. Na Tabela 4.3 pode-se observar os erros de estimação para o filtro FP1 e para o filtro FP2.

Nota-se que os erros médio quadrático e máximo das estimações realizadas pelo filtro FP2 foram, em geral, menores que os erros das estimações realizadas pelo filtro FP1. A exceção ocorreu no estado  $x_7$ , para o qual o erro médio quadrático do filtro FP1 foi levemente menor que o erro médio quadrático do filtro FP2. Por outro lado, o filtro FP2 conseguiu obter estimativas com erro zero para o estado  $x_6$ . Além disso, assim como para o estado  $x_6$ , as estimativas para os demais estados observados  $x_3$ ,  $x_5$  e  $x_8$  obtidas pelo filtro FP2 foram notavelmente mais precisas que as estimativas obtidas para estes estados pelo filtro FP1.

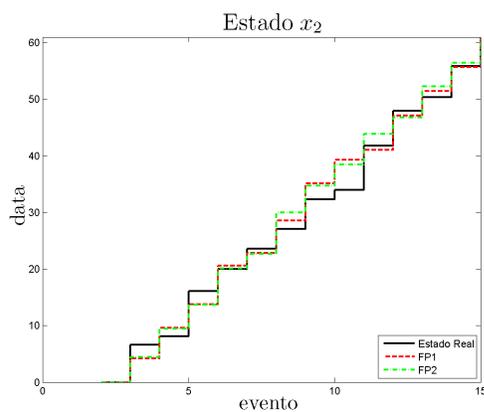
Deve-se notar que no filtro FP2 os estados observados  $\gamma^o$  (vide equação (3.23)) são amostrados segundo a densidade  $p(\mathbf{z}_k|\gamma^o)p(\gamma^o|\mathbf{x}_{k-1}^i)$  e no filtro FP1 todos os estados são amostrados segundo  $p(\mathbf{x}_k|\mathbf{x}_{k-1}^i)$ , ou seja, no filtro FP1 a informação da medida  $\mathbf{x}_k$  não é utilizada na etapa de propagação das partículas, sendo essa informação utilizada apenas na etapa de atualização dos pesos, já no filtro FP2 a informação da medida  $\mathbf{z}_k$  é considerada tanto na etapa de propagação das partículas quanto na etapa de atualização dos pesos.

Tabela 4.3: Simulação 3 - Comparando os Filtros FP1 e FP2

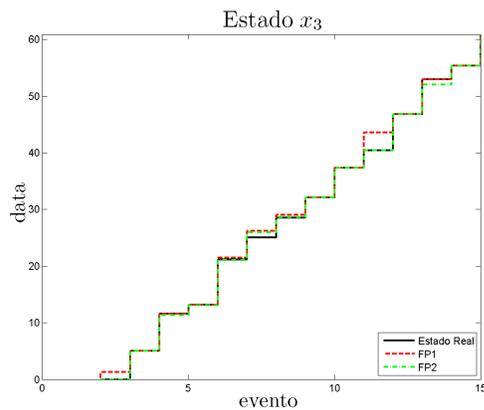
Erro quadrático médio			Erro Máximo		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	0.9599	0.8073	$x_1$	2.7307	2.0518
$x_2$	1.8712	1.6254	$x_2$	5.3264	4.4564
$x_3$	1.0062	0.4379	$x_3$	4.4845	1.3788
$x_4$	1.0009	0.8048	$x_4$	2.6371	2.0171
$x_5$	1.3093	0.5653	$x_5$	4.8706	1.9260
$x_6$	0.0125	0.0000	$x_6$	0.0741	0.0000
$x_7$	1.2167	1.2299	$x_7$	3.0629	3.0418
$x_8$	1.8600	1.2415	$x_8$	3.7817	2.9543
$x_9$	1.5712	1.2817	$x_9$	3.4461	3.3342



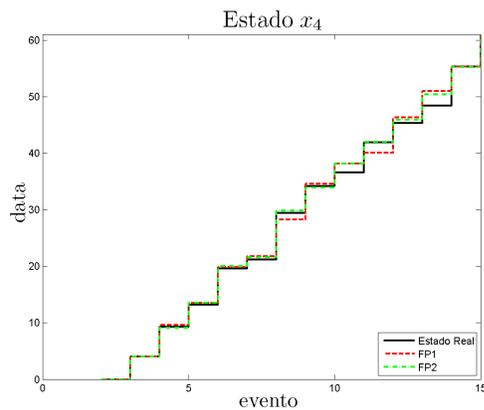
(a)



(b)

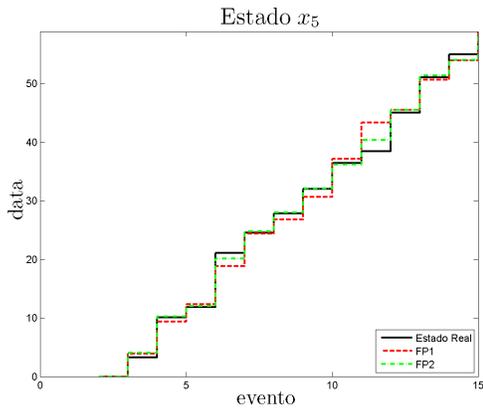


(c)

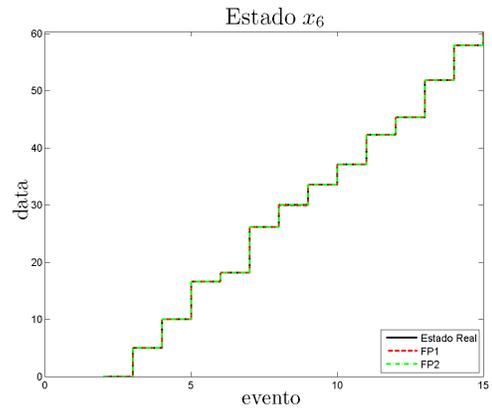


(d)

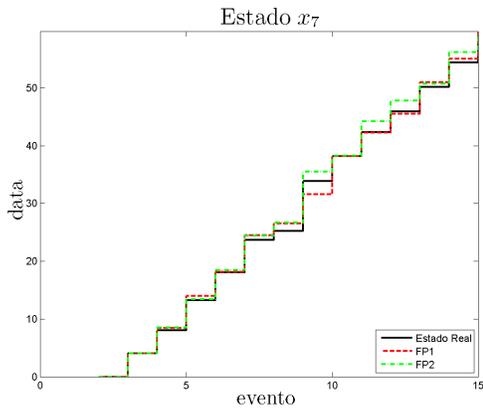
Figura 4.4: Simulação 3 - Sequências de disparo dos estados  $x_1$  a  $x_4$ .



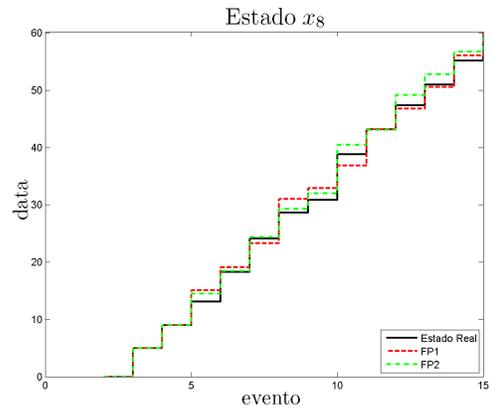
(a)



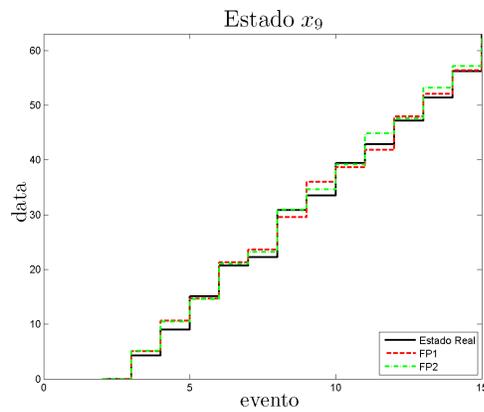
(b)



(c)



(d)



(e)

Figura 4.5: Simulação 3 - Sequências de disparo dos estados  $x_5$  a  $x_9$ .

