



OTIMIZAÇÃO DINÂMICA USANDO APROXIMAÇÃO COM BASE WAVELETS ADAPTATIVA

Lizandro de Sousa Santos

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia Química, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia Química.

Orientadores: Evaristo Chalbaud Biscaia Jr.
Argimiro Resende Secchi

Rio de Janeiro
Julho de 2014

OTIMIZAÇÃO DINÂMICA USANDO APROXIMAÇÃO COM BASE WAVELETS
ADAPTATIVA

Lizandro de Sousa Santos

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ
COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA
UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS
REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM
CIÊNCIAS EM ENGENHARIA QUÍMICA.

Examinada por:

Prof. Evaristo Chalbaud Biscaia Jr., D.Sc.

Prof. Argimiro Resende Secchi, D.Sc.

Prof. José Carlos Costa da Silva Pinto, D.Sc.

Prof. Maurício Bezerra de Souza Jr., D.Sc.

Prof. Valéria Viana Murata, D.Sc.

Dr. Euclides Almeida Neto, D.Sc.

RIO DE JANEIRO, RJ - BRASIL

JULHO DE 2014

Santos, Lizandro de Sousa

Otimização Dinâmica usando Aproximação com Base Wavelets Adaptativa/ Lizandro de Sousa Santos. – Rio de Janeiro: UFRJ/COPPE, 2014.

X, 197 p.: il.; 29,7 cm.

Orientadores: Evaristo Chalbaud Biscaia Jr.

Argimiro Resende Secchi

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia Química, 2014.

Referências Bibliográficas: p. 183-197.

1. Controle Ótimo. 2. Adaptação de Malhas. 3. Programação Não Linear. I. Biscaia Jr., Evaristo Chalbaud *et al.* II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia Química. III. Título.

Dedico esta Tese aos meus pais.

Agradecimentos

Aos meus pais, por investirem em minha educação e sempre me incentivarem desde cedo a seguir o caminho dos estudos;

À minha querida esposa, Marie Julie, por me apoiar, compreender e incentivar durante todos esses anos;

À minha família (especialmente a meus irmãos e mãe) pela compreensão aos momentos de ausência e distância;

Aos meus orientadores Evaristo Biscaia e Argimiro Secchi pela grande oportunidade, paciência e ensinamentos;

A todos os meus amigos e colegas do LADES-COPPE, em especial aos que me ajudaram e incentivaram ao longo desses anos de convívio, aprendizado e amizade: Evandro, Franklin, Orenstein, Eliza, Trica, Bendia, Dorigo, Lilian, Thiago, Michel, Caio, Pedro, Cido, Simone, Danilo, Dyego, Paiva, Thamires, Rafael, Diego Prata, Eduardo Lemos (*in memoriam*), Almeida, Danielle e todos os outros alunos e professores que já participaram das atividades do LADES;

Aos professores Verônica Calado e Rogério Pagano, por serem fundamentais no princípio de minha trajetória acadêmica;

Aos companheiros de trabalho da PETROBRAS, especialmente à célula de otimização, pela oportunidade profissional única, confiança e aprendizado ao longo desses anos (2008-2014);

À comunidade do PEQ: aos professores do PEQ-COPPE pelos ensinamentos, aos colegas de turma e aos funcionários, sempre atenciosos;

À comunidade acadêmica em geral, que sempre se ajuda voluntariamente através de fóruns, aulas, compartilhamento de ideias e materiais e discussões *online*;

A todos que no decorrer da minha vida, direta ou indiretamente me guiaram até aqui.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

OTIMIZAÇÃO DINÂMICA USANDO APROXIMAÇÃO COM BASE WAVELETS ADAPTATIVA

Lizandro de Sousa Santos

Julho/2014

Orientadores: Evaristo Chalbaud Biscaia Jr.

Argimiro Resende Secchi

Programa: Engenharia Química

Nos últimos têm havido esforços consideráveis para o desenvolvimento de estratégias que objetivam aprimorar o desempenho de algoritmos para solução de problemas de otimização dinâmica. Uma das maiores limitações desses problemas é o elevado custo computacional decorrente da estrutura matemática do sistema algébrico diferencial resultante. No algoritmo proposto, baseado em análise wavelets, uma estratégia de compressão dos parâmetros de discretização das variáveis é aplicada no domínio wavelets: coeficientes de baixa magnitude são comprimidos, provocando uma significativa redução no conjunto de variáveis de decisão do problema de programação não linear (NLP) e indicando regiões prospectivas para discretização. A quantidade e a localização dos coeficientes dependem das políticas de compressão adotadas, que é função do *threshold* (limiar). Neste trabalho comparam-se diferentes estratégias para escolha do *threshold* e como podem influenciar o desempenho do algoritmo. Resultados mostram que o desempenho está relacionado ao nível de complexidade do problema, que é influenciado majoritariamente pelo número de descontinuidades e dimensão do sistema. Em um conjunto de exemplos foi possível obter uma redução média de 60% do custo computacional em relação à solução de referência.

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

DYNAMIC OPTIMIZATION USING ADAPTIVE WAVELETS BASES AS
APPROXIMATION

Lizandro de Sousa Santos

July/2014

Advisors: Evaristo Chalbaud Biscaia Jr.

Argimiro Resende Secchi

Department: Chemical Engineering

Recent works have been focusing upon adaptive parameterization algorithms applied to dynamic optimization problems. In particular, shooting methods have been deserved special attention. The performance of these methods is much dependent on the discretization level of the control variables and, therefore, accurate solutions may require fine discretization and, consequently, high computational cost. It is proposed a wavelet thresholding algorithm which provides a sequence of successively refined grids for piecewise constantly discretized control profiles. A new wavelet thresholding, called CVPshrink, tailored to the need of optimal control problems, is developed. The different rules are applied to a set of ten dynamic optimization problems. Obtained results have shown that thresholding rule can be optimal for each case (refinement level, control profiles switching points) and must be carefully chosen in order to guarantee an efficient refinement algorithm.

Sumário

Capítulo 1	Introdução	1
Capítulo 2	Fundamentos da Otimização Dinâmica	7
2.1	Formulação Geral	7
2.2	Condições de Otimalidade	10
2.2.1	Lagrangeano Aumentado do Funcional Objetivo	10
2.2.2	Variação (incremento) de uma Função $x(t)$	11
2.2.3	Definição de Incremento de um Funcional	12
2.2.4	Equação de Euler-Lagrange (condições necessárias de otimalidade para fronteiras fixas)	12
2.2.5	Condições Necessárias de Otimalidade	14
2.2.6	Inserção da Variável de Controle	17
2.2.7	Condição de Weierstrass	18
2.2.8	Condição de Weierstrass: Generalização	21
Capítulo 3	Modelagem e Otimização Dinâmica de Sistemas Algébrico-Diferenciais	24
3.1	Sistemas Algébrico-Diferenciais (DAEs)	24
3.1.1	Aspectos Gerais de Modelagem	24
3.1.2	Métodos Numéricos para DAEs	25
3.2	Métodos de Solução de DAOPs	28
3.2.1	Programação Dinâmica e Formulação de Hamilton-Jacobi	29
3.2.2	Métodos Indiretos	31
3.2.3	Método Sequencial	36
3.2.4	Método Simultâneo	40
3.2.5	Método Multiple-Shooting	44
3.3	Considerações Adicionais sobre DAOPs	46

Capítulo 4	Aplicações, Avanços Recentes e Novas Propostas	52
4.1	Visão Gral sobre Aplicações de DAOPs em Processos Químicos	52
4.2	Estratégias de Adaptação de Malhas em Otimização Dinâmica: Aplicação de Wavelets	57
4.3	<i>Wavelet-Thresholding</i>	62
Capítulo 5	Introdução à Análise Wavelets	66
5.1	Introdução às Wavelets	67
5.1.1	Histórico	67
5.1.2	Princípios Básicos	68
5.2	Principais Propriedades das Wavelets	71
5.2.1	Ortogonalidade	71
5.2.2	Admissibilidade	71
5.2.3	Caracterização Local de Regularidade	71
5.3	Análise Multirresolução (AMR)	72
5.3.1	Interpretação das Wavelets como AMR	72
5.3.2	Banco de Filtros	78
5.4	Família de Wavelets	86
5.5	Introdução ao Procedimento <i>Thresholding</i>	91
5.6	Em busca do <i>Threshold</i> Ideal	93
5.6.1	Estratégia I	93
5.6.2	Estratégia II	94
5.7	Critérios <i>Thresholding</i>	95
5.7.1	<i>Visushrink</i>	95
5.7.2	<i>Sureshrink</i>	98
5.8	Aplicações de Wavelets em Engenharia: da Solução de Sistemas de Equações à Otimização Dinâmica	104
5.9	Resumo do Capítulo	106

Capítulo 6	Algoritmo Wavelet-Thresholding Adaptativo (WtA)	108
6.1	Algoritmo Adaptativo para Otimização Dinâmica	108
6.1.1	Introdução	109
6.1.2	Estrutura do Algoritmo Wavelet-Thresholding Adaptativo	109
6.1.3	Critério de Convergência	112
6.1.4	Rotinas Implementadas	114
6.2	Etapa de <i>Thresholding</i>	116
Capítulo 7	Aplicações	126
7.1	Reator Semi-Batelada Isotérmico (Srinivasan <i>et al.</i> , 2003a)	126
7.1.1	Apresentação do Problema	126
7.1.2	Aplicação do WtA	128
7.2	CSTR Não Linear (Balsa-Canto <i>et al.</i> , 2001)	134
7.3	Avaliação de Desempenho	141
7.3.1	Visão Geral	141
7.3.2	Observações Complementares	151
7.4	Resultados: Modelos e Trajetórias de Controle	141
Capítulo 8	Conclusões	180
	Referências	183

Capítulo 1

Introdução

Define-se como otimização dinâmica o procedimento que determina a trajetória ótima de um processo ou sistema dinâmico o qual conduz à maximização (ou minimização) de um dado funcional dependente de funções variantes no tempo e possivelmente sujeitas a restrições variantes ou não com o tempo (ATHANS e FALB, 1966). Na engenharia, tal sistema dinâmico pode ser caracterizado, por exemplo, por modelos fenomenológicos como escoamento de fluidos, reações químicas, processos físico-químicos de separação de materiais, etc. (BIEGLER, 2010). Quando o problema de otimização dinâmica é representado por um sistema algébrico-diferencial é denominado como DAOP (*Differential-Algebraic Optimization Problem*). A literatura que aborda a Teoria de Otimização Dinâmica, apresenta diferentes estratégias de solução, dividindo os métodos em duas classes: **(i)** métodos diretos (numéricos) e **(ii)** indiretos (analíticos).

Os métodos indiretos propõem a solução analítica do problema, aplicando o Princípio de Máximo de Pontryagin (Pontryagin *et al.*, 1962). Porém, apresentam sérias limitações (como desconhecimento da ordem de grandeza de algumas variáveis auxiliares) para solucionar problemas mais complexos, que envolvam grande quantidade de equações, variáveis e restrições. No entanto, essa formulação pode ser útil para ser agregada a algoritmos numéricos para o cálculo de algumas etapas, como pré-definição de arcos de controle ou gerar estimativas iniciais para os métodos numéricos (BIEGLER, 2010).

Os métodos diretos aplicam diretamente as condições necessárias de otimalidade ao DAOP, gerando um problema de Programação Não Linear (NLP - *Non Linear Programming*). Tais métodos são classificados como **(a)** simultâneo: discretização total das variáveis de estado e controle; **(b)** sequencial (ou *single-shooting*): discretização das variáveis de controle e integração do modelo algébrico-diferencial; e **(c)** híbridos (ou *multi-shooting*): discretização das variáveis de controle e partição do domínio temporal

em subdomínios conectados por condições de continuidade. Essa última classificação é geralmente tratada como uma subdivisão da abordagem sequencial.

Nos método sequencial a discretização da variável de controle resulta em pontos nodais localizados no domínio temporal, que são as variáveis do problema de otimização. Após a interpolação desses pontos por uma estratégia específica, origina-se o perfil de controle utilizado no *solver* de integração.

Uma das principais vantagens do método sequencial é a relativa facilidade de implementação, em se tratando da possibilidade de discretização apenas da variável de controle, e subsequente integração do sistema de equações algébrico-diferenciais (DAE) por rotinas de integração já consolidadas, como a DASSL (BRENAN *et al.*, 1996).

O método simultâneo se caracteriza pela realização da discretização total do sistema de equações, aproximando-se geralmente as variáveis do problema por polinômios. Tal aproximação gera um sistema algébrico que é otimizado por um método específico de otimização. Nesse contexto, as equações são efetivamente satisfeitas somente ao fim da otimização, caracterizando por isso essa estratégia como de caminho inviável. Tais métodos são geralmente menos dispendiosos que os métodos sequenciais, porém tal afirmação não é uma regra geral e não há um consenso estabelecido de qual é a melhor estratégia dentre as duas (BIEGLER, 2010).

Já no método *multiple-shooting* cada subintervalo do perfil de controle é integrado independentemente, impondo-se novas restrições ao algoritmo de otimização. Tais restrições exigem a continuidade das variáveis de estado entre o fim de um intervalo e início do intervalo vizinho. A estratégia *multiple-shooting* em geral é mais estável que a estratégia *single-shooting* e pode se tornar bastante atrativa se utilizada em conjunto com algoritmos paralelos para integração dos períodos em diferentes processadores (DIEHL *et al.*, 2006).

Em se tratando dos métodos numéricos, o nível de discretização das variáveis é crucial para a eficácia dos métodos. Em outras palavras, quanto mais refinada for a discretização do sistema, mais o mesmo se aproxima da solução analítica. No entanto, para obter um perfil de controle com elevada discretização, o custo computacional tende naturalmente a se elevar devido à maior dimensão do problema de otimização. Ademais,

a solução típica desses problemas resulta geralmente em trajetórias de controle distintas, divididas em subintervalos, denominados arcos. Geralmente esses arcos são gerados por ativações e desativações das restrições do DAOP. Na mudança de um arco para outro, pode haver descontinuidades no perfil de controle, necessitando-se de uma discretização refinada em torno dessas regiões. Além disso, a discretização também interfere no tratamento das restrições de trajetória (restrições ao longo do tempo) as quais são computadas em pontos específicos do domínio temporal. Logo, em geral, quanto maior o nível de discretização, mais correta será a avaliação dessas restrições, porém, mais custosa computacionalmente (CHACHUAT, 2006).

Constata-se que há uma crescente preocupação em sofisticar os algoritmos para solução de tais problemas, tornando cada vez mais viável a aplicação industrial. Nota-se que os aspectos mais valorizados dizem respeito a: **(i)** tratamento das restrições de trajetória, **(ii)** cômputo da matriz Jacobiana, **(iii)** aprimoramento de algoritmos de otimização **(iv)** robustez na solução dos sistemas de DAEs e **(v)** desenvolvimento de novas estratégias de discretização.

O objetivo deste estudo é propor avanços numéricos para solucionar DAOPs. Para isso, propõe-se um procedimento de discretização adaptativa baseado em bases wavelets, denominado aqui Wavelet-thresholding Adaptativo (WtA). Wavelets são funções com certas propriedades matemáticas que permitem a representação de um conjunto de dados em diferentes níveis de resolução (ou aproximação). Por consequência, permite-se analisar as principais características desse conjunto, havendo a possibilidade de manipular tais dados a fim de compactá-los ou remover possíveis ruídos ou dados irrelevantes. As wavelets são compostas por funções, em geral, bastante irregulares, dadas as propriedades que lhes são impostas. Uma das propriedades que se procura garantir é a da ortogonalidade. A tentativa de obter um sistema de funções ortogonais definidas dentro de um intervalo finito origina funções bastante irregulares e assimétricas, como é o caso das wavelets Daubechies. Não existe, em geral, uma expressão analítica para as funções de escala e wavelets, sendo que tais funções são definidas através de fórmulas recursivas (BURRUS *et al.*, 2002).