Nas Figuras 4.6(a) e 4.6(c), encontra-se a variância dos pesos, calculada após cada evento, para os filtros FP1 e FP2. Nas Figuras 4.6(b) e 4.6(d), encontra-se o tamanho efetivo das amostras, após cada evento, para os filtros FP1 e FP2. A partir destas figuras, pode-se

notar que, em geral, a taxa de crescimento da variância dos pesos para o filtro FP2 é menor que a taxa de crescimento para o filtro FP1. Este fato faz com que, para o filtro FP2, o tamanho efetivo das amostras,  $N_{eff}$ , encontre-se abaixo do limiar,  $N_{thr}$ , um número menor de vezes comparando-se com o filtro FP1. Conseqüentemente, a reamostragem é necessária um número maior de vezes no filtro FP1. Este resultado é justificado pela otimalidade da densidade de importância utilizada no filtro FP2, já que, como discutido no Capítulo 2, a escolha da densidade de importância ótima é uma das estratégias para superar o fenômeno da degeneração.

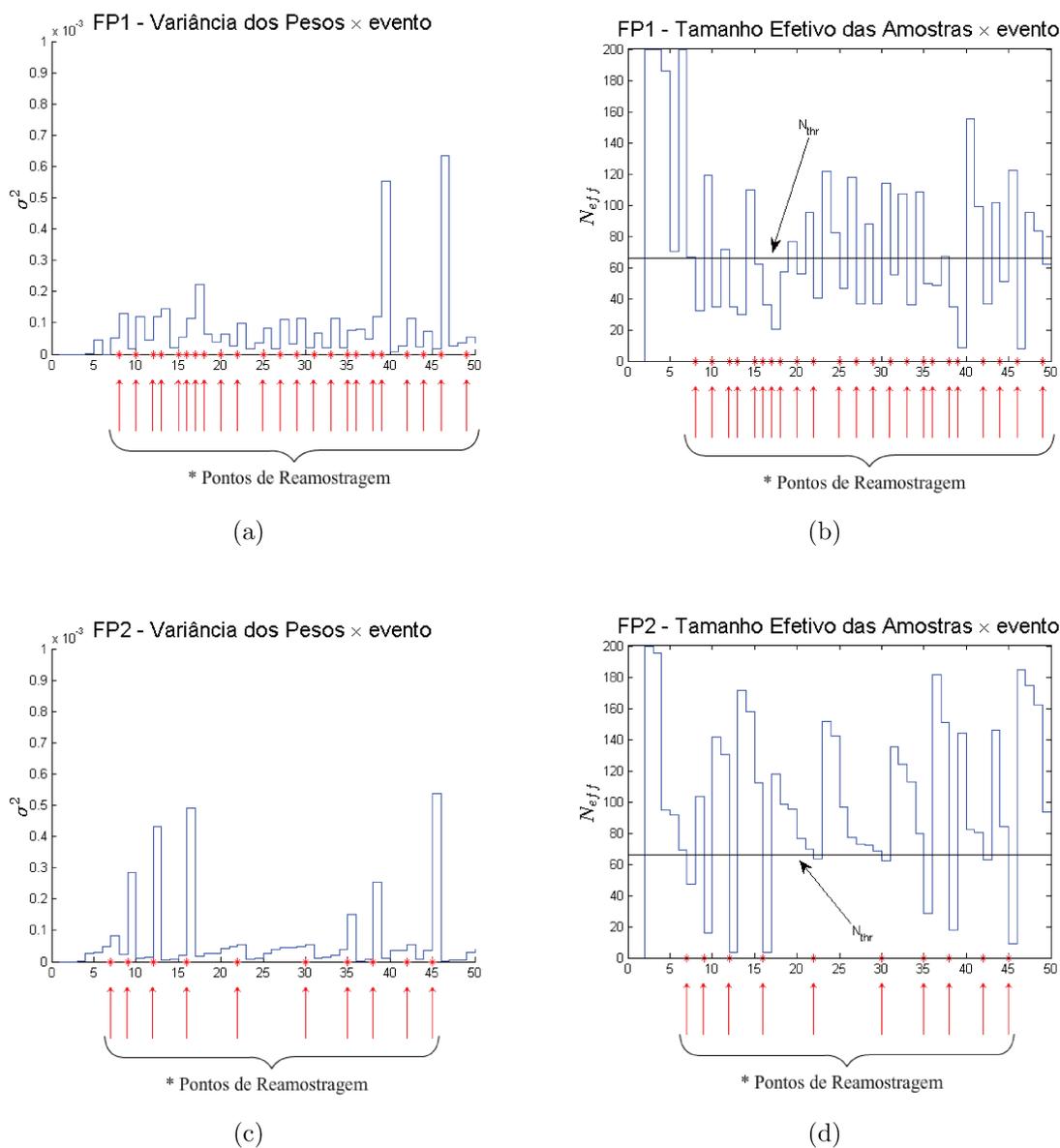


Figura 4.6: Simulação 3 - Variância dos pesos e tamanho efetivo das amostras

## 4.4 Simulação 4

Nesta simulação, considerou-se o mesmo sistema da Simulação 3, mas com distribuições exponenciais para os elementos não determinísticos das matrizes  $A$  e  $C$ . Neste caso, estas matrizes são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & 4 & \varepsilon & [3,5] & \varepsilon & 2 & \varepsilon & \varepsilon \\ [1,4] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon \\ \varepsilon & 5 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 1 \\ 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & [3,4] & \varepsilon & [1,2] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & 5 & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & 4 & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & 3 & \varepsilon & 5 & \varepsilon & \varepsilon \\ [3,5] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 2 & \varepsilon & 4 & \varepsilon \end{bmatrix}, \mathbf{C} = \begin{bmatrix} \varepsilon & \varepsilon & 0 & \varepsilon & [0,1] & \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon & \varepsilon & \varepsilon \\ \varepsilon & \varepsilon & [2,4] & \varepsilon & \varepsilon & \varepsilon & \varepsilon & 0 & \varepsilon \end{bmatrix}$$

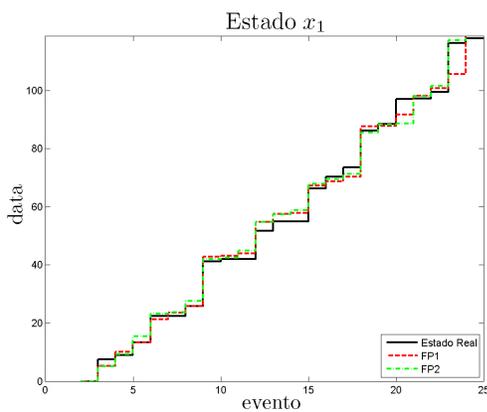
Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 3.5$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

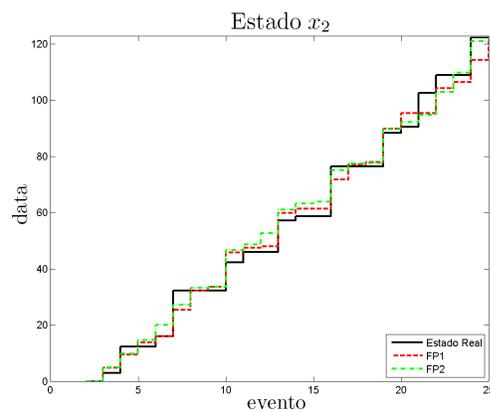
Pela Tabela 4.4, nota-se que as estimativas obtidas pelo filtro FP2 apresentaram um erro menor que as estimativas obtidas pelo filtro FP1. Novamente, o filtro FP2 gerou estimativas para o estado  $x_6$  com erro zero, e, para o estado  $x_7$ , a diferença entre os erros de estimação dos dois filtros foi muito pequena. Nas Figuras 4.7 e 4.8, encontram-se as sequências de disparo para os estados do sistema considerado nesta simulação.