A inovação consiste em importar mecanismos utilizados para compressão de dados, já difundidos na área de processamento digital e tratamento de imagens, para

seleção ótima das bases wavelets utilizadas para representação das variáveis do DAOP, no domínio wavelets. Tal seleção se baseia na escolha ótima do limiar (*threshold*). Uma nova estratégia para o cálculo do *threshold*, CVPS (*Control Vector Parameterization Shrink*) foi desenvolvida com o objetivo de aperfeiçoar o desempenho numérico em problemas com múltiplas variáveis de controle. Tal estratégia foi comparada a outras estratégias utilizadas na área de compressão de imagens, como o *Visushrink* e *Sureshrink*. Diferentemente do procedimento realizado para compressão de imagens, onde tais estratégias são utilizadas para remoção de ruídos, as wavelets são utilizadas somente para indicar regiões prospectivas para discretização.

A estratégia sugerida neste trabalho consiste em acoplar o WtA ao método *single-shooting*, no qual o nível de discretização das variáveis de controle (malha) evolui em elementos (estágios de discretização) com dimensões variadas. Tal discretização se modifica a cada iteração do algoritmo para melhorar a acurácia da aproximação em determinados pontos específicos no domínio de controle, mais especificamente em regiões de descontinuidade ou transitoriedades. Desse modo, o algoritmo proposto utiliza um procedimento automático para refinar a malha, a fim de atingir a acurácia requerida com o objetivo de reduzir o esforço computacional do algoritmo de otimização e facilitar a etapa de integração do sistema DAE. Dessa maneira, a cada iteração caminha-se para uma melhor estimativa inicial do perfil de controle e conseqüentemente facilita-se a convergência para o perfil ótimo. A necessidade de refinar a malha depende da dinâmica do processo, a qual pode afetar a qualidade da aproximação da variável de controle e, conseqüentemente, a estratégia de solução. Por conseqüência, o número e a posição dos pontos da discretização dependem da política adotada de compressão do vetor de pontos resultantes.

O algoritmo desenvolvido gerou uma *toolbox* em ambiente MATLAB[®] a qual utiliza diferentes estratégias de *wavelet-threshoding* para comprimir a malha e indicar regiões prospectivas para discretização ao longo do cálculo de otimização. Além disso, disponibiliza um ambiente intuitivo para inserção do DAOP (modelos algébrico-diferenciais, restrições e função objetivo) possibilitando também o cômputo automático das equações de sensibilidade, paralelização do código e integração com simuladores externos como o SIMULINK, EMSO, ASPEN e HYSYS.

Ao longo do texto, vários exemplos são abordados para elucidação das discussões. A metodologia também é consolidada em um capítulo de estudo de casos no qual são abordados vários *benchmarks* de DAOPs, majoritariamente de engenharia química.

A estrutura do texto é sumarizada abaixo:

Capítulo 2: Fundamentos da Otimização Dinâmica: O foco deste capítulo é introduzir os conceitos principais da teoria de otimização dinâmica, que são vitais para discussão e revisão da aplicação de DAOPs, no Capítulo 3. Procura-se deduzir de maneira sucinta as condições de otimalidade sobre problemas variacionais.

Capítulo 3: Modelagem e Otimização Dinâmica de Sistemas Algébrico Diferenciais: O objetivo deste capítulo é discutir aspectos de otimização dinâmica de sistemas algébrico-diferenciais, resultantes da modelagem de processos químicos. Os métodos mais usuais para solução de DAOPs são introduzidos assim como as principais contribuições e ferramentas disponíveis.

Capítulo 4: Aplicações, Avanços Recentes e Novas Propostas: O objetivo deste capítulo é resumir as principais pesquisas recentes sobre projetos de DAOPs. Procura-se retratar as principais limitações que desencorajam a aplicabilidade industrial. Sobre esse prisma, realiza-se uma revisão sobre pesquisas envolvendo adaptação de malhas aplicada a sistemas algébrico-diferenciais.

Capítulo 5: Introdução à Análise Wavelets: O propósito deste capítulo é apresentar as propriedades fundamentais das wavelets e levantar as suas principais características já com objetivo de destacar as ferramentas provenientes da análise multirresolução de dados que são utilizadas para intervir nas etapas de discretização nos DAOPs. Além disso, introduzem-se as principais características das wavelets para remoção de ruídos e compressão de dados.

Capítulo 6: Algoritmo Wavelet-Thresholding Adaptativo (WtA): Baseado na teoria desenvolvida nos capítulos anteriores, o escopo deste capítulo é apresentar o algoritmo *Wavelet-thresholding* adaptativo WtA proposto.

Capítulo 7: Aplicações e Discussão dos Resultados: Neste capítulo aplica-se o algoritmo WtA em mais de 20 problemas de controle ótimo. Os problemas foram agrupados de maneira a realizar um estudo estatístico de análise de desempenho das estratégias *thresholding*.

Capítulo 8: Conclusões

Capítulo 2

Fundamentos da Otimização Dinâmica

Neste Capítulo
Tópico 2.1. <u>Formulação Geral;</u>
Tópico 2.2. <u>Condições de otimalidade;</u>

O foco deste capítulo é introduzir os conceitos principais da teoria de otimização dinâmica, que são vitais para discussão e revisão da aplicação de DAOPs, no Capítulo 3. Procura-se deduzir de maneira sucinta as condições de otimalidade sobre problemas variacionais, resultando na condição de Weierstrass, que é um prelúdio ao princípio de Máximo de Pontryagin. No **Tópico 2.1** introduz-se a formulação matemática de um problema geral de otimização dinâmica. No **Tópico 2.2** as condições de otimalidade são deduzidas.

2.1 Formulação Geral

O Cálculo das Variações constitui uma excelente introdução à Teoria de Otimização Dinâmica, ou Teoria do Controle Ótimo, que é mais ampla e endereçada para tratar problemas mais realistas, ou seja, que apresentam restrições ao longo do tempo (CINTRON, 1969). No Cálculo das Variações o objetivo é encontrar o mínimo (ou máximo) de um funcional $\mathfrak{J} : \mathcal{D} \rightarrow \mathbb{R}$, em que \mathcal{D} é um subconjunto de um espaço linear de funções contínuas reais $\mathbf{x}(t) \in \mathbb{R}^{n_x}$:

$$\mathfrak{J}(\mathbf{x}) := \int_{t_0}^{t_f} L(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) dt \quad (2.1)$$

em que o funcional \mathfrak{J} é denominado função custo ou função objetivo, $t \in \mathbb{R}$, $t_0 \leq t \leq t_f$, é a variável tempo; $\mathbf{x}(t) \in \mathbb{R}^{n_x}$, com $n_x \geq 1$, é um vetor de funções reais; as funções $\mathbf{x}(t) = (x_1(t), \dots, x_{n_x}(t))$, são geralmente conhecidas como trajetórias ou curvas; $\dot{\mathbf{x}}(t) \in \mathbb{R}^{n_x}$ é a derivada temporal de $\mathbf{x}(t)$; e $L : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$ é uma função real conhecida como Lagrangeana. Nesses problemas, a função objetivo depende da variável

de decisão $\mathbf{x}(t)$, da sua taxa de variação $\dot{\mathbf{x}}(t)$, além da variável t . Logo, o problema de otimização pode ser escrito como:

$$\min_{\mathbf{x}(t)} \mathfrak{J} = \min_{\mathbf{x}(t)} \int_{t_0}^{t_f} L(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) dt \quad (2.2a)$$

$$\begin{aligned} \text{sujeito a } \mathbf{x}(t) &\in X(t) \subset \mathbb{R}^{n_x}, \quad t_0 \leq t \leq t_f \\ \mathbf{x}_0 &= \mathbf{x}(t_0) \end{aligned} \quad (2.2b)$$

A solução desse problema é uma função $\mathbf{x}^*(t)$ definida como curva ótima, ou trajetória ótima, a qual resulta no valor extremo de \mathfrak{J} .

As equações de Euler (ATHANS e FALB, 1966) são o resultado das condições necessárias de otimalidade aplicadas a esse tipo de problema. Para problemas com condições de fronteira (t_0, x_0, t_f, x_f) conhecidas, as equações de Euler são escritas como:

$$\left[\begin{array}{c} \frac{d}{dt} \frac{\partial L}{\partial \dot{x}_1} - \frac{\partial L}{\partial x_1} = 0 \\ \vdots \\ \frac{d}{dt} \frac{\partial L}{\partial \dot{x}_{n_x}} - \frac{\partial L}{\partial x_{n_x}} = 0 \end{array} \right] \quad (2.3)$$

Uma generalização desses problemas é conhecida como Controle Ótimo, em que as variáveis são separadas em duas classes: estado (dependentes) $\mathbf{x}(t)$ e controle (independentes) $\mathbf{u}(t) \in \mathbb{R}^{n_u}$, $\mathbf{u}(t) = (u_1(t), \dots, u_{n_u}(t))$. A evolução temporal das variáveis de estado depende dos valores das variáveis de controle, ou seja:

$$\begin{aligned} \dot{x}_1(t) &= f_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots \\ \dot{x}_{n_x}(t) &= f_{n_x}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \end{aligned} \quad (2.4)$$

em que $\mathbf{f} : \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}$.

Logo o problema de otimização dinâmica pode ser escrito como:

$$\min_{\mathbf{u}(t)} \mathfrak{J} = \min_{\mathbf{u}(t)} \left[\underbrace{\varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f))}_{\text{Termo de Meyer}} + \underbrace{\int_{t_0}^{t_f} L(t, \mathbf{x}(t), \mathbf{u}(t)) dt}_{\text{Termo de Lagrange}} \right] \quad (2.5)$$

$$\text{sujeito a } \dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}_0(t) = \mathbf{x}_0(t_0) \quad (2.5a)$$

em que $\mathbf{f} = (f_1, \dots, f_{n_x})$.

Na Equação (2.5) o termo de Meyer $\varphi: \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}$, representa as condições de fronteira entre as variáveis, enquanto o termo de Lagrange representa a variação das variáveis ao longo do intervalo de tempo $[t_0 \ t_f]$. Tal formulação é definida como Formulação de Bolza (CHACHUAT, 2006). Adicionalmente, na Equação (2.5) o termo de Lagrange pode ser incorporado ao sistema de equações, gerando mais uma variável de estado:

$$\dot{x}_{n_{x+1}}(t) = \frac{d}{dt} \left[\int_{t_0}^{t_f} L(t, \mathbf{x}(t), \mathbf{u}(t)) dt \right] = L(t, \mathbf{x}(t), \mathbf{u}(t)) \quad (2.6)$$

Dessa maneira, a Equação 2.5 pode ser reformulada como:

$$\min_{\mathbf{u}(t)} \mathfrak{J} = \min_{\mathbf{u}(t)} \left[\underbrace{\varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f))}_{\text{Termo de Meyer}} + x_{n_{x+1}}(t_f) \right] \quad (2.7)$$

$$\begin{aligned} \text{sujeito a } \dot{\mathbf{x}}(t) &= \mathbf{f}(t, \mathbf{x}(t), \mathbf{u}(t)), \mathbf{x}_0(t) = \mathbf{x}_0(t_0), t_0 < t < t_f \\ \dot{x}_{n_{x+1}}(t) &= l(t, \mathbf{x}(t), \mathbf{u}(t)), x_{n_{x+1},0}(t) = x_{n_{x+1},0}(t_0) \end{aligned} \quad (2.7a)$$

Além da Equação (2.7a), o problema pode apresentar restrições de igualdade definidas por equações algébricas e/ou integrais. Logo, é conveniente generalizar o sistema dinâmico, de maneira implícita já incorporando tais condições, ou seja:

$$\mathbf{h}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t)) = 0 \quad (2.8)$$

sendo $\mathbf{h}: \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_h}$.

Restrições de desigualdade também podem surgir, sendo representadas na equação seguinte:

$$\mathbf{g}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t)) \leq 0 \quad (2.9)$$

sendo $\mathbf{g}: \mathbb{R} \times \mathbb{R}^{n_x} \times \mathbb{R}^{n_u} \rightarrow \mathbb{R}^{n_g}$.

Assim como restrições sobre as variáveis de controle:

$$\mathbf{u}^{\min} \leq \mathbf{u}(t) \leq \mathbf{u}^{\max} \quad (2.10)$$

Em resumo, a formulação geral do problema do problema de otimização dinâmica é definida no quadro seguinte:

$$\min_{\mathbf{u}(t), t_f, \mathbf{p}} \mathfrak{J} = \min_{\mathbf{u}(t), t_f, \mathbf{p}} \left[\varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f), \mathbf{p}) \right] \quad (2.11a)$$

$$\text{sujeito a } \mathbf{h}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}) = 0 \quad (2.11b)$$

$$\mathbf{g}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \mathbf{u}(t), \mathbf{p}) \leq 0$$

$$\mathbf{e}(t_f, \mathbf{x}(t_f)) \leq 0$$

$$\mathbf{u}^{\min} \leq \mathbf{u}(t) \leq \mathbf{u}^{\max}$$

$$\mathbf{x}_0 = \mathbf{x}_0(t_0), t_0 < t < t_f$$

Na Equação (2.11), parâmetros independentes do tempo, $\mathbf{p} \in \mathbb{R}^{n_p}$, também são incorporados ao problema. Incluem-se também as condições terminais definidas implicitamente por $\mathbf{e}(t_f, \mathbf{x}(t_f), \dot{\mathbf{x}}(t_f)) \leq 0$, com $\mathbf{e}: \mathbb{R} \times \mathbb{R}^{n_x} \rightarrow \mathbb{R}^{n_e}$. Note que na Equação (2.11) assume-se que as condições iniciais são fixas e conhecidas.

Nos próximos tópicos deduzem-se as condições de otimalidade do sistema acima. Para facilitar a compreensão, assume-se neste capítulo que o sistema dinâmico é puramente diferencial, ou seja, constituído somente por equações diferenciais ordinárias (ODEs). Essa introdução ajudará a discussão sobre otimização dinâmica de sistemas algébrico-diferenciais, realizada no próximo capítulo.

2.2 Condições de Otimalidade

2.2.1 Lagrangeano Aumentado do Funcional Objetivo

Considere o sistema¹:

$$\mathfrak{J} = \varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} L(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) dt \quad (2.12)$$

$$\text{sujeito a } \mathbf{h}(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) = 0$$

Para aplicar as condições necessárias de otimalidade sobre o sistema representado pela Equação (2.12), define-se a função:

$$H(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \boldsymbol{\lambda}) = L(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) + \sum_{i=1}^{n_h} \lambda_i(t) h_i(t, \mathbf{x}(t), \dot{\mathbf{x}}(t)) \quad (2.13)$$

conhecido como Hamiltoniano, sendo $\boldsymbol{\lambda}(t) \in \mathbb{R}^{n_h}$, $\boldsymbol{\lambda}(t) = (\lambda_1, \dots, \lambda_{n_h})$ a função multiplicador de Lagrange. Dessa maneira, um novo funcional é definido como:

¹ Em princípio, para simplificação não se incorpora variáveis de controle.

$$\mathfrak{J} = \varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} H(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \boldsymbol{\lambda}(t)) dt \quad (2.14)$$

2.2.2 Variação (incremento) de uma Função $x(t)$

A função $x(t, \varepsilon)$ pode ser representada como expansão em série de Taylor em torno do ponto $\varepsilon = 0$, onde $\varepsilon \in \mathbb{R}$ é um parâmetro escalar que indica que $x(t, \varepsilon)$ se localiza em uma vizinhança próxima de $x(t)$:

$$x(t, \varepsilon) = x(t) + \left. \frac{\partial x}{\partial \varepsilon} \right|_{\varepsilon=0} \varepsilon + O(\varepsilon^2) \quad (2.15)$$

O termo de primeira ordem predominante na Equação (2.15) é definido como:

$$\delta x(t) \cong \left. \frac{\partial x}{\partial \varepsilon} \right|_{\varepsilon=0} \varepsilon \quad (2.16)$$

Logo, $\delta x(t)$ representa a variação no instante t entre as funções $x(t, \varepsilon)$ e $x(t)$, ou seja:

$$\delta x(t) = x(t, \varepsilon) - x(t) \quad (2.17)$$

Por consequência, define-se a quantidade $\delta \Omega$, $\Omega \in \mathbb{R}$, como:

$$\delta \Omega \cong \left. \frac{d\Omega}{d\varepsilon} \right|_{\varepsilon=0} \varepsilon \quad (2.18)$$

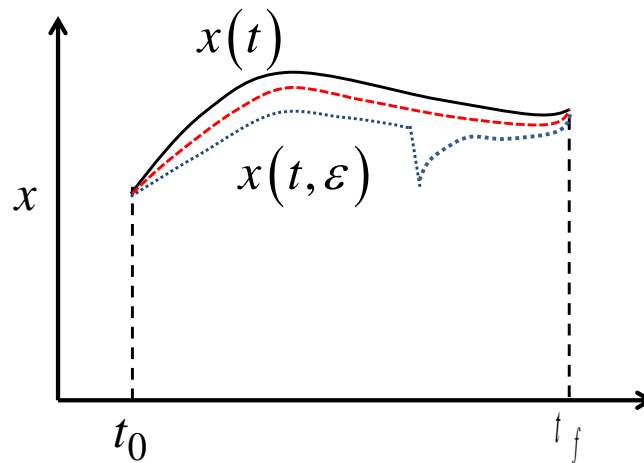


Figura 2.1. Ilustração de variação fraca e forte, considerando as condições iniciais e finais fixas. A curva tracejada indica variação fraca, enquanto a curva pontilhada indica variação forte (baseado em CINTRON, 1969).

A variação $\delta x(t)$ é considerada fraca, de acordo com as restrições: $\|x(t, \varepsilon) - x(t)\| \leq \varepsilon$ e $\|\dot{x}(t, \varepsilon) - \dot{x}(t)\| \leq \varepsilon$. O caso de variação forte implica em somente a primeira restrição ser válida. A Figura 2.1 ilustra os dois casos.

2.2.3 Definição de Incremento de um Funcional

Considere agora o funcional definido pela Equação (2.14):

$$\mathfrak{J} = \varphi(t_0, \mathbf{x}(t_0), t_f, \mathbf{x}(t_f)) + \int_{t_0}^{t_f} H(t, \mathbf{x}(t), \dot{\mathbf{x}}(t), \lambda(t)) dt \quad (2.14)$$

Substituindo-se a aproximação de cada variável em torno de uma vizinhança próxima:

$$\begin{aligned} \mathfrak{J}(\varepsilon) = & \varphi(t_0(\varepsilon), \mathbf{x}(t_0(\varepsilon)), t_f(\varepsilon), \mathbf{x}(t_f(\varepsilon), \varepsilon)) + \\ & \int_{t_0(\varepsilon)}^{t_f(\varepsilon)} H(t, \mathbf{x}(t, \varepsilon), \dot{\mathbf{x}}(t, \varepsilon), \lambda(t, \varepsilon)) dt \end{aligned} \quad (2.19)$$

Sabendo-se que:

$$\delta x \cong \left. \frac{\partial x}{\partial \varepsilon} \right|_{\varepsilon=0} \varepsilon, \quad \delta \dot{x} \cong \left. \frac{\partial \dot{x}}{\partial \varepsilon} \right|_{\varepsilon=0} \varepsilon \quad \text{e} \quad \delta \lambda \cong \left. \frac{\partial \lambda}{\partial \varepsilon} \right|_{\varepsilon=0} \varepsilon \quad (2.20)$$

Ao expandir a Equação (2.14) em torno de $\varepsilon = 0$ resulta em:

$$\mathfrak{J}(\varepsilon) = \mathfrak{J}(0) + \left. \frac{d\mathfrak{J}}{d\varepsilon} \right|_{\varepsilon=0} \varepsilon + \left. \frac{d^2\mathfrak{J}}{d\varepsilon^2} \right|_{\varepsilon=0} \frac{\varepsilon^2}{2} + O(\varepsilon^3) \quad (2.21)$$

A condição necessária para $\mathfrak{J}(0)$ ser um extremo é:

$$\delta \mathfrak{J} = \left. \frac{d\mathfrak{J}}{d\varepsilon} \right|_{\varepsilon=0} \varepsilon = 0 \quad (2.22)$$

que é a primeira variação do funcional. Logo, a definição do incremento do funcional \mathfrak{J} pode ser escrita como:

$$\delta \mathfrak{J} = \mathfrak{J}(\varepsilon) - \mathfrak{J}(0) \quad (2.23)$$

2.2.4 Equação de Euler-Lagrange (condições necessárias de otimalidade para fronteiras fixas)

Considere primeiramente o sistema:

$$\mathfrak{J}(\varepsilon) = \varphi(t_0, x(t_0), t_f, x(t_f)) + \int_{t_0}^{t_f} H(t, x(t, \varepsilon), \dot{x}(t, \varepsilon), \lambda) dt \quad (2.24)$$

em que:

$$H(t, x(t), \dot{x}(t), \lambda) = L(t, x(t)) + \lambda(t)h(t, x(t), \dot{x}(t)).$$

Supondo-se que as condições iniciais $(t_0, x(t_0))$ e finais $(t_f, x(t_f))$ são fixas, a primeira variação de \mathfrak{J} é escrita como:

$$\delta\mathfrak{J}(\varepsilon) = \int_{t_0}^{t_f} \delta H(t, x(t, \varepsilon), \dot{x}(t, \varepsilon), \lambda(t, \varepsilon)) dt \quad (2.25)$$

Aplicando o operador δ , definido pela Equação (2.23), e usando a regra da cadeia, resulta em:

$$\delta\mathfrak{J}(\varepsilon) = \frac{d\mathfrak{J}(\varepsilon)}{d\varepsilon} \Big|_{\varepsilon=0} = \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial x} \left(\underbrace{\frac{\partial x}{\partial \varepsilon} \Big|_{\varepsilon=0}}_{\delta x} \right) + \frac{\partial H}{\partial \dot{x}} \left(\underbrace{\frac{\partial \dot{x}}{\partial \varepsilon} \Big|_{\varepsilon=0}}_{\delta \dot{x}} \right) \right] dt \quad (2.26)$$

ou

$$\delta\mathfrak{J} = \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial x} \delta x + \frac{\partial H}{\partial \dot{x}} \delta \dot{x} \right] dt \quad (2.27)$$

Sabendo-se que:

$$\delta \dot{x} = \frac{d}{dt} \delta x \quad (2.28)$$

a Equação (2.27) é escrita como:

$$\delta\mathfrak{J} = \int_{t_0}^{t_f} \left[\frac{\partial H}{\partial x} \delta x + \frac{\partial H}{\partial \dot{x}} \left(\frac{d}{dt} \delta x \right) \right] dt \quad (2.29)$$

Como o segundo termo da Equação (2.29) pode ser integrado por partes:

$$\begin{aligned} \int_{t_0}^{t_f} \frac{\partial H}{\partial \dot{x}} \left(\frac{d}{dt} \delta x \right) dt &= \frac{\partial H}{\partial \dot{x}} \delta x \Big|_{t_0}^{t_f} - \int_{t_0}^{t_f} \delta x \left(\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} \right) dt \\ &\quad \downarrow \qquad \qquad \downarrow \qquad \qquad \downarrow \\ &\int A \cdot \dot{B} \qquad A \cdot B \Big|_{t_0}^{t_f} \qquad \int B \cdot \dot{A} \end{aligned} \quad (2.30)$$

reescreve-se a Equação (2.29):

$$\delta\mathfrak{J} = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial x} - \frac{d}{dt} \frac{\partial H}{\partial \dot{x}} \right) \delta x dt - \left(\frac{\partial H}{\partial \dot{x}} \delta x \right) \Big|_{t_0}^{t_f} \quad (2.31)$$

Como para o caso de condições inicial e final fixas $\delta x(t_0) = \delta x(t_f) = 0$, a condição necessária para atingir o mínimo (ou máximo) é:

$$\delta \mathfrak{J} = \int_{t_0}^{t_f} \left(\frac{\partial H}{\partial x} - \frac{d}{dt} \frac{\partial H}{\partial \dot{x}} \right) \delta x \, dt = 0 \quad (2.32)$$

que resulta em:

$$\frac{\partial H}{\partial x} - \frac{d}{dt} \frac{\partial H}{\partial \dot{x}} = 0 \quad (2.33)$$

Essa equação é conhecida como equação de Euler-Lagrange ou equação característica. A equação de Euler-Lagrange é proveniente da aplicação da condição necessária de primeira ordem: $\delta \mathfrak{J} = 0$ (ATHANS e FALB, 1966).

2.2.5 Condições Necessárias de Otimalidade para Fronteiras Livres

Considere agora que $t_0 = t_0(\varepsilon)$, $x(t_0) = x(t_0(\varepsilon), \varepsilon)$, $t_f = t_f(\varepsilon)$ e $x(t_f) = x(t_f(\varepsilon), \varepsilon)$. Ou seja, as condições de contorno não são fixas² (Figura 2.2). Aplicando o operador δ sobre esses termos temos:

$$\delta t_f = \left. \frac{dt_f}{d\varepsilon} \right|_{\varepsilon=0} \varepsilon \quad (2.34)$$

$$\delta t_0 = \left. \frac{dt_0}{d\varepsilon} \right|_{\varepsilon=0} \varepsilon \quad (2.34a)$$

$$\delta x(t_0, \varepsilon) = \left. \frac{dx(t_0)}{dt_0} \frac{dt_0}{d\varepsilon} \varepsilon + \frac{\partial x(t_0)}{\partial \varepsilon} \varepsilon \right|_{\varepsilon=0} = \dot{x}(t_0) \delta t_0 + \delta x(t_0) \quad (2.34b)$$

$$\delta x(t_f, \varepsilon) = \left. \frac{dx(t_f)}{dt_f} \frac{dt_f}{d\varepsilon} \varepsilon + \frac{\partial x(t_f)}{\partial \varepsilon} \varepsilon \right|_{\varepsilon=0} = \dot{x}(t_f) \delta t_f + \delta x(t_f) \quad (2.34c)$$

Substituindo as aproximações na Equação (2.25), resulta em:

$$\mathfrak{J}(\varepsilon) = \varphi(t_0, x(t_0, \varepsilon), t_f, x(t_f, \varepsilon)) + \int_{t_0}^{t_f} H(t, x(t, \varepsilon), \dot{x}(t, \varepsilon), \lambda) \, dt \quad (2.35)$$

Em seguida, ao aplicar a condição necessária de otimalidade sobre $\mathfrak{J}(\varepsilon)$, resulta na Equação (2.36):

² Esse caso é uma generalização do anterior.

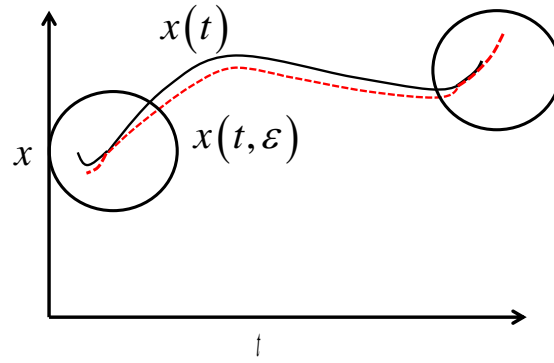


Figura 2.2. Ilustração de trajetórias com fronteiras móveis (ver FLORES, 2011).

$$\delta\mathfrak{S} = \delta\varphi + \int_{t_0}^{t_f} \delta H dt + H|_{t_f} \delta t_f - H|_{t_0} \delta t_0 = 0 \quad (2.36)$$

em que:

$$\delta\varphi = \frac{\partial\varphi}{\partial t_0} \delta t_0 + \frac{\partial\varphi}{\partial x(t_0)} \delta x(t_0) + \frac{\partial\varphi}{\partial x(t_f)} \delta x(t_f) + \frac{\partial\varphi}{\partial t_f} \delta t_f \quad (2.37)$$

A Equação (2.36) é resultante da regra de Leibniz (CINTRON, 1969):

$$I(\alpha) = \int_{a(\alpha)}^{b(\alpha)} f(y, \alpha) dy$$

$$\frac{dI(\alpha)}{d\alpha} = \int_{a(\alpha)}^{b(\alpha)} \frac{\partial f}{\partial \alpha}(y, \alpha) dy + f(b(\alpha), \alpha) \frac{db(\alpha)}{d\alpha} - f(a(\alpha), \alpha) \frac{da(\alpha)}{d\alpha}$$

Por consequência, reescreve-se a Equação (2.36) como:

$$\delta\mathfrak{S} = \delta\varphi + (H \delta t)|_{t_0}^{t_f} - \frac{\partial H}{\partial \dot{x}} \delta x(t, \varepsilon)|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left(\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} - \frac{\partial H}{\partial x} \right) \delta x dt = 0 \quad (2.38)$$

Sabendo-se que:

$$\delta x(t, \varepsilon) = \frac{dx}{dt} \frac{dt}{d\varepsilon} \varepsilon + \frac{\partial x}{\partial \varepsilon} \varepsilon = \dot{x} \delta t + \delta x \quad (2.39)$$

a Equação (2.36) pode ser reformulada como:

$$\delta\mathfrak{S} = \underbrace{\delta\varphi - \left(\frac{\partial H}{\partial \dot{x}} \dot{x} - H \right) \delta t|_{t_0}^{t_f}}_{\text{condições de transversalidade}} - \underbrace{\frac{\partial H}{\partial \dot{x}} \delta x|_{t_0}^{t_f} - \int_{t_0}^{t_f} \left(\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} - \frac{\partial H}{\partial x} \right) \delta x dt}_{\text{equação de Euler-Lagrange}} = 0 \quad (2.40)$$

Os três primeiros termos da Equação (2.40) representam a variação de $\delta\mathfrak{S}$ causada por perturbações nas condições de fronteira. Já o último termo representa a variação de $\delta\mathfrak{S}$ causada por perturbações em x entre os pontos iniciais e finais (entre as fronteiras). Note que a Equação (2.40) é uma generalização da Equação (2.32), pois as condições de transversalidade são adicionadas à equação de Euler-Lagrange.

As condições de transversalidade devem ser satisfeitas para todos os valores possíveis das condições de fronteira $t_0, x(t_0), t_f, x(t_f)$. No entanto, tais condições podem ser restringidas por condições (restrições) extras definidas implicitamente por:

$$\chi_j(t_0, x(t_0), t_f, x(t_f)) = 0, \quad j = 1, 2, \dots, n_e \quad (2.41)$$

sendo que $n_e \leq 2n_x + 2$. Note que χ pode, por exemplo, definir condições iniciais ou finais das variáveis.

Em sequência, a Equação (2.41) pode então ser adicionada ao termo de Meyer do funcional objetivo, utilizando novos multiplicadores de Lagrange σ :

$$G(t_0, x(t_0), t_f, x(t_f)) = \varphi(t_0, x(t_0), t_f, x(t_f)) + \sum_{j=1}^{n_e} \sigma_j \chi_j(t_0, x(t_0), t_f, x(t_f)) \quad (2.42)$$

Logo, a Equação (2.40) é reescrita como:

$$\delta\mathfrak{S} = \underbrace{\delta G - \left(\frac{\partial H}{\partial \dot{x}} \dot{x} - H \right) \delta t \Big|_{t_0}^{t_f} - \frac{\partial H}{\partial \dot{x}} \delta x \Big|_{t_0}^{t_f}}_{\text{condições de transversalidade}} - \underbrace{\int_{t_0}^{t_f} \left(\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} - \frac{\partial H}{\partial x} \right) \delta x dt}_{\text{equação de Euler-Lagrange}} = 0 \quad (2.43)$$

Finalmente, para um sistema com n_x variáveis as equações de Euler-Lagrange são definidas como:

$$\frac{d}{dt} \frac{\partial H}{\partial \dot{x}_1} - \frac{\partial H}{\partial x_1} = 0$$

$$\vdots \quad (2.44)$$

$$\frac{d}{dt} \frac{\partial H}{\partial \dot{x}_{n_x}} - \frac{\partial H}{\partial x_{n_x}} = 0$$

e as condições de transversalidade:

$$\delta\mathfrak{S} = \underbrace{\delta G - \left(\sum_{i=1}^{n_x} \frac{\partial H}{\partial \dot{x}_i} \dot{x}_i - H \right) \delta t \Big|_{t_0}^{t_f} - \sum_{i=1}^{n_x} \frac{\partial H}{\partial \dot{x}_i} \delta x_i \Big|_{t_0}^{t_f}}_{\text{condições de transversalidade}} = 0 \quad (2.45)$$

2.2.6 Inserção da Variável de Controle

Considerando agora que o modelo dinâmico é representado, explicitamente, por $n_x + n_u$ variáveis, governadas pelo seguinte sistema de equações:

$$\begin{aligned} \dot{x}_1 &= f_1(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \\ &\vdots \\ \dot{x}_{n_x} &= f_{n_x}(t, x_1, \dots, x_{n_x}, u_1, \dots, u_{n_u}) \end{aligned} \quad (2.46)$$

Com as variáveis separadas agora em estado \mathbf{x} e controle \mathbf{u} . A função Lagrangeano aumentado³ é definida como:

$$M = \sum_{i=1}^{n_x} \lambda_i(t) [\dot{x}_i - f_i(t, \mathbf{x}, \mathbf{u})] \quad (2.47)$$

Ao aplicar a equação de Euler-Lagrange temos:

$$\begin{aligned} \dot{\lambda}_1 &= - \sum_{i=1}^{n_x} \lambda_i \frac{\partial f_i}{\partial x_1}(t, \mathbf{x}, \mathbf{u}) \\ &\vdots \\ \dot{\lambda}_{n_x} &= - \sum_{i=1}^{n_x} \lambda_i \frac{\partial f_i}{\partial x_{n_x}}(t, \mathbf{x}, \mathbf{u}) \end{aligned} \quad (2.48)$$

e

$$\begin{aligned} \sum_{i=1}^{n_x} \lambda_i \frac{\partial f_i}{\partial u_1}(t, \mathbf{x}, \mathbf{u}) &= 0 \\ &\vdots \\ \sum_{i=1}^{n_x} \lambda_i \frac{\partial f_i}{\partial u_{n_u}}(t, \mathbf{x}, \mathbf{u}) &= 0 \end{aligned} \quad (2.49)$$

As equações (2.48) são definidas como equações adjuntas enquanto as equações (2.49) são definidas como equações de controle.