Tabela 4.4: Simulação 4 - Comparando os Filtros FP1 e FP2

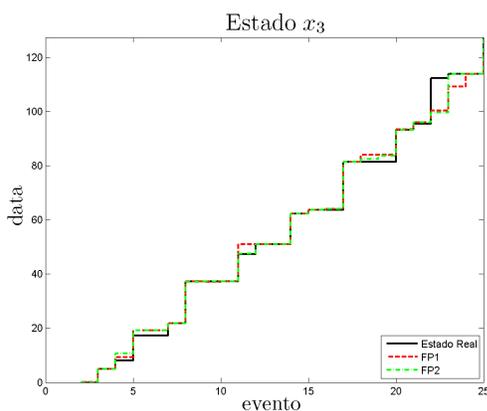
Erro quadrático médio			Erro Máximo		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	3.8783	2.6274	$x_1$	13.9446	8.5261
$x_2$	4.7767	4.3707	$x_2$	16.8985	16.9533
$x_3$	3.9777	2.4577	$x_3$	15.8857	17.1172
$x_4$	3.7922	3.5580	$x_4$	13.9446	10.1017
$x_5$	3.8923	1.7531	$x_5$	14.9430	9.9441
$x_6$	3.6243	0.0000	$x_6$	15.1956	0.0000
$x_7$	3.7879	3.7372	$x_7$	15.9486	14.0033
$x_8$	3.8275	2.5618	$x_8$	15.8585	7.2664
$x_9$	4.0659	4.0672	$x_9$	14.8386	15.1842



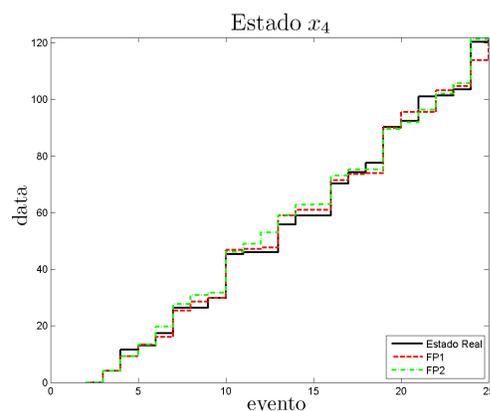
(a)



(b)

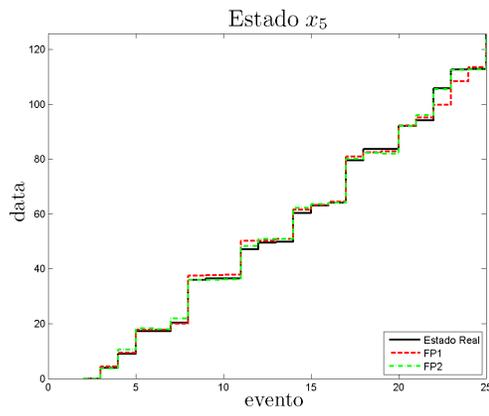


(c)

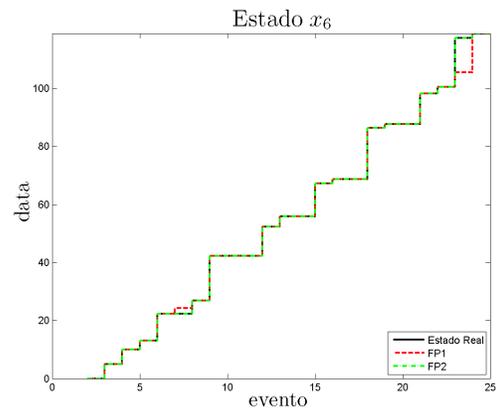


(d)

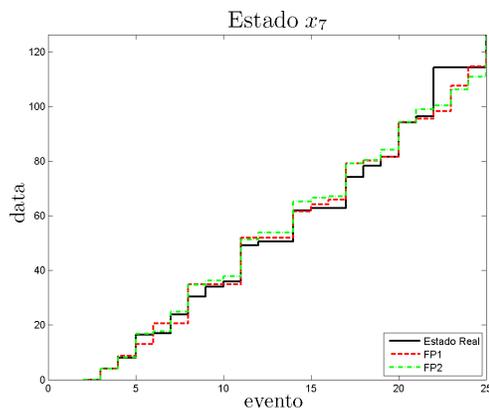
Figura 4.7: Simulação 4 - Sequências de disparo dos estados  $x_1$  a  $x_4$



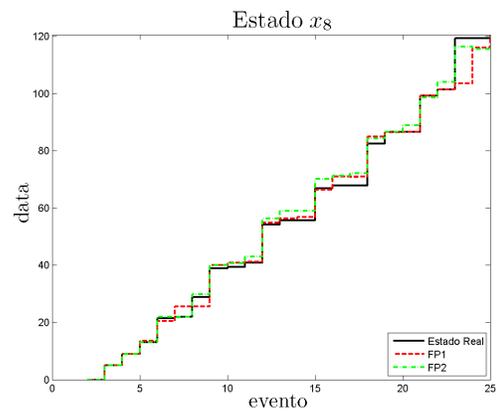
(a)



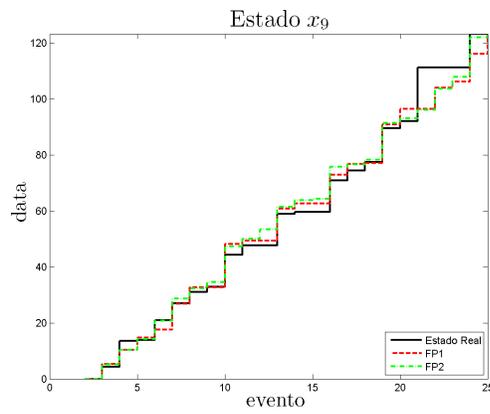
(b)



(c)



(d)



(e)

Figura 4.8: Simulação 4 - Sequências de disparo dos estados  $x_5$  a  $x_9$ .

## 4.5 Simulação 5

Para esta simulação, considere o GET da Figura 4.9 proposto em (Cohen 2001), no qual alguns elementos determinísticos foram substituídos por variáveis aleatórias.

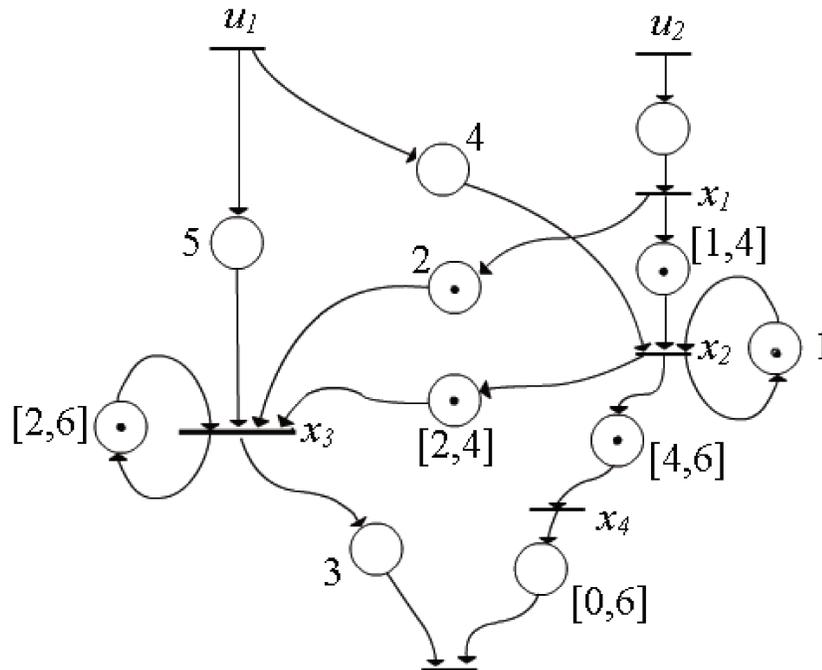


Figura 4.9: Um GET com duas entradas e uma saída

As matrizes que descrevem este GET são:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ [1,4] & 1 & \varepsilon & \varepsilon \\ 2 & [2,4] & [2,6] & \varepsilon \\ \varepsilon & [4,6] & \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \varepsilon & 0 \\ 4 & \varepsilon \\ 5 & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} \varepsilon \\ \varepsilon \\ 3 \\ [0,6] \end{bmatrix}$$

Deve-se notar que o GET da Figura 4.9 possui duas entradas, diferentemente dos sistemas considerados nas simulações anteriores, que eram autônomos. Nesta simulação, considerou-se que os disparos das transições de entrada são processos de Poisson, deste modo os tempos inter-transições possuem distribuição exponencial. Sendo assim, as entradas  $u_1$  e  $u_2$  foram geradas a partir de distribuições exponenciais com médias 5 e 8 respectivamente. Na Figura 4.10, pode-se observar as sequências de disparo para as entradas  $u_1$  e  $u_2$ . Deve-se

notar que, no processo de filtragem, supõe-se que as sequências de disparo das transições de entrada são conhecidas.