Já as condições de transversalidade são definidas como:

$$\sum_{i=1}^{n_x} \lambda_i f_i \Big|_{t_0} + \frac{\partial G}{\partial t_i} = 0 \quad (2.50)$$

$$\sum_{i=1}^{n_x} \lambda_i f_i \Big|_{t_F} - \frac{\partial G}{\partial t_F} = 0 \quad (2.51)$$

e

³ Note agora que o termo de Lagrange, que está incorporado ao modelo dinâmico.

$$\lambda_j(t_0) = \frac{\partial G}{\partial x_j(t_0)} \quad j = 1, \dots, n_x \quad (2.52)$$

$$\lambda_j(t_F) = \frac{\partial G}{\partial x_j(t_F)} \quad j = 1, \dots, n_x \quad (2.53)$$

Se houver descontinuidade na variável de controle em algum instante específico t_i as condições de junção de Weierstrass-Erdmann (CINTRON, 1969) devem ser satisfeitas. Tais condições tratam problemas que apresentam descontinuidades, ou seja:

$$\begin{aligned} \lambda_i|_{t_i(+)} &= \lambda_i|_{t_i(-)} \\ &\vdots \\ \lambda_{n_x}|_{t_i(+)} &= \lambda_{n_x}|_{t_i(-)} \end{aligned} \quad (2.54)$$

e

$$\sum_{i=1}^{n_x} \lambda_i f_i|_{t_i(+)} = \sum_{i=1}^{n_x} \lambda_i f_i|_{t_i(-)} \quad (2.55)$$

em que $t_i^{(+)}$ corresponde ao instante de tempo antes do ponto de junção e $t_i^{(-)}$ corresponde ao instante de tempo após o ponto de junção.

2.2.7 Condição de Weierstrass

As condições necessárias definidas nos tópicos anteriores determinam as trajetórias ótimas, restringidas por variações fracas, conforme introduzido no Tópico 2.2.1. Quando essas restrições são relaxadas surgem as condições de Weierstrass. Tais condições são de extrema importância para projetos de controle ótimo, pois prenunciam o Princípio de Máximo de Pontryagin (Pontryagin *et al.*, 1962) que é uma generalização das condições de otimalidade.

Considere as quatro regiões ilustradas na Figura 2.3, definidas como:

$$x(t): t_0 \leq t \leq t_1 \quad (2.56a)$$

$$X(t): t_1 \leq t \leq t_1 + \varsigma \quad (2.56b)$$

$$x(t, \varsigma): t_1 + \varsigma \leq t \leq t_2 \quad (2.56c)$$

$$x(t): t_2 \leq t \leq t_f \quad (2.56d)$$

em que $\varsigma \in \mathbb{R}, \varsigma > 0$ é uma perturbação na variável tempo.

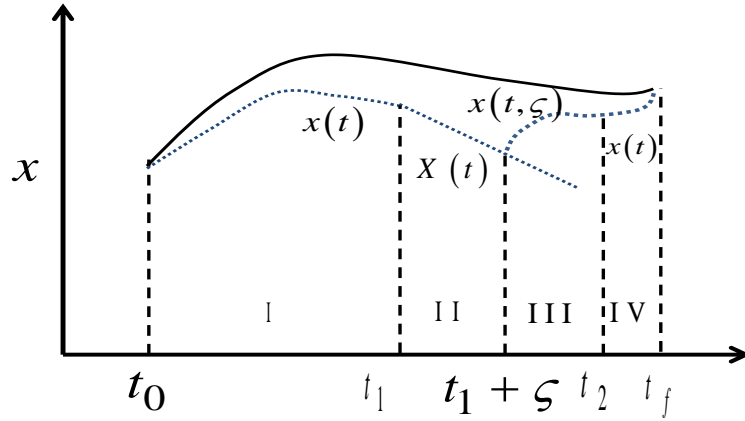


Figura 2.3. Ilustração: variação forte na região II (baseado em CINTRON, 1969).

Supõe-se que no intervalo II uma variação forte é permitida, isto é, $\|\dot{X}(t, \varepsilon) - \dot{X}(t)\| \leq \varepsilon$ não é uma condição necessária. Note que $X(t)$ independe de ζ , enquanto que na terceira região $x(t, \zeta)$ depende de ζ .

Novamente, o objetivo é minimizar a função objetivo (em princípio sem variável de controle):⁴

$$\mathfrak{J}(\zeta) = \int_{t_0}^{t_f} H(t, x(t), \dot{x}(t)) dt \quad (2.57)$$

A integral é dividida em quatro regiões:

$$\begin{aligned} \mathfrak{J}(\zeta) = & \int_{t_0}^{t_1} H(t, x(t), \dot{x}(t)) dt + \int_{t_1}^{t_1+\zeta} H(t, X(t), \dot{X}(t)) dt + \\ & \int_{t_1+\zeta}^{t_2} H(t, x(t, \zeta), \dot{x}(t, \zeta)) dt + \int_{t_2}^{t_f} H(t, x(t), \dot{x}(t)) dt \end{aligned} \quad (2.58)$$

Considerando que a variável $x(t)$, mas não $X(t)$, é a variável que minimiza $\mathfrak{J}(\zeta)$, logo:

$$d\mathfrak{J}(0) = \left. \frac{d\mathfrak{J}}{d\zeta} \right|_{\zeta=0} \quad \zeta \geq 0 \quad (2.59)$$

Calculando $\frac{d\mathfrak{J}}{d\zeta}$:

⁴ Note que aqui, para fins de simplificação, não se considera as condições de fronteira. Porém, a dedução para condições de fronteira livre gera o mesmo resultado (CINTRON, 1969).

$$\begin{aligned} \frac{d\mathfrak{J}(\zeta)}{d\zeta} &= \left[H(t, X(t), \dot{X}(t)) - H(t, x(t, \zeta), \dot{x}(t, \zeta)) \right] \Big|_{t_1+\zeta} \\ &+ \int_{t_1+\zeta}^{t_2} \left(\frac{\partial H}{\partial x} \frac{\partial x}{\partial \zeta}(t, \zeta) + \frac{\partial H}{\partial \dot{x}} \frac{\partial \dot{x}}{\partial \zeta}(t, \zeta) \right) dt \end{aligned} \quad (2.60)$$

e integrando o segundo termo por partes, resulta em:

$$\begin{aligned} \frac{d\mathfrak{J}(\zeta)}{d\zeta} &= \left[H(t, X(t), \dot{X}(t)) - H(t, x(t, \zeta), \dot{x}(t, \zeta)) \right] \Big|_{t_1+\zeta} \\ &+ \frac{\partial H}{\partial \zeta}(t, x(t, \zeta), \dot{x}(t, \zeta)) \frac{\partial x}{\partial \zeta}(t, \zeta) \Big|_{t_1+\zeta}^{t_2} - \int_{t_1+\zeta}^{t_2} \left(\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} - \frac{\partial H}{\partial x} \right) \frac{\partial x}{\partial \zeta}(t, \zeta) dt \end{aligned} \quad (2.61)$$

Como a equação de Euler-Lagrange implica em:

$$\frac{d}{dt} \frac{\partial H}{\partial \dot{x}} - \frac{\partial H}{\partial x} = 0$$

Logo, avaliando a Equação (2.61) em $\zeta = 0$ temos:

$$\frac{d\mathfrak{J}(0)}{d\zeta} = \left[H(t, X(t), \dot{X}(t)) - H(t, x(t), \dot{x}(t)) \right] \Big|_{t_1} + \frac{\partial H}{\partial \dot{x}}(t, x(t), \dot{x}(t)) \frac{\partial x}{\partial \zeta}(t, 0) \Big|_{t_1}^{t_2} \quad (2.62)$$

Da Figura 2.3 é possível afirmar que:

$$x(t_1) = X(t_1) \quad (2.63a)$$

$$X(t_1 + \zeta) = x(t_1 + \zeta, \zeta) \quad (2.63b)$$

$$x(t_2, \zeta) = x(t_2) \quad (2.63c)$$

Por consequência, conclui-se que:

$$\frac{\partial x}{\partial \zeta}(t_2, 0) = 0 \quad (2.64)$$

e

$$\frac{\partial x}{\partial \zeta}(t_1, 0) = \dot{X}(t_1) - \dot{x}(t_1) \quad (2.65)$$

Desse modo, a Equação (2.62) é redefinida como:

$$\frac{d\mathfrak{J}(0)}{d\zeta} = H(t, X(t), \dot{X}(t)) - H(t, x(t), \dot{x}(t)) - \frac{\partial H}{\partial \dot{x}}(t, x(t), \dot{x}(t)) (\dot{X}(t) - \dot{x}(t)) \Big|_{t_1} \quad (2.66)$$

Observa-se que a Equação (2.66) é exatamente o coeficiente da Equação (2.59).

ou seja:

$$d\mathfrak{J}(0) = \underbrace{\frac{d\mathfrak{J}}{d\zeta}}_{\text{Equação 2.66}} \Big|_{\zeta=0} \zeta \geq 0 \quad (2.59)$$

Logo:⁵

$$\begin{aligned} \frac{d\mathfrak{J}(0)}{d\zeta} &= H(t, X(t), \dot{X}(t)) - H(t, x(t), \dot{x}(t)) \\ &\quad - \frac{\partial H}{\partial \dot{x}}(t, x(t), \dot{x}(t))(\dot{X}(t) - \dot{x}(t)) \geq 0 \end{aligned} \quad (2.67)$$

A Equação (2.67) é definida como função de Weierstrass W . Tal equação deve ser satisfeita para todo \dot{X} consistente com o modelo dinâmico $\mathbf{h}(t, X, \dot{X}) = 0$. Note que para um problema multidimensional, a Equação (2.67) é escrita como:

$$\begin{aligned} W &= H(t, X(t), \dot{X}(t)) - H(t, x(t), \dot{x}(t)) - \\ &\quad \sum_{j=1}^{n_x} \frac{\partial H}{\partial \dot{x}_j}(t, x(t), \dot{x}(t))(\dot{X}_j(t) - \dot{x}_j(t)) \geq 0 \end{aligned} \quad (2.68)$$

A condição para $W \geq 0$ é conhecida como condição de Weierstrass.

2.2.8 Condição de Weierstrass: Generalização

A condição de Weierstrass pode ser aplicada à formulação desenvolvida no Tópico 2.2.5. Em particular, sabendo-se que a função M é escrita como:

$$M = \sum_{i=1}^{n_x} \lambda_i(t) [\dot{x}_i - f_i(t, \mathbf{x}, \mathbf{u})] \quad (2.47)$$

A condição de Weierstrass, considerando que $M \cong 0$, resulta em:

$$W = - \sum_{i=1}^{n_x} \lambda_i(t) (\dot{X}_i(t) - \dot{x}_i(t)) - \sum_{j=1}^{n_u} \left(- \sum_{i=1}^{n_x} \lambda_i(t) \frac{\partial f_i}{\partial u_j} \right) (U_j(t) - u_j(t)) \geq 0 \quad (2.69)$$

em que U é a variação forte da variável de controle. Como as equações de controle prenunciam que:

$$\sum_{i=1}^{n_x} \lambda_i \frac{\partial f_i}{\partial u_j}(t, \mathbf{x}, \mathbf{u}) = 0, \quad j = 1, \dots, n_u \quad (2.49)$$

Por consequência, o segundo termo da Equação (2.69) é eliminado, resultando em:

$$W = \sum_{i=1}^{n_x} \lambda_i(t) \dot{x}_i(t) \geq \sum_{i=1}^{n_x} \lambda_i(t) \dot{X}_i(t) \quad (2.70)$$

⁵ Em que t_1 é omitido por ser arbitrário.

Sabendo-se que $\dot{X}_i(t) = f_i(t, \mathbf{x}, \mathbf{U})$ e $\dot{x}_i(t) = f_i(t, \mathbf{x}, \mathbf{u})$, a Equação (2.70) é escrita como:

$$\sum_{i=1}^{n_x} \lambda_i(t) f_i(t, \mathbf{x}, \mathbf{u}) \geq \sum_{i=1}^{n_x} \lambda_i(t) f_i(t, \mathbf{x}, \mathbf{U}) \quad (2.71)$$

Claramente, a condição de Weierstrass torna-se mais óbvia na Equação (2.71). As variáveis de controle \mathbf{u} devem maximizar o lado esquerdo da Equação (2.71). Já, do lado direito, qualquer variação da variável de controle \mathbf{U} pode gerar resultados piores. Redefinindo a função Hamiltoniano como:

$$\hbar = \sum_{i=1}^{n_x} \lambda_i(t) f_i(t, \mathbf{x}, \mathbf{u}) \quad (2.72)$$

Conclui-se que:

$$\hbar(t, \mathbf{x}, \mathbf{u}) \geq \hbar(t, \mathbf{x}, \mathbf{U}) \quad (2.73)$$

Desse modo, a condição necessária para maximização de \hbar é:

$$\left. \frac{\partial \hbar}{\partial u_j} \right|_{\mathbf{u}^*} = 0 \quad j = 1, \dots, n_u \quad (2.74)$$

resultando em:

$$\hbar(t, \mathbf{x}^*, \mathbf{u}^*) \geq \hbar(t, \mathbf{x}, \mathbf{u}) \quad (2.75)$$

em que \mathbf{x}^* e \mathbf{u}^* são as trajetórias ótimas de estado e controle. Consequentemente, as equações de controle podem ser escritas como:

$$\sum_{i=1}^{n_x} \lambda_i(t) \left. \frac{\partial f_i}{\partial u_j} \right|_{\mathbf{u}^*} = 0 \quad j = 1, \dots, n_u \quad (2.76)$$

Para assegurar que os perfis de controle da Equação (2.76) sejam as próprias equações de controle que maximizam o Hamiltoniano, aplica-se a segunda derivação sobre o Hamiltoniano (condições necessárias de segunda ordem):

$$\sum_{j=1}^{n_u} \sum_{i=1}^{n_u} \left. \frac{\partial^2 \hbar}{\partial u_j \partial u_i} \right|_{\mathbf{u}^*} \delta u_j \delta u_i < 0 \quad (2.77)$$

que são equivalentes à condição de Legendre-Chebyshev (ATHANS e FALB, 1966).

Cabe enfatizar que, geralmente nos problemas de otimização dinâmica é muito difícil assegurar a existência de uma solução viável, ou seja, as condições de

otimalidade são apenas instrumentos para detectar possíveis soluções ótimas, mas não garantem a existência da melhor solução. As condições suficientes de otimalidade são na maioria das vezes impossíveis de serem aplicadas (ATHANS e FALB, 1966). CHACHUAT (2006) ilustra alguns exemplos em que não há solução ótima, ou até mesmo o problema possui múltiplas soluções.

O próximo capítulo aprofunda os conceitos definidos aqui, ao aplicar as condições de otimalidade sobre problemas mais realistas, ou seja, modelos algébrico-diferenciais sujeitos a restrições. Além do PMP, serão introduzidas também estratégias numéricas para solução de problemas de otimização dinâmica.

Capítulo 3

Modelagem e Otimização Dinâmica de Sistemas Algébrico-Diferenciais

Neste Capítulo
Tópico 3.1. <u>Sistemas Algébrico-Diferenciais</u>
Tópico 3.2. <u>Métodos de Solução</u>
Tópico 3.3. <u>Considerações Adicionais sobre DAOPs</u>

O objetivo deste capítulo é discutir aspectos de otimização dinâmica de sistemas algébrico-diferenciais, resultantes da modelagem de processos químicos. O **Tópico 3.1** introduz as principais características de sistemas DAEs e métodos de solução. O **Tópico 3.2** apresenta os principais métodos de otimização dinâmica. Já o **Tópico 3.3** reporta as principais rotinas de DAOPS utilizados atualmente e enfatiza as principais inovações recentes.

3.1. Sistemas Algébrico-Diferenciais (DAEs)

3.1.1. Aspectos Gerais de Modelagem

Geralmente, a modelagem de processos químicos resulta em sistemas de equações algébrico-diferenciais (DAEs), que podem ser constituídos por equações diferenciais ordinárias (ODEs) e equações algébricas, formulados como problemas de valor inicial ou problemas de valor de contorno (RICE e DO, 1994).

A modelagem de processos que lida com propriedades espacialmente distribuídas, como transporte de energia e escoamento de fluidos, resulta em sistemas de equações diferenciais parciais (PDEs). Há poucos trabalhos na literatura que abordam otimização dinâmica de sistemas de PDEs, porém, dependendo da estratégia de solução, o sistema pode resultar em DAEs, se aplicada, por exemplo, a discretização das variáveis espaciais e integração no tempo (como no método das linhas) (BRENAN *et al.*, 1996).

Sistemas de DAEs são representados semi-explicitamente como:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.1a)$$

$$0 = \ell(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}) \quad (3.1b)$$

em que $\mathbf{x}(t) \in \mathbb{R}^{n_x}$ são as variáveis de estado diferenciais⁶, $\mathbf{u}(t) \in \mathbb{R}^{n_u}$ são as variáveis de controle (manipuladas), $\mathbf{y}(t) \in \mathbb{R}^{n_y}$ são as variáveis algébricas e $\mathbf{p} \in \mathbb{R}^{n_p}$ são parâmetros independentes do tempo.

Nos casos em que $\partial \ell / \partial \mathbf{y}$ for não singular para todos os valores de $\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t)$ e \mathbf{p} , é possível representar explicitamente $\mathbf{y}(t) = \mathbf{y}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p})$, resultando em um sistema de ODEs (BRENAN *et al.*, 1996):

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{y}(\mathbf{x}(t), \mathbf{u}(t), \mathbf{p}), \mathbf{u}(t), \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.2)$$

Uma das características de um sistema de DAEs é o índice diferencial, que significa o número de vezes que o sistema ou subconjunto desse sistema, deve ser diferenciado em relação à variável independente, de modo que se transforme em um sistema de ODEs (que possui índice diferencial zero), ou seja:

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.3a)$$

$$\dot{\mathbf{y}} = \ell_s(\mathbf{x}(t), \mathbf{y}(t), \mathbf{u}(t), \mathbf{p}), \quad \mathbf{x}(0) = \mathbf{x}_0 \quad (3.3b)$$

em que ℓ_s resulta no sistema de equações diferenciais resultante de s diferenciações do sistema 3.1 em relação à variável independente.

Embora o sistema reformulado (Equação 3.3) possa ser solucionado por rotinas de integração de ODEs, é necessário determinar condições iniciais consistentes com as equações algébricas e as equações resultantes do processo de diferenciação para obtenção do sistema (3.3). Logo, sob o ponto de vista numérico, a solução dos modelos descritos por sistemas de DAEs depende de uma análise prévia do seu índice diferencial e da definição das condições iniciais consistentes, que são adequadas para inicializar a integração.

3.1.2. Métodos Numéricos para DAEs

As estratégias numéricas de solução de DAOPs requerem a solução de sistemas de DAEs, ao longo do processo de otimização. Existem duas classes específicas de

⁶ Nota-se que aqui dividimos variáveis diferenciais e algébricas, diferentemente do Capítulo 2.

métodos para solução de ODEs representadas pela Equação 3.3: os métodos de passo único e os métodos de passos múltiplos. Os métodos de passo único utilizam apenas informação dentro do intervalo $[t_i, t_{i+1}]$ para o cálculo do ponto $x(t_{i+1})$ a partir de um ponto prévio $x(t_i)$. São exemplos de métodos pertencentes a essa classe, o método de Euler e o método de Runge-Kutta. Já os métodos de passos múltiplos utilizam informações anteriores em mais de um ponto para determinar a aproximação do ponto seguinte. Como exemplo desse tipo de método pode-se citar o método BDF (*Backward Differentiation Formula*) (RICE e DO, 1994).

3.1.2.1. Métodos de Passo Único

Para o sistema semi-implícito (Equações (3.1a) e (3.1b)), supondo $n_x = 1$ e $n_y = 1$, desconsiderando-se as variáveis de controle e parâmetros), os métodos de passo único (Euler e Runge-Kutta) resultam em:

$$x_{i+1} = x_i + \Delta t_i \sum_{k=1}^{n_e} b_k f(t_i + c_k \Delta t_i, \hat{x}_k, \hat{y}_k) \quad (3.4a)$$

$$\hat{x}_k = x_i + \Delta t_i \sum_{j=1}^{n_{rk}} a_{kj} f(t_i + c_j \Delta t_i, \hat{x}_j, \hat{y}_j) \quad (3.4b)$$

$$0 = \ell(\hat{x}_k, \hat{y}_k), \quad k = 1, \dots, n_e \quad (3.4c)$$

em que $t_{i+1} = t_i + \Delta t_i$, sendo $\Delta t_i \in \mathbb{R}$, $i = 0, \dots, N$ o passo de integração, $x(t_i)$ (solução exata) é aproximada por x_i (solução numérica), n_e é o número de passos de discretização, \hat{x} e \hat{y} são soluções intermediárias, $n_{rk} > k$, $c_k \in [0, 1]$. :

$$\sum_{k=1}^{n_e} b_k = 1 \quad (3.5)$$

e

$$\sum_{j=1}^{n_e} a_{kj} = c_k \quad (3.6)$$

Os coeficientes a , b e c são obtidos através da expansão em série de Taylor das Equações (3.5a) e (3.5b), de modo a satisfazer as equações de ordem do método. A estrutura do método geralmente é representada pelo arranjo de Butcher:

$$\begin{array}{c|c} c & \mathbf{A} \\ \hline \mathbf{b}^T & \end{array} \quad (3.7)$$

em que \mathbf{A} é uma matriz característica e \mathbf{c} e \mathbf{b} são vetores, que dependem da ordem do método (RICE e DO, 1994).

Por exemplo, o método de Runge-Kutta Implícito de segunda ordem (Método de Euler implícito do ponto médio) resulta em:

$$\begin{array}{c|c} 1/2 & 1/2 \\ \hline & 1 \end{array} \quad (3.8)$$

em que:

$$x_{i+1} = x_i + \Delta t_i f \left(t_i + \frac{\Delta t_i}{2}, \hat{x}_1, \hat{y}_1 \right) \quad (3.9a)$$

$$\hat{x}_1 = x_i + \frac{\Delta t_i}{2} f \left(t_i + \frac{\Delta t_i}{2}, \hat{x}_1, \hat{y}_1 \right) \quad (3.9b)$$

$$0 = \ell(\hat{x}_1, \hat{y}_1) \quad (3.9c)$$

Se o sistema de DAEs possui índice diferencial um as equações algébricas podem ser representadas implicitamente no sistema. Logo, as propriedades de estabilidade do método numérico acima seguem as mesmas regras verificadas para os métodos aplicados a sistemas de ODEs (BIEGLER, 2010).

Para garantir estabilidade, certamente o passo Δt_i deve ser configurado de maneira a resultar em um erro local desprezível em relação à variação de x . No entanto, o passo também pode ser limitado pela estabilidade, apesar do erro ser pequeno. Tal ocorrência caracteriza sistemas de DAEs rígidos (*stiff systems*). Isso ocorre principalmente em métodos explícitos. Por exemplo:

$$\dot{x} = -\lambda x \quad (3.10)$$

em que $\lambda \gg 1$. A versão discretizada do problema é formulada como:

$$x_{i+1} = x_i - \Delta t_i \lambda x_i = x_i (1 - \Delta t_i \lambda) \quad (3.11)$$

O sistema se torna instável se $\Delta t_i > 2\lambda^{-1}$, o que resulta em valores muito pequenos de Δt_i para valores muito grandes de λ (BIEGLER, 2010).

Uma alternativa para solucionar problemas rígidos é utilizar algoritmos numéricos de integração que sejam implícitos, que são geralmente mais estáveis por não restringirem ao tamanho do passo de integração. Outra estratégia possível é a utilização

de métodos de retro-diferenciação (BDF) que são classificados como métodos de passos múltiplos (RICE e DO, 1994).

3.1.2.2. Métodos de Passos Múltiplos

O método de retro-diferenciação (BDF) aplicado ao mesmo sistema de DAEs (Equação 3.1) pode ser escrito como:

$$\sum_{j=1}^{n_e} \alpha_j x_{i-j+1} + \Delta t \sum_{j=1}^{n_e} \beta_j f(x_{i-j+1}, y_{i-j+1}) \quad (3.12a)$$

$$0 = \ell(x_{i+1}, y_{i+1}) \quad (3.12b)$$

em que α_j e β_0 são os coeficientes do método BDF, resultando em um sistema algébrico implícito. Geralmente o método de Newton (ou semelhante) é utilizado para solucionar o sistema não linear gerado (BRENAN *et al.*, 1996).

Uma das dificuldades de implementação de métodos de passos múltiplos é a necessidade do conhecimento de valores de x no início do passo e também em passos anteriores. Porém, no início do processo integração o valor de x é apenas conhecido no início do passo obrigando assim ao método de passos múltiplos ser o de menor ordem possível (método de Euler). Para superar esse problema algumas estratégias são adotadas, como começar o processo de integração com um método de passo simples (Euler), no segundo passo adotar um método de dois passos e assim sucessivamente até atingir o número de passos desejado. Esse procedimento pode tornar a convergência mais lenta que a convergência de algoritmos de passo único.

3.2. Métodos de Solução de DAOPs

A literatura recente divide os métodos em duas classes: os métodos indiretos e os métodos diretos. Nos métodos diretos, aplicam-se diretamente os fundamentos da programação não linear ao problema de otimização. Tais métodos podem fornecer soluções sub-ótimas e resultados menos acurados quando comparados aos métodos indiretos (BIEGLER, 2010).

Os métodos de solução de DAOPs se dividem conforme a Figura 3.1.

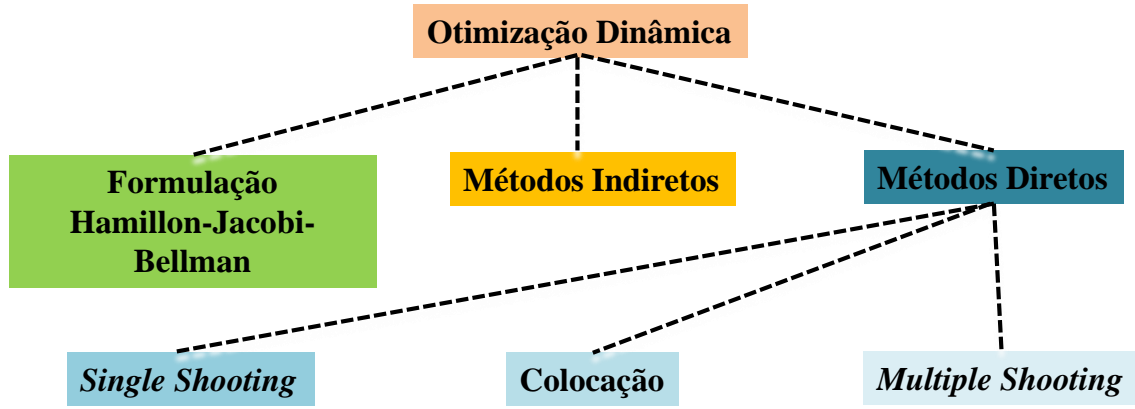


Figura 3.1. Métodos de solução de DAOPs.

3.2.1. Programação Dinâmica e Formulação de Hamilton-Jacobi-Bellman (HJB)

O método de programação dinâmica foi desenvolvido por Richard Bellman (BELLMAN, 1957) para solução de problemas dinâmicos de modelos discretos. Considere uma simplificação da formulação apresentada no Capítulo 2:

$$\min_{u(t)} \mathfrak{J} = \min_{u(t)} \left[\underbrace{\varphi(t_f, x(t_f))}_{\text{Termo de Meyer}} + \underbrace{\int_{t_0}^{t_f} L(x(t), u(t)) dt}_{\text{Termo de Lagrange}} \right] \quad (3.13)$$

sujeito a:

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x_0(t) = x_0(t_0) \quad (3.14)$$

A versão discretizada do problema pode ser escrita como:

$$\min_{\mathbf{u}} \mathfrak{J} = \min_{\mathbf{u}} \left[\underbrace{\varphi(t_f, x_{n_s})}_{\text{Termo de Meyer}} + \underbrace{\Delta t \sum_{k=0}^{n_s-1} L(x_k, u_k)}_{\text{Termo de Lagrange}} \right] \quad (3.15)$$

sujeito a:

$$\begin{aligned} x_{k+1} &= x_k + \Delta t f(x_k, u_k), \quad k = 0, \dots, n_s - 1 \\ x_0 &= \bar{x}_0 \end{aligned} \quad (3.16)$$

em que $\Delta t = (t_f - t_0)/n_s$ e n_s é o número de estágios de discretização.

Na programação dinâmica o horizonte de tempo é dividido em n_s estágios. Logo, a variável de controle é então discretizada gerando uma malha equidistante, sob todo o

intervalo de tempo. Para cada estágio, o algoritmo deve encontrar o perfil de controle que minimize uma dada função objetivo quantificada no intervalo determinado.

O princípio de otimalidade estabelece que cada subtrajetória é ótima. Ou seja, para cada $k \in \{0, \dots, n_s - 1\}$:

$$\mathfrak{J}_k = \mathfrak{J}(x, t_k) = \min_u \Delta t L(x, u) + \mathfrak{J}(x + \Delta t f(x, u), t_k + \Delta t) \quad (3.17)$$

onde $t_k = \Delta t \cdot k$.

Esse procedimento possibilita computar cada função \mathfrak{J}_k sequencialmente a partir de $n_s - 1$ e reduzindo k até a obtenção de J_0 .

Assumindo que $\mathfrak{J}(x, t)$ é diferenciável em (x, t) , a expansão em série de Taylor de $\mathfrak{J}(x, t)$ é escrita como:

$$\mathfrak{J}(x, t) = \min_u \Delta t L(x, u) + \mathfrak{J}(x, t) + \Delta t \nabla_x \mathfrak{J}(x, t)^T f(x, u) + \Delta t \frac{\partial \mathfrak{J}}{\partial t} + O(\Delta t^2) \quad (3.18)$$

Ao separar todos os termos independentes de u para a esquerda e dividindo por $\Delta t \rightarrow 0$ obtém-se a equação de HJB:

$$-\frac{\partial \mathfrak{J}}{\partial t}(x, t) = \min_u L(x, u) + \nabla_x \mathfrak{J}(x, t)^T f(x, u) \quad (3.19)$$

Essa equação diferencial parcial descreve a evolução da função objetivo ao longo do tempo. Logo, soluciona-se a equação a partir do instante final, $t \in [t_0, t_f]$, sendo que a condição final é conhecida:

$$\mathfrak{J}(x, t) = \varphi(t_f, x(t_f)) \quad (3.20)$$

Consequentemente o controle ótimo em um estado x e instante t é definido como:

$$u^*(x, t) = \operatorname{argmin}_u L(x, u) + \nabla_x \mathfrak{J}(x, t)^T f(x, u) \quad (3.21)$$

Definindo $\nabla_x \mathfrak{J}$ como $\lambda \in \mathbb{R}^{n_x}$, na Equação (3.21), define-se a função Hamiltoniano:

$$\hat{h}(x, \lambda, u) = L(x, u) + \lambda^T f(x, u) \quad (3.22)$$

Logo:

$$u^*(x, \lambda, t) = \operatorname{argmin}_u \hat{h}(x, \lambda, u) \quad (3.23)$$

Obtendo-se u^* , computa-se o Hamiltoniano ótimo:

$$\hat{h}^*(x, \lambda) = \min_u \hat{h}(x, \lambda, u) = \hat{h}(x, \lambda, u^*) \quad (3.24)$$

Dessa maneira, utilizando $\hat{h}^*(x, \lambda)$ a equação de HJB pode ser escrita compactamente como:

$$-\frac{\partial \mathfrak{J}}{\partial t}(x, t) = \hat{h}^*(x, \lambda) \quad (3.25)$$

A programação dinâmica foi aplicada para solucionar inúmeros exemplos de controle ótimo de processos batelada como (LUUS e ROSEN, 1991; BOJKOV e LUUS, 1996). Mais recentemente TOSUKHOWONG e LEE (2009) aplicaram esse método para otimização de uma planta que integra um reator PFR e uma torre de destilação contínua.