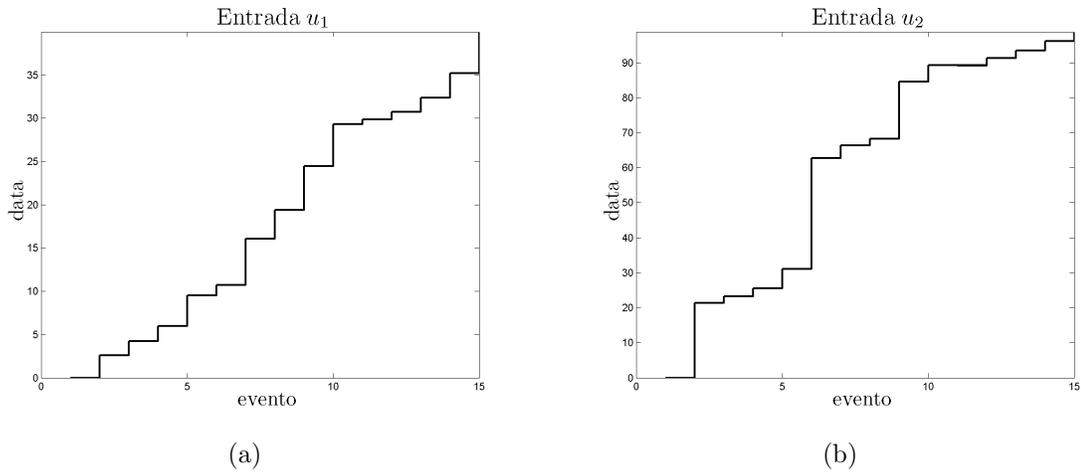


Figura 4.10: Simulação 5 - Sequências de disparo das transições de entrada

Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 0.8$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

Pela Tabela 4.5, percebe-se que, para todos os estados, as estimativas do filtro FP2 apresentaram erros médios quadráticos menores que as estimativas do filtro FP1, sendo que a maior diferença ocorre para o estado  $x_3$ . Para o estado  $x_4$ , o erro máximo cometido pelo filtro FP2 foi ligeiramente maior que o erro cometido pelo filtro FP1. Na Figura 4.11 encontram-se as sequências de disparo reais e estimadas para o estado  $x_3$ ,  $x_4$  e  $x_5$ . Nota-se que, de fato, sequência de disparos estimada pelo filtro FP2 é mais próxima da sequência de disparo real.

Tabela 4.5: Simulação 5 - Comparando os Filtros FP1 e FP2

Erro quadrático médio			Erro Máximo		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	0.0000	0.0000	$x_1$	0.0000	0.0000
$x_2$	0.8470	0.8432	$x_2$	1.5209	1.6058
$x_3$	0.9935	0.5431	$x_3$	4.3968	1.7980
$x_4$	0.9243	0.9023	$x_4$	2.2428	2.2552

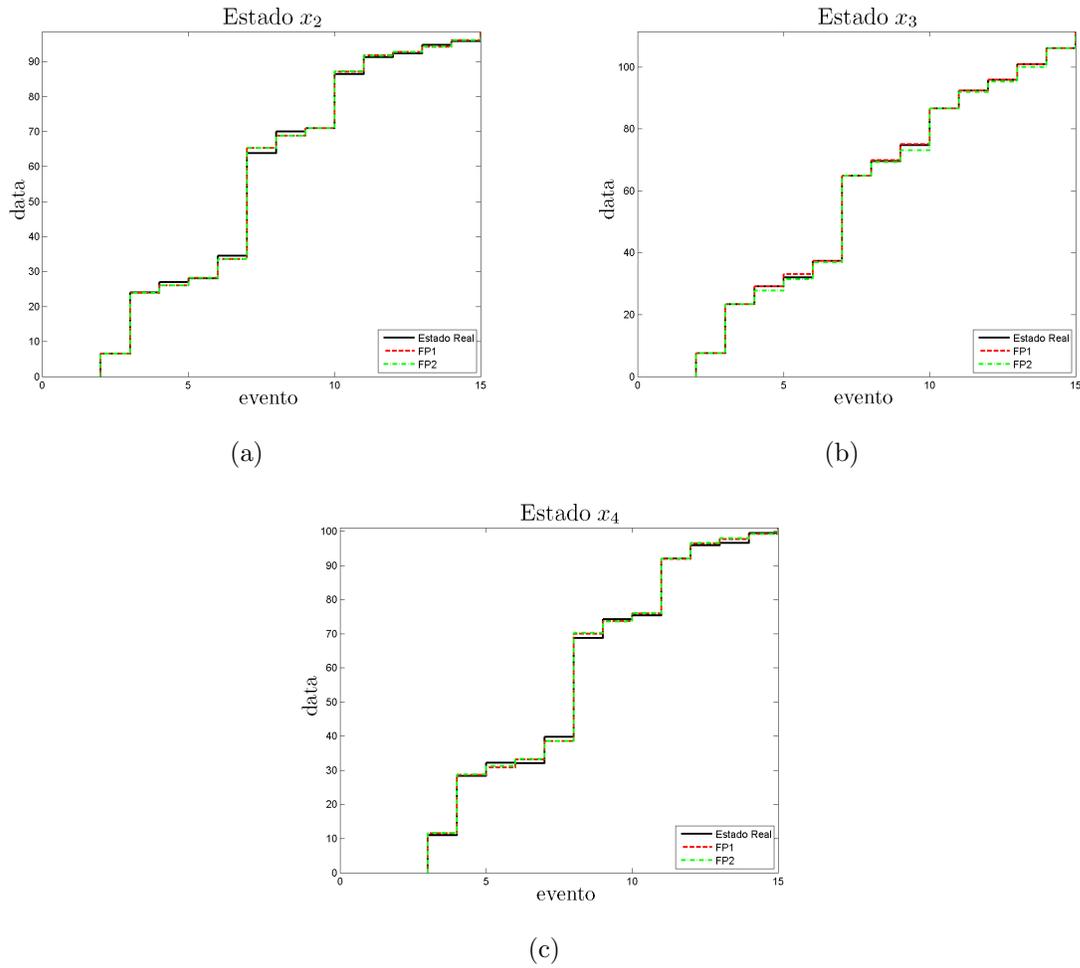


Figura 4.11: Simulação 5 - Sequências de disparo dos estados  $x_2$ ,  $x_3$  e  $x_4$

## 4.6 Simulação 6

Considerando o sistema da Simulação 5 e assumindo que seus elementos aleatórios possuem distribuição exponencial, as matrizes que descrevem o sistema desta simulação são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ [1, 2.5] & 1 & \varepsilon & \varepsilon \\ 2 & [2, 3] & [2, 4] & \varepsilon \\ \varepsilon & [4, 5] & \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \varepsilon & 0 \\ 4 & \varepsilon \\ 5 & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} \varepsilon \\ \varepsilon \\ 3 \\ [0, 3] \end{bmatrix}$$

Assim como na simulação anterior, as entradas  $u_1$  e  $u_2$  foram geradas a partir de distribuições exponenciais com médias 5 e 8 respectivamente. Na Figura 4.12, pode-se observar

as sequências de disparo para as entradas  $u_1$  e  $u_2$ .

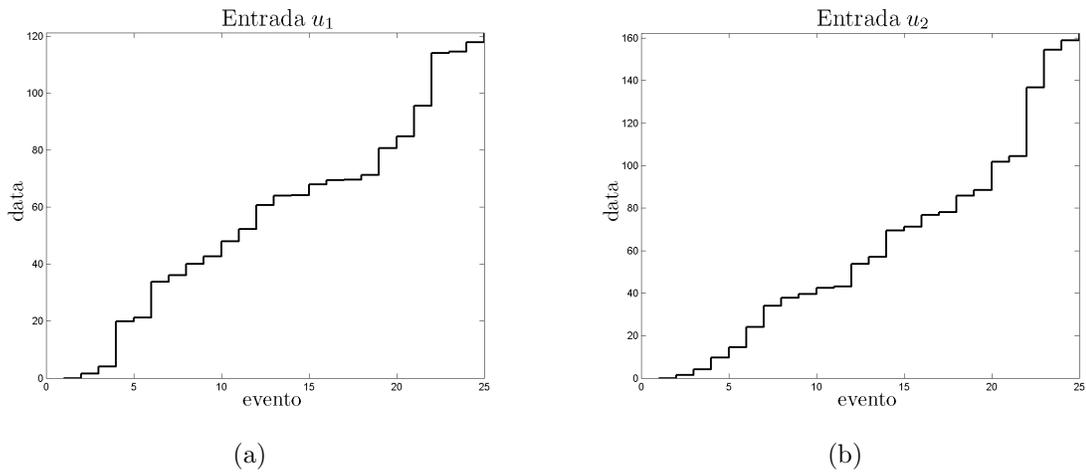


Figura 4.12: Simulação 6 - Sequências de disparo das transições de entrada

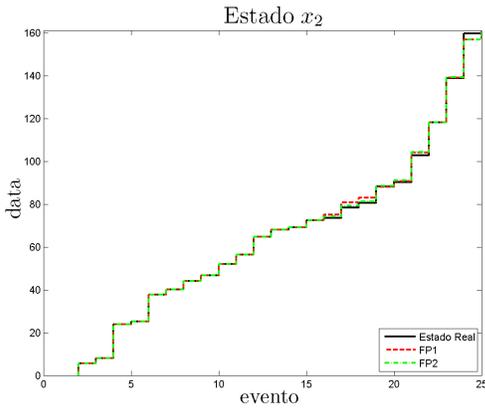
Os parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 1.5$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

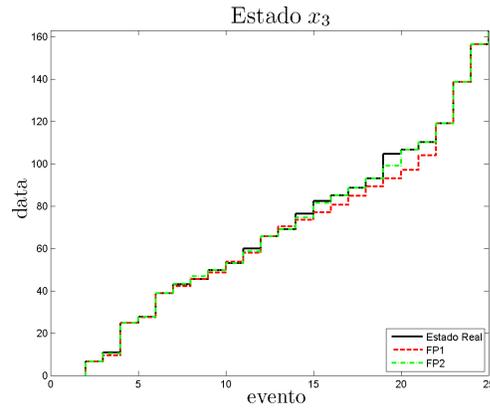
Na Tabela 4.6, observa-se que, para todos os estados, as estimativas do filtro FP2 apresentaram erros quadrático médio e máximo menores que as estimativas do filtro FP1, sendo que, assim como na simulação anterior, a maior diferença ocorre para o estado  $x_3$ . Na Figura 4.13 encontra-se as sequências de disparo reais e estimadas para os estados  $x_3$ ,  $x_4$  e  $x_5$ . Nota-se a sequência de disparos estimada pelo filtro FP2 é mais próxima da curva real que a sequência disparos estimada pelo filtro FP1. Nota-se ainda que as estimativas do filtro FP1 para o estado  $x_3$  obteve um maior desvio dos valores reais no intervalo entre as transições 14 e 22 sendo que em seguida as estimativas para este estado voltam a ficar mais próximas da curva real.

Tabela 4.6: Comparando os Filtros FP1 e FP2

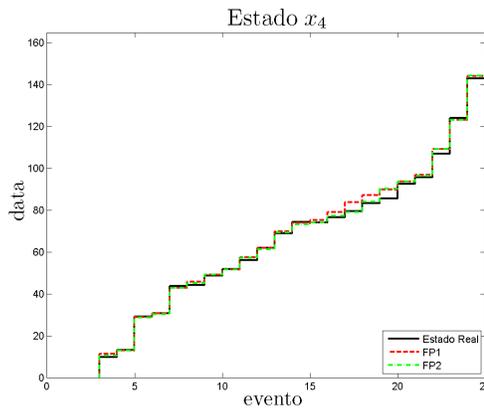
Erro quadrático médio			Erro Máximo		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	0.0000	0.0000	$x_1$	0.0000	0.0000
$x_2$	1.3887	1.3128	$x_2$	5.0796	4.8166
$x_3$	2.9448	1.3828	$x_3$	8.4801	4.5781
$x_4$	2.2531	2.2209	$x_4$	7.8695	5.7245



(a)



(b)



(c)

Figura 4.13: Simulação 6 - Sequências de disparo dos estados  $x_2$ ,  $x_3$  e  $x_4$

## 4.7 Simulação 7

Nesta simulação, considera-se o caso em que todos os elementos não nulos das matrizes  $A$  e  $C$  do sistema utilizado nas simulações 4 e 5 são variáveis aleatórias com distribuição uniforme. Deste modo, as matrizes que descrevem o sistema considerado nesta simulação

são:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ [1, 4] & [1, 5] & \varepsilon & \varepsilon \\ [2, 6] & [2, 4] & [2, 6] & \varepsilon \\ \varepsilon & [0, 2] & \varepsilon & \varepsilon \end{bmatrix}, \mathbf{B} = \begin{bmatrix} \varepsilon & 0 \\ 4 & \varepsilon \\ 5 & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, \mathbf{C}^T = \begin{bmatrix} \varepsilon \\ \varepsilon \\ [3, 5] \\ [0, 6] \end{bmatrix}$$

As entradas  $u_1$  e  $u_2$  foram geradas a partir de distribuições exponenciais com médias 5 e 8, respectivamente, e os demais parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 0$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

Pela Tabela 4.7 e pela Figura 4.14, observa-se que as estimativas obtidas pelos filtros FP1 e FP2 foram bem próximas de seus valores reais. Comparando-se os dois filtros, o filtro FP2 apresentou resultados mais precisos para os estados  $x_2$  e  $x_4$  e, para o estado,  $x_3$  a diferença entre os erros de estimação médio quadráticos foram pequenos, sendo assim, pode-se considerar que, para este estado, o desempenho dos filtros FP1 e FP2 foram equivalentes.

Tabela 4.7: Comparando os Filtros FP1 e FP2

Erro quadrático médio			Erro Máximo		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	0.0000	0.0000	$x_1$	0.0000	0.0000
$x_2$	1.4543	1.3669	$x_2$	5.2994	4.7991
$x_3$	0.3962	0.3975	$x_3$	0.9975	0.9366
$x_4$	1.6080	1.5007	$x_4$	5.6015	5.0931

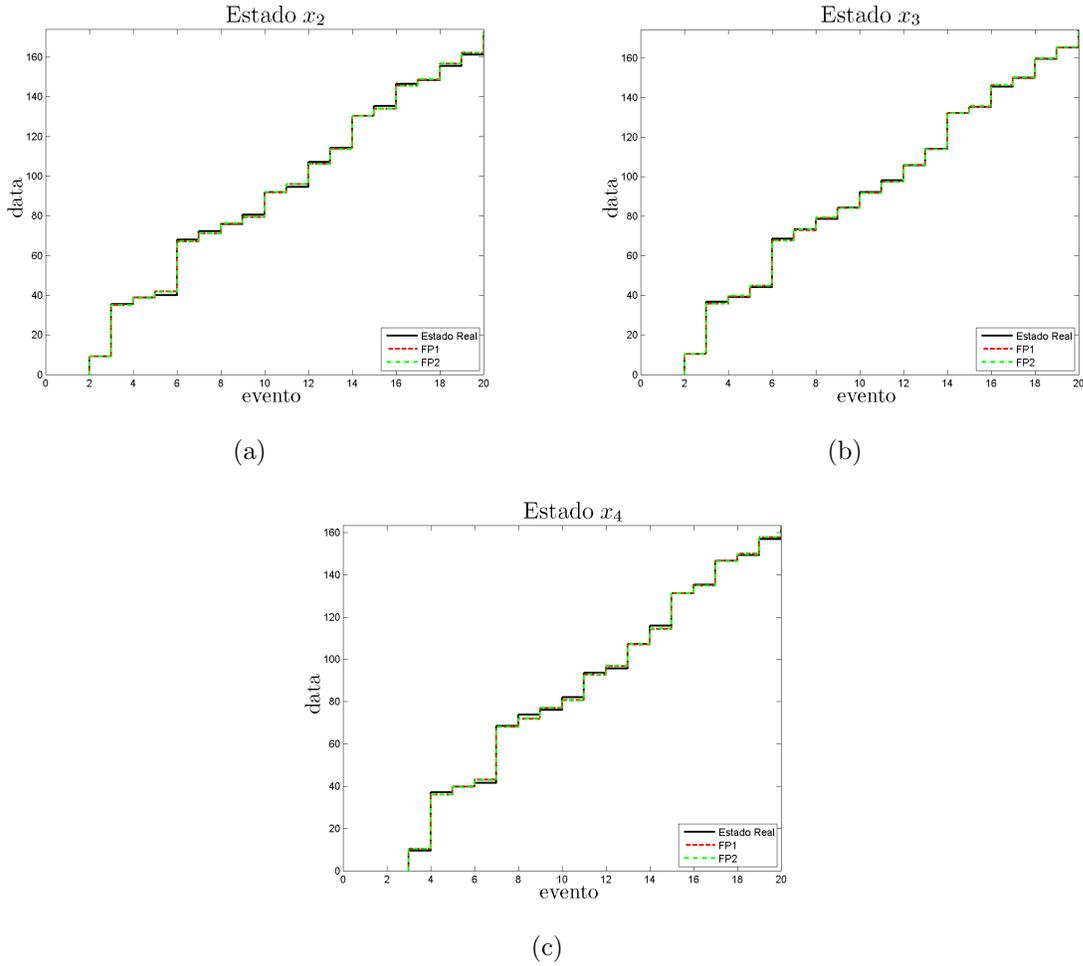


Figura 4.14: Simulação 7 - Sequências de disparo dos estados  $x_2$ ,  $x_3$  e  $x_4$