Os pesquisadores evidenciam que a solução numérica da equação de HJB é bastante custosa para problemas com um número elevado de variáveis de estado e controle. Adicionalmente, a diferenciabilidade da função objetivo nem sempre é garantida assim como a existência de solução (DIEHL 2011).

3.2.2. Métodos Indiretos

Os métodos indiretos fundamentam-se no Princípio de Máximo de Pontryagin (PMP) a qual se baseia na maximização da função Hamiltoniano (Equação 3.22) e conduz de maneira indireta à maximização (ou minimização) da função objetivo (PONTYAGIN *et al.*, 1962). Esses métodos surgiram com o desenvolvimento do Cálculo Variacional, permitindo a dedução das condições necessárias e suficientes da teoria de otimização para a solução de problemas de otimização dinâmica (BRYSON e HO, 1975).

Para o desenvolvimento do PMP considere o DAOP do tópico anterior:

$$\min_{u(t)} \mathfrak{J} = \min_{u(t)} \left[\underbrace{\varphi(x(t_f))}_{\text{Termo de Meyer}} + \underbrace{\int_{t_0}^{t_f} L(x(t), u(t)) dt}_{\text{Termo de Lagrange}} \right] \quad (3.26)$$

sujeito a

$$\dot{x}(t) = f(t, x(t), u(t)), \quad x_0(t) = x_0 \quad (3.27)$$

sabendo-se que a função Hamiltoniano é definida como:

$$\hat{h}(x, \lambda, u) = L(x, u) + \lambda^T f(x, u) \quad (3.28)$$

e a equação de HJB:

$$-\frac{\partial \mathfrak{J}}{\partial t}(x, t) = \hat{h}^*(x, \lambda) = \min_u \hat{h}(x, \lambda, u) \quad (3.29)$$

O PMP propõe a obtenção analítica de x, λ e u . Para isso, assume-se que a trajetória ótima é conhecida e diferencia-se a equação de HJB em relação a x ao longo da trajetória, ou seja:

$$\frac{-\partial^2 \mathfrak{J}}{\partial t \partial x}(x^*, t) = \frac{\partial \hat{h}}{\partial x}(x^*, \lambda^*, u^*) + \underbrace{\frac{\partial \hat{h}}{\partial \lambda}(x^*, \lambda^*, u^*) \nabla_x^2 J(x^*, t)}_{=f(x^*, u^*)^T} + \frac{\partial \hat{h}}{\partial u}(x^*, \lambda^*, u^*) \quad (3.30)$$

$$\text{Como } \frac{\partial \hat{h}}{\partial u}(x^*, \lambda^*, u^*) = 0,$$

$$\frac{-\partial^2 \mathfrak{J}}{\partial t \partial x}(x^*, t) = \frac{\partial \hat{h}}{\partial x}(x^*, \lambda^*, u^*) + \underbrace{\frac{\partial \hat{h}}{\partial \lambda}(x^*, \lambda^*, u^*) \nabla_x^2 \mathfrak{J}(x^*, t)}_{=f(x^*, u^*)^T} \quad (3.31)$$

Com $\dot{x}^* = f(x^*, u^*)$ reescreve-se a Equação (3.31) como:

$$\underbrace{\frac{\partial}{\partial t} \nabla_x \mathfrak{J}(x^*, t) + \nabla_x^2 \mathfrak{J}(x^*, t) \dot{x}^*}_{\frac{d}{dt} \nabla_x \mathfrak{J}(x^*, t)} = -\nabla_x \hat{h}(x^*, \lambda^*, u^*) = \dot{\lambda}^* \quad (3.32)$$

que é uma equação diferencial para da variável adjunta λ .

Da mesma maneira a condição terminal é diferenciada em relação a x :

$$\nabla_x(x(t_f)) = \lambda(t_f) \quad (3.33)$$

Por consequência, as Equações (3.32) e (3.33) são incorporadas ao DAOP original, resultando em um problema de duplo valor de contorno para $t \in [t_0, t_f]$:

$$\dot{x}^*(t) = f(x^*, u^*) \quad (3.34a)$$

$$x^*(t_0) = x_0 \quad (3.34b)$$

$$\dot{\lambda}^*(t) = -\nabla_x \hat{h}(x^*, \lambda^*, u^*) \quad (3.34c)$$

$$u^*(t) = \underset{u}{\operatorname{argmin}} \hat{h}(x^*, \lambda^*, u^*) \quad (3.34d)$$

$$\lambda(t_f) = \nabla_x(\varphi(t_f)) \quad (3.34e)$$

O PMP descreve as condições necessárias de otimalidade, resultando em um problema de duplo valor de contorno (TBVP – *Two-point Boundary Value Problem*) (VILLADSEN e STEWART, 1967). Há diversas metodologias aplicáveis para solucionar o TBVP, como o método *single-shooting*, *multiple-shooting* ou métodos de discretização como colocação em elementos finitos (BIEGLER, 2010).

Note que o sistema (3.34) não considera restrições de desigualdade e restrições finais sobre o problema. A generalização do PMP para esse caso baseia-se na redefinição da função Hamiltoniano, com inclusão das restrições:

$$\dot{h}(x, \lambda, u) = L(x, u) + \lambda(t) f(x, u) + \mu(t) g(x, u) \quad (3.35)$$

em que $g(x, u) \leq 0$ são as restrições de desigualdade e $\mu \in \mathbb{R}^{n_g}$ são os multiplicadores de Lagrange associados às restrições. Nesse caso:

$$\dot{\lambda}(t)^* = -\nabla_x \dot{h}(x^*, \lambda^*, u^*) = -\lambda \frac{\partial f}{\partial x} - \mu \frac{\partial g}{\partial x} \quad (3.36)$$

e

$$\lambda(t_f) = \left. \frac{\partial \varphi}{\partial x} \right|_{t_f} + \nu \left. \frac{\partial e}{\partial x} \right|_{t_f} \quad (3.37)$$

em que $\nu \in \mathbb{R}^{n_e}$ são os multiplicadores de Lagrange associados às restrições finais:

$$e(t_f, x(t_f)) = 0 \quad (3.38)$$

Adicionalmente surgem as condições de complementaridade (NOCEDAL e WRIGHT, 1999):

$$\mu(t) g(x^*, \lambda^*, u^*) = 0 \quad (3.39)$$

$$\nu(t) e(t_f, x(t_f)) = 0 \quad (3.40)$$

Em resumo, as condições de otimalidade, juntamente com o sistema de Equações (3.34), se baseiam nas seguintes condições:

$$\frac{\partial \dot{h}(x, \lambda, \mu)}{\partial u} = \lambda(t) \frac{\partial f}{\partial u} + \mu(t) \frac{\partial g}{\partial u} = 0 \quad (3.41a)$$

$$\mu(t) \begin{cases} = 0, \text{ se } g(x^*, \lambda^*, u^*) < 0 \\ > 0, \text{ se } g(x^*, \lambda^*, u^*) \geq 0 \end{cases} \quad (3.41b)$$

$$v(t) \begin{cases} = 0, \text{ se } e(x^*, \lambda^*, u^*) < 0 \\ > 0, \text{ se } e(x^*, \lambda^*, u^*) = 0 \end{cases} \quad (3.41c)$$

SOUZA (2007) discute as principais vantagens e desvantagens dos métodos indiretos. Em seu trabalho afirma-se que tais métodos garantem o atendimento às condições de otimalidade e restrições do problema, porém possuem inconvenientes como a convergência lenta para o ótimo, o desconhecimento da ordem de grandeza das variáveis adjuntas, o tratamento das restrições e as singularidades.

A origem das singularidades em problemas de controle ótimo pode ser atribuída à dependência linear da função Hamiltoniano em relação à variável de controle que, sob certas condições, deixa de ser uma função explícita, não havendo informação sobre a tendência da variável ao longo do tempo. Tais singularidades ocorrem em um instante do tempo não conhecido e o lugar geométrico em que ocorre é chamado de arco singular. Como consequência uma nova equação algébrica referente ao arco singular deve ser acrescentada ao sistema original, podendo resultar em uma variação do índice diferencial do sistema de equações.

Na ocorrência de singularidades a variável de controle não afeta diretamente o Hamiltoniano, ou seja, a condição $\nabla_u \dot{h} = 0$ é trivialmente satisfeita em qualquer instante de tempo. Logo, a função Hamiltoniano também não sofre variação com o tempo, ou seja:

$$\frac{d}{dt} \nabla_u \dot{h} = 0 \quad (3.42)$$

que é equivalente a:

$$\frac{\partial}{\partial x} \frac{\partial \dot{h}}{\partial u} \dot{x} + \frac{\partial}{\partial \lambda} \frac{\partial \dot{h}}{\partial u} \dot{\lambda} = 0 \quad (3.43)$$

Ao substituir as expressões de \dot{x} e $\dot{\lambda}$ na Equação (3.43) espera-se que a variável de controle seja explicitada. Caso contrário, o procedimento de diferenciação em relação ao tempo é repetido novamente até que a variável de controle seja explicitada, ou seja:

$$\left(\frac{d}{dt}\right)^n \nabla_u \dot{h} = 0 \quad (3.44)$$

A abordagem utilizada por VIEIRA (2001) e SOUZA (2007) consiste em explorar a linearidade do Hamiltoniano, quando possível, em relação à variável de controle e às variáveis adjuntas. A estratégia permite eliminar as variáveis adjuntas do sistema e identificar a solução analítica do perfil de controle em arcos específicos, faltando apenas identificar os pontos de junção.

QUINTO (2010) propôs a solução analítica de DAOPs com a incorporação automática das restrições às variáveis de controle e/ou às variáveis de estado com o emprego de funções de regularização. A abordagem associa o procedimento de eliminação das variáveis adjuntas, portanto evitando a solução de um problema de valor de contorno. Porém, o método limita-se a problemas de pequena dimensão, pois a determinação das expressões analíticas das trajetórias ótimas em problemas que possuam um número elevado de variáveis de estado e controle torna-se inviável.

Uma solução analítica completa para o problema de otimização não é possível exceto para casos muito simples. No entanto, expressões analíticas podem ser computadas por partes da solução ótima, que podem ser utilizadas, por exemplo, para identificar os arcos que constituem a trajetória de controle ou aumentar a eficiência do algoritmo de otimização numérica, uma vez que através dessas se pode conhecer de antemão a trajetória da variável de controle (BETTS, 2010).

Alguns pesquisadores, como BAYON *et al.* (2010), utilizaram o PMP para gerar como estimativas iniciais para soluções numéricas de DAOPs irrestritos. O procedimento se baseia na solução via PMP para estimativa inicial do perfil de controle a ser utilizado em métodos de discretização (métodos diretos). Dessa maneira, é possível garantir maior independência em relação à estimativa inicial numérica.

Alguns autores (FAHOO e ROSS, 2008) mostraram que a convergência das variáveis adjuntas e de estado não implica necessariamente convergência das variáveis de controle. Em outras palavras, há dúvidas se a solução numérica do TPBV resultante é confiável para alguns casos. Por exemplo, ROSS (2006) obteve resultados contraditórios

ao obter a solução ótima via métodos diretos e não conseguir solucionar (convergir) o problema via método indireto.

(BETTS, 2010) sumariza as três principais dificuldades que limitam a aplicação do PMP para solução de DAOPs, dando margem ao uso de métodos numéricos:

- (i) O cômputo do Hamiltoniano e aplicação das condições necessárias de otimalidade requerem conhecimento avançado da teoria de controle ótimo, além de se tornar uma tarefa complexa para modelos de elevada dimensão.
- (ii) Se o problema for restrito por restrições de trajetória torna-se necessário estimar a sequência de arcos *a priori*. A formulação do TBVP resultante pode ser diferente em cada arco de controle, e se a estimativa for inadequada, a solução pode ser computada erroneamente.
- (iii) Outra dificuldade é a necessidade de se estimar os valores iniciais das variáveis adjuntas, que não são variáveis físicas. A experiência numérica de alguns pesquisadores (BRYSON e HO, 1975; BETTS, 2010) mostra que a função objetivo é extremamente sensível a variações desses parâmetros, o que pode provocar soluções instáveis.

Essas limitações têm levado a pesquisadores investirem esforços para o aprimoramento dos métodos numéricos (métodos diretos) para solução de DAOPs. Tais métodos serão discutidos nos próximos tópicos.

3.2.3. Método Sequencial

No método *single-shooting* (ou sequencial) a variável de controle é discretizada e configura o conjunto das variáveis de decisão do problema de otimização dinâmica, enquanto as equações de estado se mantêm na sua estrutura de DAE original, sendo solucionadas por métodos de integração específicos. A variável de controle pode ser parametrizada por funções locais como expansão polinomial de Lagrange, *splines*, linear, etc. Uma maneira conveniente de parametrizar a variável de controle é subdividir o horizonte de tempo em $n_s \geq 1$ estágios (elementos):

$$\boldsymbol{\tau} = \left\{ \tau_1, \dots, \tau_{n_s} \right\} \quad (3.45)$$

em que

$\tau_i|_{i=1,\dots,n_s}$ corresponde a cada estágio, sendo que $\tau_o = t_o$.

Normalmente utiliza-se a parametrização⁷ constante por partes:

$$u(t) = U(\tau, \mathbf{v}) = u(t) = v_i \text{ se } \tau_{i-1} < t \leq \tau_i, \quad i = 1, \dots, n_s \quad (3.46)$$

em que:

$\mathbf{v} = \{v_1, \dots, v_{n_s}\}$ é o vetor de coeficientes de controle (variáveis de decisão).

Logo, o DAOP (formulação de Meyer) pode ser descrito como:

$\begin{aligned} \min_{\mathbf{v}, t_f, \mathbf{p}} \mathfrak{J} &= \min_{\mathbf{v}, t_f, \mathbf{p}} \left[\varphi(t_f, \mathbf{x}(t_f), \mathbf{p}) \right] \\ \text{sujeito a } \mathbf{h}(t, \dot{\mathbf{x}}(t), \mathbf{x}(t), \mathbf{y}(t), \mathbf{u}, \mathbf{p}) &= 0 \\ \mathbf{g}(t, \mathbf{x}(t), \mathbf{y}(t), \mathbf{u}, \mathbf{p}) &\leq 0 \\ \mathbf{e}(t_f, \mathbf{x}(t_f)) & \\ \mathbf{u}^{\min} &\leq \mathbf{u} \leq \mathbf{u}^{\max} \\ \mathbf{x}_0 &= \mathbf{x}(t_0), \quad t_0 < t < t_f \end{aligned}$	<p>(3.47)</p> <p>(3.48)</p>
---	-----------------------------

Com a parametrização da variável de controle, o problema de otimização dinâmica é solucionado numericamente, utilizando-se um algoritmo de programação não linear, NLP (PSHENICHNY e DANILIN, 1978). A cada iteração do NLP as restrições do sistema devem ser computadas, o que exige a utilização de rotinas de integração (por ex.: Runge-Kutta ou BDF). Métodos de otimização como SQP e Ponto Interior (NOCEDAL e WRIGHT, 1999) podem ser aplicados para solução do sistema.

O cômputo do gradiente da função objetivo e restrições normalmente é realizado por técnicas como (i) análise de sensibilidade, (ii) equações adjuntas ou (iii) diferenças finitas.

A estrutura básica do método sequencial é esquematizada na Figura 3.2.