## 4.8 Simulação 8

Considerando agora o caso em que as variáveis aleatórias do sistema utilizado na simulação anterior possuem distribuição exponencial, as matrizes do sistema são dadas por:

$$\mathbf{A} = \begin{bmatrix} \varepsilon & \varepsilon & \varepsilon & \varepsilon \\ [1, 2.5] & [1, 3] & \varepsilon & \varepsilon \\ [2, 4] & [2, 3] & [2, 4] & \varepsilon \\ \varepsilon & [0, 1] & \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \varepsilon & 0 \\ 4 & \varepsilon \\ 5 & \varepsilon \\ \varepsilon & \varepsilon \end{bmatrix}, \quad \mathbf{C}^T = \begin{bmatrix} \varepsilon \\ \varepsilon \\ [3, 4] \\ [0, 3] \end{bmatrix}$$

Novamente, as entradas  $u_1$  e  $u_2$  foram geradas a partir de distribuições exponenciais

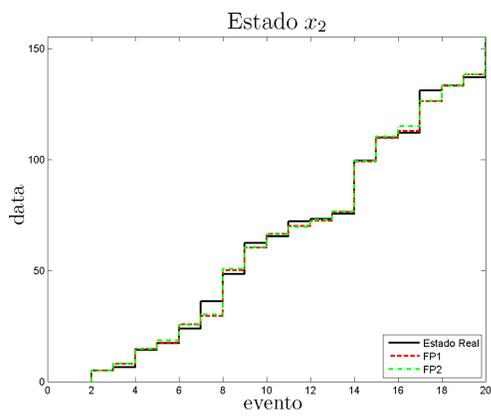
com médias 5 e 8, respectivamente, e os demais parâmetros desta simulação foram:

- **FP1:**  $N = 200$  e  $K_{thr} = 0$ ;
- **FP2:**  $N = 200$  e  $\Lambda = 50$ .

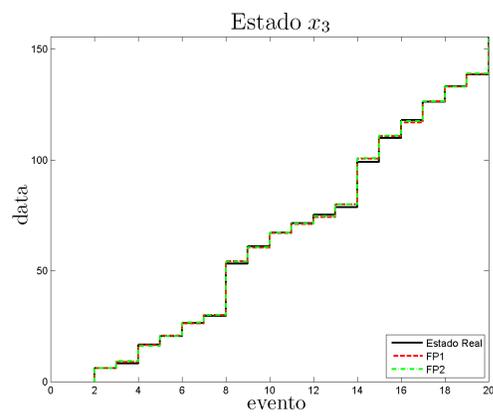
Assim como nas outras simulações, o filtro FP2 apresentou um desempenho melhor que o filtro FP1. Comparando as Tabelas 4.7 e 4.8 percebe-se que, para o caso exponencial, os erros de estimação são maiores que para o caso uniforme. Pela Figura 4.15 observa-se a grande proximidade entre as estimativas e os valores reais para as sequências de disparo de cada estado.

Tabela 4.8: Simulação 8 - Comparando os Filtros FP1 e FP2

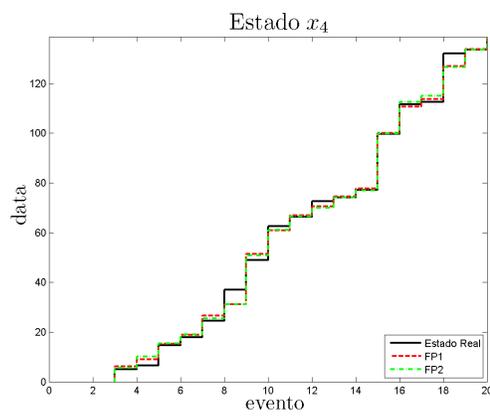
<b>Erro quadrático médio</b>			<b>Erro Máximo</b>		
Est.	FP1	FP2	Est.	FP1	FP2
$x_1$	0.0000	0.0000	$x_1$	0.0000	0.0000
$x_2$	1.9198	1.8771	$x_2$	6.4262	5.7549
$x_3$	1.1034	1.0543	$x_3$	4.9302	3.7997
$x_4$	2.4393	2.3637	$x_4$	8.2207	8.2582



(a)



(b)



(c)

Figura 4.15: Simulação 8 - Sequências de disparo dos estados  $x_2$ ,  $x_3$  e  $x_4$

## Conclusão

Neste trabalho, desenvolveram-se dois algoritmos para filtragem de partículas em sistemas max-plus. Como pode-se observar no Capítulo 3, a síntese do algoritmo com densidade de importância ótima (Filtro FP2) é bem mais complexa que a síntese do primeiro algoritmo (Filtro FP1), no qual utiliza-se uma densidade de importância sub-ótima. Uma descrição mais detalhada das principais contribuições deste trabalho, pode ser encontrada na Seção 3.3.

O algoritmo do filtro FP1 é composto por quatro passos chave que são:

1. a etapa de propagação das partículas;
2. a etapa de atualização dos pesos;
3. a etapa de reamostragem;
4. a etapa de recondicionamento das partículas.

A escolha sub-ótima da densidade de importância proporcionou simplicidade ao algoritmo, já que as partículas podem ser propagadas para  $k$  usando diretamente a equação dinâmica de transição dos estados definida por (3.1) a partir de um conjunto de partículas existentes em  $k - 1$ . A etapa de recondicionamento das partículas possibilitou a aplicação do filtro a sistemas onde os estados podem ser relacionados de maneira determinística com a saída, entretanto, esta etapa pode causar pequenas variações na distribuição das partículas  $\mathbf{x}_{k-1}^i$ . Esta variação pode ser maior ou menor dependendo da escolha do limiar  $K_{thr}$  utilizado

na etapa de condicionamento apresentada no Algoritmo 3.1. A etapa de reamostragem é de grande importância, pois evita a degeneração do algoritmo.

As etapas principais do algoritmo do filtro FP2 são:

1. a etapa de propagação das partículas;
2. a etapa de atualização dos pesos;
3. a etapa de reamostragem;
4. a etapa de correção de inconsistências.

No algoritmo do filtro FP2, a etapa de propagação das partículas é feita utilizando-se o método de aceitação rejeição para gerar amostras distribuídas segundo a densidade de importância ótima definida pela equação 2.25. Neste algoritmo, para cada uma das  $N$  partículas, deve-se aplicar o método de aceitação rejeição  $\Lambda$  vezes (vide equação (3.30)) a fim de se obter uma estimativa para a densidade  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$ , a qual é utilizada na etapa de atualização dos pesos das partículas segundo (2.27). Isto faz com que o tempo de estimação do Filtro FP2 seja consideravelmente maior que o tempo de estimação do filtro FP1. A etapa de correção de inconsistências é de grande importância pois evita uma situação de bloqueio do algoritmo na qual é impossível realizar a etapa de propagação das partículas.

Como pode-se perceber ao longo das simulações apresentadas no Capítulo 4, o filtro FP1 é bem mais sensível aos seus parâmetros ( $K_{thr}$  e  $N$ ) que o filtro FP2 ( $\Lambda$  e  $N$ ). Para cada sistema, alguns testes devem ser feitos até que se encontre um valor de  $K_{thr}$  que diminua os erros de estimação do filtro FP1. Por outro lado, os parâmetros no filtro FP2 não variaram ao para os diferentes sistemas simulados, deve-se notar entretanto que quanto maior for o valor de  $\Lambda$  mais refinada será a estimativa de  $p(\mathbf{z}_k | \mathbf{x}_{k-1}^i)$  e, conseqüentemente, as estimativas para os estados do sistema.

Os resultados apresentados mostraram que o filtro FP2 obteve estimativas para os estados do sistema com desempenho igual em alguns estados e melhor em outros, sendo que, as maiores diferenças de desempenho se dão para os estados observados dos sistemas. Em suma, o filtro FP2 apresentou um desempenho melhor que o filtro FP1 ao custo de um tempo de estimação consideravelmente maior.

## 5.1 Trabalhos Futuros

Não se vislumbra a partir deste trabalho a obtenção de soluções analíticas recorrentes similares ao algoritmo proposto por Kalman para os sistemas lineares com ruído gaussiano. Contudo, nos parece uma linha de trabalho promissora a análise de possibilidades de filtragem não baseadas na abordagem Monte Carlo, mas similares à abordagem do filtro UKF (*Unscented Kalman Filter*).

Além disso, parecem viáveis algoritmos de controle baseados em projetos de controladores de realimentação de estado nos quais a informação sobre os estados é fornecida por um mecanismo de filtragem.



# Bibliografía

- Alspach, D. & Sorenson, H. (1972). Nonlinear bayesian estimation using gaussian sum approximations, *Automatic Control, IEEE Transactions on* **17**(4): 439–448.
- Anderson, B. & Moore, J. (2005). *Optimal Filtering*, Dover Books on Electrical Engineering Series, Dover Publications.
- Arulampalam, M., Maskell, S., Gordon, N. & Clapp, T. (2002). A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking, *Signal Processing, IEEE Transactions on* **50**(2): 174–188.
- Baccelli, F. (1992). *Synchronization and Linearity: An Algebra for Discrete Event Systems*, Wiley series in probability and mathematical statistics: Probability and mathematical statistics, Wiley.
- Beneš, V. E. (1981). Exact finite dimensional filters for certain diffusions with nonlinear drift, *Stochastics* **5**.
- Cassandras, C. & Lafortune, S. (2007). *Introduction to Discrete Event Systems*, SpringerLink Engineering, Springer.
- Cândido, R. M. F., Santos-Mendes, R., Hardouin, L. & Maia, C. (2013). Particle filter for max-plus systems, *European Control Conference, ECC 2013*.
- Cohen, G. (2001). *Análisis y control de sistemas de eventos discretos: de redes de Petri temporizadas al álgebra*, Vol. 29 of *Cuadernos del Instituto de Matemática “Beppo Levi”*, Facultad de Ciencias Exactas e Ingeniería.
- Cohen, G., Gaubert, S. & Quadrat, J.-P. (1999). Max-plus algebra and system theory: Where we are and where to go now, *Annual Reviews in Control*, pp. 207–219.
- Cohen, G., Moller, P., Quadrat, J.-P. & Viot, M. (1989). Algebraic tools for the performance evaluation of discrete event systems, *Proceedings of the IEEE* **77**(1): 39–85.
- Cottenceau, B. (1999). *Contribution à la commande de systèmes à événements discrets: synthèse de correcteurs pour les graphes d’événements temporisés dans les dioïdes*, PhD thesis.

- Cottenceau, B., Hardouin, L., Boimond, J.-L. & Ferrier, J.-L. (2001). Model reference control for timed event graphs in dioids, *Automatica* **37**(9): 1451 – 1458.
- Daum, F. (1986). Exact finite-dimensional nonlinear filters, *Automatic Control, IEEE Transactions on*, Vol. 31, pp. 616 – 622.
- De Schutter, B. & van den Boom, T. (2001). Model predictive control for max-plus-linear discrete event systems, *Automatica* **37**(7): 1049–1056.
- Doucet, A., Godsill, S. & Andrieu, C. (2000). On sequential monte carlo sampling methods for bayesian filtering, *Statistics and Computing* **10**.
- Gilks, W. R. & Berzuini, C. (2001). Following a moving target-monte carlo inference for dynamic bayesian models, *Journal of the Royal Statistical Society* **63**: 127–146.
- Giua, A. & Seatzu, C. (2002). Observability of place/transition nets, *Automatic Control, IEEE Transactions on* **47**(9): 1424–1437.
- Gordon, N., Salmond, D. & Smith, A. (1993). Novel approach to nonlinear/non-gaussian bayesian state estimation, *Radar and Signal Processing, IEE Proceedings F* **140**(2): 107–113.
- Gustafsson, F. (2010). Particle filter theory and practice with positioning applications, *Aerospace and Electronic Systems Magazine, IEEE* **25**(7): 53–82.
- Hardouin, L., Maia, C. A., Cottenceau, B. & Lhommeau, M. (2010). Observer design for (max,+) linear systems, *IEEE Trans. on Automatic Control* **55 - 2**: 538 – 543.
- Kitagawa, G. (1996). Monte Carlo Filter and Smoother for Non-Gaussian Nonlinear State Space Models, *Journal of Computational and Graphical Statistics* **5**(1): 1–25.
- Kong, A., Liu, J. S. & Wong, W. H. (1994). Sequential imputations and bayesian missing data problems, *Journal of the American Statistical Association* **89**(425).
- Kramer, S. C. & Sorenson, H. W. (1988). Recursive bayesian estimation using piece-wise constant approximations, *Automatica* **24**(6): 789–801.
- Law, A. & Kelton, W. (2000). *Simulation modeling and analysis*, McGraw-Hill series in industrial engineering and management science, McGraw-Hill.
- Lhommeau, M., Hardouin, L. & Cottenceau, B. (2003). Optimal control for (max,+)-linear systems in the presence of disturbances, in L. Benvenuti, A. Santis & L. Farina (eds), *Positive Systems*, Vol. 294 of *Lecture Notes in Control and Information Science*, Springer Berlin Heidelberg, pp. 47–54.
- Lhommeau, M., Hardouin, L., Ferrier, J.-L. & Ouerghi, I. (2005). Interval analysis in dioid: Application to robust open-loop control for timed event graphs, *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 7744–7749.

- Liu, J. S. & Chen, R. (1998). Sequential monte carlo methods for dynamic systems, *Journal of the American Statistical Association* **93**: 1032–1044.
- Loreto, M. D., Gaubert, S., Katz, R. D. & Loiseau, J. (2010). Duality between invariant spaces for max-plus linear discrete event systems, *SIAM J. on Control and Optimization* .
- Maia, C. A., Lüders, R., Mendes, R. S. & Hardouin, L. (2005). Estratégias de controle por modelo de referência de sistemas a eventos discretos max-plus lineares, *Sba Controle & Automação* **16**(3): 263–278.
- Maia, C., Hardouin, L., Santos-Mendes, R. & Cottenceau, B. (2003). Optimal closed-loop control of timed event graphs in dioids, *Automatic Control, IEEE Transactions on* **48**(12): 2284–2287.
- Maia, C., Hardouin, L., Santos-Mendes, R. & Cottenceau, B. (2005). On the model reference control for max-plus linear systems, *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC '05. 44th IEEE Conference on*, pp. 7799–7803.
- Menguy, E., Boimond, J.-L., Hardouin, L. & Ferrier, J.-L. (2000). Just-in-time control of timed event graphs: update of reference input, presence of uncontrollable input, *Automatic Control, IEEE Transactions on* **45**(11): 2155–2159.
- Musso, C., Oudjane, N. & Gland, F. L. (2001). *Improving Regularised Particle Filters*, Springer New York.
- Necoara, I., De Schutter, B., van den Boom, T. & Hellendoorn, H. (2007). Stable model predictive control for constrained max-plus-linear systems, *Discrete Event Dynamic Systems: Theory and Applications* **17**(3): 329–354.
- Ouerghi, I. & Hardouin, L. (2006). A precompensator synthesis for P-temporal event graphs., *Positive Systems* pp. 391–398.
- Quadrat, J. (1999). *Semi-anillos en matematica aplicada*, Vol. 28 of *Cuadernos del Instituto de Matemática “Beppo Levi”*, Universidad Nacional de Rosario, Facultad de Ciencias Exactas, Ingeniería y Agrimensura.
- Ristic, B., Arulampalam, S. & Gordon, N. (2004). *Beyond the Kalman Filter*, Artech House.
- Silva, D. F. e., Mendes, R. S., Hardouin, L., Maia, C. A. & Cottenceau, B. (2011). Filtragem estocástica aplicada a sistemas max-plus lineares, *Anais do X Simpósio Brasileiro de Automação Inteligente*, pp. 1388–1393.
- van den Boom, T. & De Schutter, B. (2004). Model predictive control for perturbed max-plus-linear systems: A stochastic approach, *International Journal of Control* **77**(3): 302–309.
- van der Merwe, R., Doucet, A., de Freitas, N. & Wan, E. (2000). The unscented particle filter, *Technical Report CUED/F-INFENG/TR 380* .