Para algoritmos de otimização baseados no cômputo do Jacobiana é necessário calcular a sensibilidade do funcional φ em relação às variáveis de decisão, ou seja:

$$\nabla_{\mathbf{v}_j} \varphi = \frac{\partial \varphi}{\partial x_1} \frac{\partial x_1}{\partial v_j} + \frac{\partial \varphi}{\partial x_2} \frac{\partial x_2}{\partial v_j} + \dots + \frac{\partial \varphi}{\partial x_{n_x}} \frac{\partial x_{n_x}}{\partial v_j} + \frac{\partial \varphi}{\partial v_j} = \frac{\partial \varphi}{\partial \mathbf{x}} \cdot \frac{\partial \mathbf{x}}{\partial v_j} + \frac{\partial \varphi}{\partial v_j} \quad (3.49)$$

⁷ Considera-se, para fins de simplificação, apenas uma variável de controle.

Note que $\frac{\partial \mathbf{x}}{\partial v_j}$, $j = 1, \dots, n_s$ depende da dinâmica do processo, logo:

$$\nabla_{v_j} h_i = \frac{\partial h_i}{\partial \dot{x}_1} \frac{\partial \dot{x}_1}{\partial v_j} + \frac{\partial h_i}{\partial x_1} \frac{\partial x_1}{\partial v_j} + \dots + \frac{\partial h_i}{\partial \dot{x}_{n_x}} \frac{\partial \dot{x}_{n_x}}{\partial v_j} + \frac{\partial h_i}{\partial x_{n_x}} \frac{\partial x_{n_x}}{\partial v_j} + \frac{\partial h_i}{\partial v_j} \quad (3.50)$$

obtendo-se a variação as equações de sensibilidade para cada função h_i , $i = 1, \dots, n_h$. Note que se o DAOP obtiver parâmetros \mathbf{p} como variáveis de decisão, os mesmos também são incorporados ao sistema.

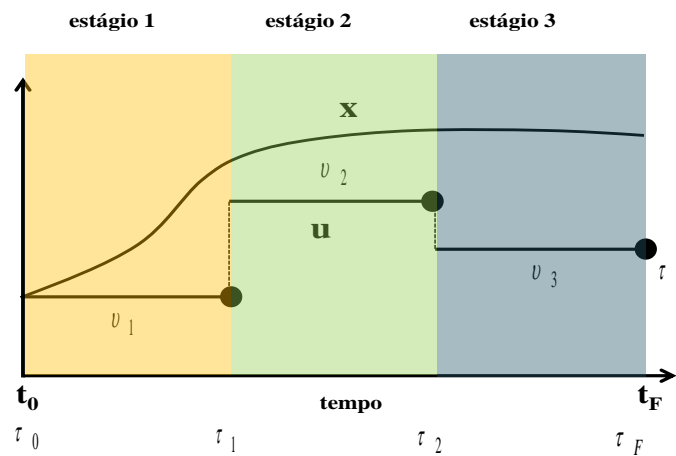


Figura 3.2. Ilustração do método sequencial (considerando uma variável de controle e uma variável de estado).

Uma das vantagens do cômputo das equações de sensibilidade é a possibilidade de integração das mesmas juntamente com o sistema de DAEs. Além disso, se utilizado algoritmos de integração que utilizam informações da Jacobiana, essa estrutura pode ser vantajosa, pois já pré-calcula os coeficientes de sensibilidade (BIEGLER, 2010).

O cômputo da Jacobiana é uma etapa crucial para o desempenho dos algoritmos de otimização, principalmente em estratégias sequenciais em que o cálculo do gradiente e Hessiana da função objetivo é elevado e corresponde a cerca de 80% do custo computacional (BIEGLER, 2010). Em decorrência disso, métodos numéricos, como diferenças finitas, tornam-se impraticáveis para sistemas de elevada dimensão. BALSACANTO *et al.* (2001) também implementaram o método sequencial provido de informações na Jacobiana e hessiana analíticas. Os pesquisadores utilizaram um procedimento de refinamento sucessivo do perfil de controle a fim de adaptar a discretização à mudança dos arcos de controle, sendo possível obter malhas com níveis de discretização suficientes a ponto de detectar de maneira eficaz os pontos de junção.

As aplicações dos métodos *single shooting* remontam da década de 70. POLLARD e SARGENT (1970) aplicaram a abordagem sequencial na busca pelas condições ótimas para operação de uma coluna de destilação. Estas técnicas utilizavam a discretização dos perfis de controle com parametrização constante por partes para integrar os modelos descritos por EDOs e solucionar o problema de otimização sequencialmente.

Na década de 1990, tais métodos passaram a usufruir de grande interesse acadêmico devido ao maciço investimento em computação e consequente aproveitamento de novas ferramentas para solução de problemas de engenharia. Diante desse cenário, os métodos de integração de DAEs passaram a se tornar cada vez mais eficazes para solucionar tais problemas, antes muito dispendiosos, o que permitiu viabilizar ainda mais a utilização dos métodos sequenciais.

VAASSILIADIS *et al.* (1994a, 1994b) testaram a discretização parcial (das variáveis de controle) com polinômios de Lagrange. BETTS e HUFFMAN (1999) investiram em melhorias do algoritmo sequencial ao utilizarem um algoritmo baseado em sequência de parametrizações sucessivas a fim de melhorar o esquema de discretização e aprimorar a estimativa inicial da trajetória de controle.

LEE *et al.* (1999) propôs utilizar algoritmos evolutivos para melhorar a estimativa inicial do perfil de controle. Outras tentativas de melhorar a estimativa inicial foram testadas mais recentemente por WEISER (2001), BINDER (2002), BINDER *et al.* (2000a; 2000b), SCHLEGEL (2004), SCHLEGEL *et al.* (2005) e JAIN e TSOTRAS (2007). Esses autores utilizaram estratégias de adaptação de malhas para parametrizar a variável de controle.

A literatura vem mostrando que o método sequencial apresenta vantagens frente aos métodos simultâneos para solucionar problemas de dimensão elevada, em especial por manipular pouco o sistema de EADs e a possibilidade de utilizar softwares de integração robustos, como a rotina DASSL (BRENAN, 1996; PETZOLD, 1989), capazes de lidar com problemas de rigidez e índice diferenciais.

Uma limitação do método sequencial é que, dependendo da natureza do problema, há necessidade de realizar a integração do modelo exaustivamente, o que pode acarretar em um custo computacional elevado. Além disso, a aplicação desses métodos depende da

estabilidade do estado estacionário quando aplicado em uma unidade industrial (BIEGLER, 2010). Outra limitação é o tratamento das restrições, que também dependem do nível de discretização das variáveis de controle assim como do nível de amostragem das variáveis de estado (BIEGLER, 2007).

3.2.4. Método Simultâneo

Uma alternativa menos usual para solução de sistemas de EDOs (ou DAEs de índice diferencial 1) é a aproximação polinomial das variáveis. A aproximação por um polinômio \wp de ordem q deve satisfazer as equações diferenciais nos pontos de colocação selecionados dentro de um intervalo $t_I = t \in [t_i, t_{i+1}]$, ou seja:

$$x_I = \wp(t_I, \mathbf{r}) = \sum_{j=0}^q r_j t_I^j = r_0 t_I^0 + r_1 t_I^1 + \dots + r_q t_I^q \quad (3.51)$$

Note que no intervalo $[t_i, t_{i+1}]$ existem $q+1$ pontos de colocação, ou seja, $t_I^{(k)} = t_I + c_k \Delta t$, $k = 0, \dots, q$, onde $0 \leq c_1 < c_2 < \dots < c_q \leq 1$. Os $q+1$ coeficientes $\mathbf{r} = r_0, \dots, r_q$ são determinados pelas $q+1$ equações que impõem que a derivada de x_i em cada ponto de colocação corresponda à função f , ou seja:

$$\begin{aligned} f\left(\wp\left(t_I^{(1)}, \mathbf{r}\right), t_I^{(1)}\right) &= \dot{\wp}\left(t_I^{(1)}, \mathbf{r}\right) = \sum_{j=0}^q j r_j \left(t_I^{(1)}\right)^{j-1} \\ &\vdots \\ f\left(\wp\left(t_I^{(q)}, \mathbf{r}\right), t_I^{(q)}\right) &= \dot{\wp}\left(t_I^{(q)}, \mathbf{r}\right) = \sum_{j=0}^q j r_j \left(t_I^{(q)}\right)^{j-1} \end{aligned} \quad (3.52)$$

Note que as Equações (3.51) e (3.52) formam um sistema de $q+1$ equações. Uma importante observação é a escolha dos pontos de colocação. A quantidade e localização dos pontos que determinam os diferentes métodos. Por exemplo, para $q=1$ e $c_1=0$ a aproximação equivale ao método de Euler implícito.

Para a escolha da configuração da discretização, geralmente utilizam-se métodos de quadratura numérica. Por exemplo, como a trajetória exata de x é computada como:

$$x(t_{i+1}) = x(t_i) + \int_{t_i}^{t_{i+1}} f(x(t), t) dt \quad (3.53)$$

A aproximação é dada por:

$$\wp(t_{I+1}, \mathbf{r}) = \wp(t_I, \mathbf{r}) + \Delta t \cdot \sum_{j=1}^q \omega_j f\left(P(t_I^{(j)}, \mathbf{r}), t_I^{(j)}\right) \quad (3.54)$$

em que ω_j corresponde aos pesos de quadratura numérica. Utilizando, por exemplo, quadratura de Gauss para essa integral, escolhendo os pontos de colocação em um intervalo $[t_i, t_{i+1}]$, a integral é exata para polinômios de ordem até $2q-1$. Tal estratégia é definida como colocação Gauss-Legendre. Outra metodologia, conhecida como colocação Gauss-Radau é mais indicada para problemas com sistemas de DAEs rígidas (DIEHL2011).

Os métodos de colocação são extensivamente utilizados como estratégia de discretização em métodos para solução de DAOPs conhecidos como métodos simultâneos.

No método simultâneo o sistema de DAEs é discretizado gerando um sistema algébrico, resultando em:

$$\min_{\mathbf{v}, \mathbf{r}, \mathbf{p}, t_f} J = \min_{\mathbf{v}, \mathbf{r}, \mathbf{p}, t_f} \left[\varphi(t_f, \mathbf{x}_{n_s}, \mathbf{p}) \right] \quad (3.55)$$

$$\begin{aligned} \text{sujeito a } \chi(\mathbf{v}_I, \mathbf{r}_I, \mathbf{p}) &= 0, & I = 0, \dots, N-1 \\ \wp_I(t_{I+1}, \mathbf{r}_I, \mathbf{p}) - x_{I+1} &= 0, & I = 0, \dots, N-1 \\ g(\mathbf{v}_I, \mathbf{r}_I, \mathbf{p}) &\leq 0, & I = 0, \dots, N-1 \\ e(t_f, \mathbf{r}_I) &= 0, \\ x_{I=0} &= x(t_0) \end{aligned} \quad (3.56)$$

em que N é o número de elementos e discretização.

No sistema acima as equações $\chi(\mathbf{v}_I, \mathbf{r}_I, \mathbf{p}) = 0$ são representadas pelas equações que impõem as condições de colocação, ou seja:

$$x_I = \wp(t_I, \mathbf{r}_I) = \sum_{j=0}^q r_{I,j} t_I^j \quad (3.51)$$

$$u_I = U(t_I, \mathbf{v}_I) = \sum_{j=0}^q v_{I,j} t_I^j \quad (3.57)$$

$$\left. \begin{array}{l}
 f\left(\phi\left(t_I^{(1)}, \mathbf{r}_I\right), U\left(t_I, \mathbf{v}_I\right), \mathbf{p}\right) = \dot{\phi}\left(t_I^{(1)}, \mathbf{r}_I\right) = \sum_{j=0}^q j r_{I,j} \left(t_I^{(1)}\right)^{j-1} \\
 \vdots \\
 f\left(\phi\left(t_I^{(q)}, \mathbf{r}_I\right), U\left(t_I, \mathbf{v}_I\right), \mathbf{p}\right) = \dot{\phi}\left(t_I^{(q)}, \mathbf{r}_I\right) = \sum_{j=0}^q j r_{I,j} \left(t_I^{(q)}\right)^{j-1}
 \end{array} \right\} \begin{array}{l}
 q \text{ equações} \\
 q \text{ variáveis } \left(r_{I,j}\right)
 \end{array} \quad (3.52)$$

No sistema acima a continuidade das variáveis de estado é obtida automaticamente entre os elementos, pelo fato de serem funções implícitas das variáveis contínuas diferenciáveis. Já para as variáveis de controle, descontinuidades entre os elementos são possíveis, o que é vantajoso para a captura de soluções com arcos e pontos de junção, conforme ilustra a Figura 3.3.

O método simultâneo se consolidou como uma estratégia eficaz para solução de DAOPs, após uma extensa investigação sobre aplicações em processos químicos, que se iniciou na década de 1980 a partir de trabalhos de BIEGLER (1984). Na década de 1990 outros pesquisadores aplicaram esses métodos para solução de problemas de engenharia química: BETTS e HUFFMAN (1992; 1993; 1999); CERVANTES e BIEGLER (2000). Para BIEGLER *et al.* (2002) e BIEGLER (2010) uma das vantagens de se utilizar a discretização total é a possibilidade de solucionar o sistema de apenas uma vez, tornando em geral a solução mais veloz, ou seja, as restrições impostas pelo sistema de DAEs são satisfeitas somente no final do cálculo de otimização, conduzindo o algoritmo a percorrer um caminho inviável, ao contrário do método sequencial, que necessita satisfazer as restrições a cada iteração do NLP. Em outras palavras, a otimização e integração do sistema são realizadas simultaneamente.

De acordo com DIEHL *et al.* (2006), as principais vantagens dos métodos simultâneos são: (i) geração de sistemas NLP esparsos; (ii) estimativas da variável de estado podem ser utilizadas na inicialização; (iii) a estrutura de discretização favorece solucionar DAEs com características instáveis e (iv) fácil tratamento de restrições de trajetória de igualdade. Tais características são defendidas por LANG e BIEGLER (2007) que enaltecem a superioridade dos métodos simultâneos sobre os métodos sequenciais.

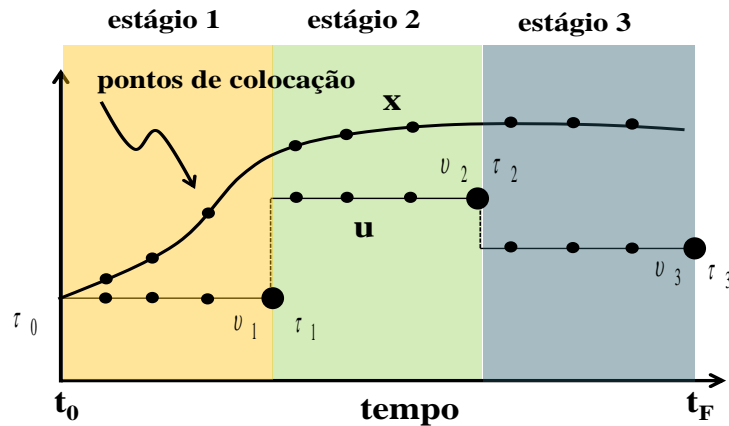


Figura 3.3. Estrutura do método simultâneo (considerando uma variável de controle e uma variável de estado).

De fato, historicamente o método simultâneo mostra-se bastante superior ao método sequencial em se tratando de desempenho computacional. Por exemplo, BIEGLER *et al.* (2002) aplicou o método simultâneo acoplado a um método de ponto interior que possibilitou solucionar um sistema de DAEs com mais de 800 mil variáveis em cerca de 67 minutos. É muito raro encontrar estudos envolvendo o método sequencial que tratam problemas dessa ordem de complexidade, com exceção do trabalho de HARTWICH e MARQUARDT (2010) que aplicaram o método sequencial para otimizar um processo representado por um modelo de DAEs com mais de 12000 variáveis.