## Cálculo da função de verossimilhança

Para o cálculo da função de verossimilhança, considera-se a equação max-plus  $\mathbf{z} = C \otimes \mathbf{x}$ , sendo  $\mathbf{x} \in \mathbb{R}^{n \times 1}$ ,  $\mathbf{z} \in \mathbb{R}^{q \times 1}$  e  $C \in \mathbb{R}^{q \times n}$ . Supõe-se que os elementos da matriz  $C$ , notados por  $c_{ij}$ , sejam variáveis aleatórias independentes uniformemente distribuídas entre  $\underline{c}_{ij}$  e  $\overline{c}_{ij}$  ou exponencialmente distribuídas a partir de um valor  $\underline{c}_{ij}$  e com média  $\underline{c}_{ij} + \frac{1}{\lambda}$ . As funções de probabilidade acumulada e de densidade de probabilidade de uma variável aleatória com distribuição uniforme são dadas respectivamente por:

$$F_{ij}(\tau) = \begin{cases} 0 & \text{se } \tau \leq \underline{c}_{ij} \\ \frac{\tau - \underline{c}_{ij}}{\overline{c}_{ij} - \underline{c}_{ij}} & \text{se } \underline{c}_{ij} < \tau \leq \overline{c}_{ij} \\ 1 & \text{se } \tau > \overline{c}_{ij} \end{cases}$$

$$p_{ij}(\tau) = \begin{cases} \frac{1}{\overline{c}_{ij} - \underline{c}_{ij}} & \text{se } \underline{c}_{ij} < \tau \leq \overline{c}_{ij} \\ 0 & \text{se } \tau \leq \underline{c}_{ij} \text{ ou } \tau > \overline{c}_{ij} \end{cases}$$

e, para uma variável aleatória com distribuição exponencial, as funções de probabilidade acumulada e de densidade de probabilidade são respectivamente:

$$F_{ij}(\tau) = \begin{cases} 0 & \text{se } \tau \leq \underline{c}_{ij} \\ 1 - \lambda e^{-\lambda(\tau - \underline{c}_{ij})} & \text{se } \tau > \underline{c}_{ij} \end{cases}$$

$$p_{ij}(\tau) = \begin{cases} 0 & \text{se } \tau \leq \underline{c}_{ij} \\ \lambda e^{-\lambda(\tau - \underline{c}_{ij})} & \text{se } \tau > \underline{c}_{ij} \end{cases}$$

A densidade de probabilidade da variável  $\mathbf{z}$  condicionada a  $\mathbf{x}$ , ou seja,  $p_Z(t|\mathbf{x})$ , onde  $t = [t_1 \cdots t_q]^T \in \mathbb{R}^{q \times 1}$  é determinada a seguir (Silva et al. 2011).

Devido a independência dos elementos de  $C$  segue-se que:

$$P[\mathbf{z} \leq t] = P[z_1 \leq t_1 \ \& \ \dots \ \& \ z_q \leq t_q] = \prod_{i=1}^q P[z_i \leq t_i]$$

Porém:

$$\begin{aligned} P[z_i \leq t_i] &= P[\max_j(c_{ij} + x_j) \leq t_i] \\ &= P[c_{i1} \leq t_i - x_1 \ \& \ \dots \ \& \ c_{in} \leq t_i - x_n] \\ &= \prod_{j=1}^n F_{ij}(t_i - x_j) \end{aligned}$$

Portanto a função de probabilidade acumulada conjunta da variável aleatória  $\mathbf{z}$ , condicionada por  $\mathbf{x}$ , é dada por:

$$F_Z(t|\mathbf{x}) = \prod_{i=1}^q \prod_{j=1}^n F_{ij}(t_i - x_j)$$

Derivando-se sucessivamente em relação a  $t_1 \cdots t_q$  obtém-se a densidade de probabilidade procurada:

$$\begin{aligned} p_Z(t|\mathbf{x}) &= \prod_{i=1}^q \frac{\partial}{\partial t_i} \left( \prod_{j=1}^n F_{ij}(t_i - x_j) \right) \\ &= \prod_{i=1}^q \left( \sum_{j=1}^n \frac{\partial}{\partial t_i} (F_{ij}(t_i - x_j)) \prod_{\substack{k=1, \\ k \neq j}}^n F_{ik}(t_i - x_k) \right) \\ &= \prod_{i=1}^q \left( \sum_{j=1}^n p_{ij}(t_i - x_j) \prod_{\substack{k=1, \\ k \neq j}}^n F_{ik}(t_i - x_k) \right) \end{aligned}$$

Se o vetor  $\mathbf{z}$  for conhecido, esta função é chamada de função de verossimilhança de  $\mathbf{x}$ . Neste caso, substituindo-se  $t$  pelo valor conhecido de  $\mathbf{z}$ , obtém-se:

$$V(\mathbf{x}, \mathbf{z}) = \prod_{i=1}^q \left( \sum_{j=1}^n p_{ij}(z_i - x_j) \prod_{\substack{k=1, \\ k \neq j}}^n F_{ik}(z_i - x_k) \right) \quad (\text{A.1})$$

## Demonstração do Método de Aceitação Rejeição

Deseja-se provar que:

$$P[X \leq x] = P\left[Y \leq x \mid U \leq \frac{f(Y)}{g(Y)}\right] \quad (\text{B.1})$$

Relembrando que:

1.  $g(Y) \geq f(Y) \forall Y$ ;
2.  $\int_{-\infty}^{\infty} g(x)dx = c$ ;
3.  $Y$  é uma variável aleatória gerada a partir da densidade  $h(y) = \frac{g(Y)}{c}$ ;
4.  $U \leq \frac{f(Y)}{g(Y)}$  é a condição necessária para que a variável aleatória  $Y$  seja aceita pelo algoritmo;
5.  $U$  e  $Y$  são variáveis aleatórias independentes.

Por definição, tem-se que:

$$P[Y \leq x \mid U \leq f(Y)/g(Y)] = \frac{P[Y \leq x \text{ e } U \leq f(Y)/g(Y)]}{P[U \leq f(Y)/g(Y)]} \quad (\text{B.2})$$

onde  $P[U \leq f(Y)/g(Y)]$  é a probabilidade de aceitação de  $Y$ .

Como  $U$  e  $Y$  são independentes, sua densidade de probabilidade conjunta é:

$$p_{UY}(u, y) = p_U(u) \cdot p_Y(y) \quad (\text{B.3})$$

Além disso,  $p_Y(y) = h(y)$  e:

$$p_U(u) = \begin{cases} 1, & \text{se } 0 < u \leq 1 \\ 0, & \text{se } u \leq 0 \text{ ou } u > 0 \end{cases} \quad (\text{B.4})$$

Deste modo:

$$p_{UY}(u, y) = \begin{cases} h(y), & \text{se } 0 < u \leq 1 \\ 0, & \text{se } u \leq 0 \text{ ou } u > 0 \end{cases} \quad (\text{B.5})$$

Considerando agora a curva  $u(y) = \frac{f(y)}{g(y)}$ , tem-se por construção que  $0 \leq u(y) \leq 1, \forall y$ .

Portanto:

$$\begin{aligned} P[Y \leq x \text{ e } U \leq f(y)/g(y)] &= \int_{-\infty}^x \int_{-\infty}^{\frac{f(y)}{g(y)}} p_{UY}(u, y) du dy \\ &= \int_{-\infty}^x \int_0^{\frac{f(y)}{g(y)}} h(y) du dy \\ &= \int_{-\infty}^x h(y) \frac{f(y)}{g(y)} dy \\ &= \int_{-\infty}^x h(y) \frac{f(y)}{c \cdot h(y)} dy \\ &= \frac{1}{c} \int_{-\infty}^x f(y) dy \end{aligned} \quad (\text{B.6})$$

A probabilidade de aceitação de  $Y$  é:

$$\begin{aligned} P[U \leq f(Y)/g(Y)] &= P[Y \leq \infty \text{ e } U \leq f(y)/g(y)] \\ &= \frac{1}{c} \int_{-\infty}^{\infty} f(y) dy \\ &= \frac{1}{c} \end{aligned} \quad (\text{B.7})$$

Substituindo (B.6) e (B.7) em (B.2):

$$\begin{aligned} P[Y \leq x | U \leq f(Y)/g(Y)] &= \frac{\frac{1}{c} \int_{-\infty}^x f(y) dy}{\frac{1}{c}} \\ &= \int_{-\infty}^x f(y) dy \\ &= P[X \leq x] \end{aligned} \tag{B.8}$$

Provou-se portanto que:

$$P[X \leq x] = P\left[Y \leq x \mid U \leq \frac{f(Y)}{g(Y)}\right]$$