Geralmente, no método simultâneo, a dimensão (coeficientes das variáveis de estado e controle) do sistema discretizado é bastante elevada (podendo exceder um milhão de variáveis), se comparado ao método sequencial. Por outro lado, a estrutura da matriz Hessiana é esparsa, o que facilita a implementação de rotinas que exploram a esparsividade do sistema, aumentando o desempenho computacional. Com o desenvolvimento de métodos como SQP reduzido (rQSP) (NOCEDAL e WRITE, 1999) e algoritmos de ponto interior (principalmente para problemas com um número elevado de restrições) tais métodos têm se tornado mais atrativos (LANG e BIEGLER, 2007). CERVANTES *et al.* (2000) aplicaram um método de ponto interior (IPOPT) com função barreira logarítmica para tratamento do elevado número de restrições do sistema NLP evidenciando elevado desempenho do IPOPT em problemas dessa natureza.

Uma das maiores desvantagens claramente é o desconhecimento prévio do número de elementos N necessários para obter uma solução factível. Segundo

BIEGLER (2010), essa avaliação é geralmente realizada na tentativa e erro, pois não há conhecimento prévio da solução. Em particular, a localização dos pontos de ativação e desativação das restrições também dificulta o desempenho do método. Uma alternativa é a inclusão do número de elementos como variáveis de decisão do problema, gerando um procedimento de elementos finitos móveis (LEMOS, 2007), conforme realizado por BIEGLER *et al.* (2002). Mais recentemente ALMEIDA Nt. e SECCHI (2011) avaliaram diversos aspectos do método simultâneo ao otimizarem uma planta de craqueamento catalítico no *software* DynoPC (LANG e BIEGLER, 2007). Os autores destacaram que a qualidade da discretização é mais sensível ao número de pontos de colocação que ao número de elementos. Adicionalmente, destacaram a preocupação com o custo computacional sempre dependente do nível de discretização.

CUTHRELL e BIEGLER (1989) enfatizam que as condições KKT do sistema NLP, sem restrições, resultante são consistentes com as condições de otimalidade deduzidas a partir do PMP. No entanto, em problemas com restrições de trajetória e com arcos singulares essa verificação não é uma tarefa trivial. Por exemplo, BIEGLER (2010) mostra que em problemas de arco singular, a Hessiana do sistema tende a se tornar mal condicionada com o aumento do nível de discretização.

Segundo BIEGLER *et al.* (2002) e BIEGLER (2010) a discretização de DAEs por métodos de colocação limitam a possibilidade de ocorrência de modos dinâmicos instáveis. Essas condições podem ser verificadas pela função de Green (HOOG e MATTHEIJ, 1987) associada ao problema de valor de contorno equivalente à estrutura resultante da discretização total no método simultâneo. Tal controle é mais difícil de ser obtido no método sequencial.

3.2.5. Método *Multiple-Shooting*

O método *multiple-shooting* é considerado como um método híbrido entre o método sequencial e simultâneo, em que o domínio temporal é dividido em elementos e o modelo do sistema de EADs é integrado separadamente em cada elemento. A variável de controle é discretizada e parametrizada, como no método sequencial, e as informações do gradiente da função objetivo e variáveis de estado são obtidas em cada elemento. Os elementos são conectados por novas restrições de igualdade do problema de otimização, que dependerão do número de elementos (BIEGLER, 2007). Em termos estruturais, a

maior diferença do método sequencial é que as condições iniciais em cada elemento são incluídas como variáveis de decisão do problema. Para garantir a continuidade das variáveis de estado as restrições de igualdade garantem que o valor obtido no final de um elemento deve ser igual ao valor inicial do próximo elemento. As restrições de desigualdade são computadas diretamente nos pontos de discretização.

A estrutura do método *multiple-shooting* é escrito como (para uma variável de controle):

$$\min_{\mathbf{v}, \mathbf{s}, \mathbf{p}, t_f} J = \min_{\mathbf{v}, \mathbf{s}, \mathbf{p}, t_f} \left[\varphi(t_f, \mathbf{x}(t_f), \mathbf{p}) \right] \quad (3.58)$$

$$\begin{aligned} \text{sujeito a } & h(t, \dot{x}(t), x(t), y(t), U(t, \mathbf{v}), p) = 0 \\ & g(t, x(t), y(t), U(t, \mathbf{v}), p) \leq 0 \\ & e(t_f, x(t_f)) \\ & u^{\min} \leq U(t, \mathbf{v}) \leq u^{\max} \\ & s_i = x_i(\tau_i), \quad i = 2, \dots, n_s \\ & s_{i+1} = x_i(\tau_{i+1}), \quad i = 2, \dots, n_s \\ & x_0 = s_0 \end{aligned} \quad (3.59)$$

No sistema o parâmetro s corresponde às condições iniciais em cada estágio (elemento), ilustrado na Figura 3.4.

À primeira vista, o método *multiple-shooting* oferece poucas vantagens frente ao método *single-shooting*. Novas variáveis de decisão são incorporadas ao problema, requisitando maior esforço computacional para cômputo da Jacobiana. No entanto, para sistemas instáveis as vantagens se tornam claras. As mesmas condições observadas por CERVANTES *et al.* (2002) nos métodos de colocação também são válidas para métodos *multiple-shooting*. Ademais, como a integração é realizada de maneira independente em cada estágio, essa tarefa pode ser implementada em um algoritmo paralelo (LEINEWEBER *et al.*, 2003; HARTWICH *et al.*, 2011; DRAG e STYCZEN, 2012). Adicionalmente, segundo DIEHL *et al.* (2006), o sistema NLP resultante da discretização *multiple-shooting* é menor que o correspondente gerado pelo método de colocação, porém, com menor grau de esparsividade.

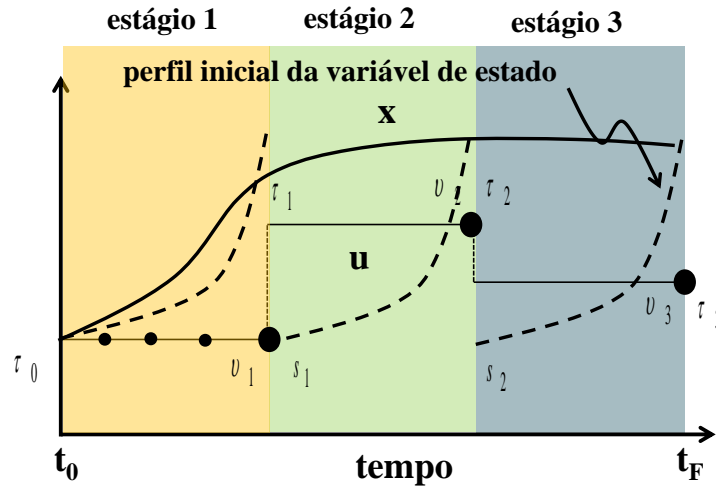


Figura 3.4. Estrutura do método *multiple-shooting* (considerando uma variável de controle e uma variável de estado).

No algoritmo proposto por HARTWICH *et al.* (2011) foi possível reduzir em até quatro vezes o custo computacional em modelos com mais de 13500 DAEs e 75 parâmetros. A paralelização foi implementada na biblioteca Intel MPI em oito núcleos de processamento.

Recentemente, LOGIST *et al.* (2012) implementou DAOPs multi-objetivos no software ACADO. Problemas multi-objetivos resultam em múltiplas soluções (conjunto de Pareto) ao invés de uma solução única. Nos três casos analisados os autores avaliaram diferentes níveis de discretização do perfil de controle, concluindo que intervalos de 20 a 50 estágios são aceitáveis. Para aprimoramento do algoritmo, foi sugerido acoplamento de técnicas de adaptação de malhas (como proposta por SCHLEGEL *et al.*, 2005).

HARTWICH *et al.* (2010) propôs uma estratégia híbrida entre o *multiple shooting* e sequencial. O algoritmo posposto utiliza o sequencial na fase inicial e monitora a estabilidade das variáveis de estado. Na ocorrência de possíveis instabilidades, a estratégia *multiple shooting* é utilizada nos respectivos estágios.

3.3 Considerações Adicionais sobre DAOPs

Conforme introduzido, os métodos diretos para solução de DAOPs são baseados na discretização das variáveis de estado e controle. A qualidade da solução depende da resolução da discretização, especialmente das variáveis de controle. Como o perfil de controle não é conhecido *a priori*, é difícil preestabelecer um perfil de discretização

adequado. Problemas numéricos adicionais podem surgir quando restrições de trajetória são incorporadas ao problema ou quando arcos singulares também estão presentes (BIEGLER, 2010).

Geralmente, DAOPs são constituídos por restrições de desigualdade e igualdade as quais podem ser configuradas em alguns instantes específicos ou somente no instante final ou inicial do problema. Os problemas de restrição de trajetória são notavelmente mais complexos de serem solucionados, pois a dimensão do sistema pode se tornar consideravelmente elevada, dependendo do nível de discretização/amostragem das variáveis de estado, o que pode sobrecarregar o algoritmo de otimização (BELL e SARGENT, 2000). Diversos trabalhos sobre tratamento de restrições de trajetória são abordados na literatura.

Na metodologia utilizada por VASSILIADIS *et al.* (1994b) as restrições de trajetória de desigualdade foram tratadas através da aplicação combinada da discretização dessas restrições em um número finito de pontos, forçando-se que a integral do erro de violação das mesmas tendesse a zero.

Já na estratégia de FEEHERY e BARTON (1996) restrições de trajetória são detectadas durante a etapa de integração. Semelhantemente a VASSILIADIS *et al.* (1994b) as restrições são incorporadas ao sistema de DAEs permitindo assim reduzir o número de avaliações da função objetivo.

Em PFEIFER (2007) o sistema de DAEs foi associado com a técnica de parametrização dos instantes em que ocorrem a ativação e a desativação das restrições ao longo da trajetória desenvolvida originalmente para problemas chaveados, com o objetivo de estimá-los utilizando o método indireto de solução. Foram utilizadas funções identificadoras de fase que expressam a variável de controle analiticamente em cada fase, baseadas na condição de otimalidade e na análise física do problema para distinguir sequências apropriadas de ativação/desativação das restrições, que causam flutuações no índice diferencial.

Em SOUZA (2007) o esforço computacional requerido para outros algoritmos, que pode ser atribuído às estratégias de busca e localização das discontinuidades e aos procedimentos de reinicialização da integração, motivou a concepção de uma nova

abordagem para manipulação de descontinuidades, que podem surgir da violação de restrições de desigualdades originárias: da própria formulação do modelo matemático e, de problemas de otimização, onde tais restrições são inerentes. A essência dessa abordagem é a caracterização das inequações através de funções de regularização (VIEIRA 2001), permitindo comutação automática entre equações. A técnica se mostrou eficiente para manipulação de pontos de junção no perfil de controle.

ALMEIDA e SECCHI (2012) destacam que, em se tratando de otimização dinâmica em tempo real (DRTO), há situações em que a solução do DAOP pode ser inviável devido a conflitos de especificação das restrições, condições iniciais ou mesmo mudanças das restrições durante o cálculo. Os pesquisadores exemplificaram um caso em que as variáveis de estado satisfazem às restrições, porém, a derivada das variáveis pode violar as restrições, sem possibilidade de ações de controle. Motivado por essas possíveis limitações, a estratégia adotada foi reestruturar o sistema DAEs incorporando as restrições de trajetória ao sistema. No entanto, consideraram-se as variáveis de folga resultantes como variáveis de controle, a fim de garantir as relaxações apenas nos intervalos de violação das restrições.

Há diversos códigos e softwares estruturados para solucionar DAOPs. As principais ferramentas, listadas na Tabela 3.1, utilizam, em sua maioria, métodos de colocação, *single-shooting* e *multiple-shooting*.

Tabela 3.1. Principais *softwares* de otimização dinâmica

	Nome	Métodos e Algoritmos	Características
1	ICLOCS (<i>Imperial College London Optimal Control User Guide</i>) * (FALUGI <i>et al.</i> , 2010). site: http://www.ee.ic.ac.uk/ICLOCS/	Métodos <i>Multiple Shooting</i> e Simultâneo; Acoplamento aos pacotes Sundials e IPOPT.	Gratuito; Código em MATLAB; Cômputo da Jacobiana analítica (fornecida pelo usuário).
2	DYNOPT (<i>Dynamic Optimization of Processes</i>) * (CIZNIAR <i>et al.</i> , 2005) . site: http://www.kirp.chtf.stuba.sk/~fikar/research/dynopt/dynopt.htm	Método Simultâneo: colocação em elementos finitos;	Gratuito; Código em MATLAB; Cômputo da Jacobiana analítica (fornecida pelo usuário).

3	DOTcvpSB (<i>MATLAB Toolbox for Dynamic Optimization in Systems Biology</i>)* (HIRMAJER <i>et al.</i> , 2009). site: http://www.iim.csic.es/~dotcvpsb/	Método <i>Single Shooting</i> ; Acoplamento aos pacotes Sundials e IPOPT. Algoritmos de Otimização Estocásticos e Híbridos: ACOMI, MITS.	Gratuito; Código em MATLAB; Otimização em sucessivas malhas; Cômputo automático das equações de sensibilidade.
4	MISER3 (JENNINGS e TEO, 1990) site: http://school.maths.uwa.edu.au/~les/miser/	Método <i>Single Shooting</i> ; Acoplamento ao pacote DASSL;	Gratuito; Código em FORTRAN e MATLAB;
5	RIOTS (<i>Recursive Integration Optimal Trajectory Solver</i>) (SCHWARTZ <i>et al.</i> , 1997). site: http://mechatronics.ece.usu.edu/riots/	Método <i>Single Shooting</i> ; Acoplamento ao pacote NPSOL;	Gratuito; Código em MATLAB e C; Limitações para problemas com restrições de trajetória; Cômputo da Jacobiana analítica (fornecida pelo usuário).
6	ACADO (<i>Automatic Control and Dynamic Optimization</i>) (LOGIST <i>et al.</i> , 2012)	Método <i>Multiple Shooting</i> ; Método <i>Single Shooting</i> ;	Gratuito; Código em MATLAB e C++; Problemas Multi-objetivos; MPC; Estimação de parâmetros.
7	DIDO (ROSS, 1998) site: http://www.elissarglobal.com/industry/products/software-3/	Método Simultâneo: colocação em elementos finitos; Método indireto;	Licenciado (ELISSAR GLOBAL); Código em MATLAB;
8	TOMLAB PROPT* (RUTQUIST e EDVALL, 2010) site: http://tomdyn.com/	Método Simultâneo: colocação em elementos finitos; Acoplamento aos pacotes: KNITRO, CONOPT, SNOPT, CPLEX, etc.	Licenciado; Código em MATLAB; Estimação de parâmetros. Computação simbólica.
9	DIRCOL (<i>Direct Collocation Method for the Numerical Solution of Optimal Control Problems</i>) (STRYC, 1999). site: http://www.sim.informatik.tu-darmstadt.de/1/res/sw/dircol/	Método Simultâneo: colocação em elementos finitos; Método <i>Multiple Shooting</i> ; Acoplamento aos pacotes NPSOL, SNOPT;	Licenciado; Código FORTRAN; Ambiente Linux e Windows.
10	MINOPT (<i>A Modeling Language and Algorithm Framework for Linear, Mixed Integer, Nonlinear, Dynamic</i>)	Integração com diversos pacotes: CPLEX, LPSOLVE, MINOS, NPSOL, SNOPT, DASOLV, etc.	Licenciado; Cômputo automático das equações de sensibilidade.

	<p><i>and Mixed Integer Non Linear Optimization</i>) (SCHEWIGER e FLOUDAS, 1998)</p> <p>site: http://titan.princeton.edu/MINOPT/minopt.html</p>		
11	<p>SOCS (<i>Sparse Optimal Control Software Family</i>)</p> <p>site: http://www.boeing.com/boeing/phantom/socs/</p>	<p>Método Simultâneo: colocação em elementos finitos; Método Indireto;</p>	<p>Licenciado (BOEING); Interface com C e MATLAB; Explora técnicas de álgebra esparsa para problemas de elevada dimensão; Refinamento automático de malhas;</p>
12	<p>TACO (<i>Toolkit for AMPL Control Optimization</i>) (KIRCHES e LEYFER, 2011)</p> <p>site: http://www.mcs.anl.gov/publication/taco-toolkit-ampl-control-optimization</p>	<p>Método <i>Multiple Shooting</i>; Método Simultâneo: colocação em elementos finitos;</p>	<p>Licenciado; Interface com AMPL e MUSCOD-II;</p>
13	<p>OCMAT (Optimal Control Problems in Matlab) (GRASS <i>et al.</i>, 2008)</p> <p>site: http://orcos.tuwien.ac.at/research/ocmat_software/</p>	<p>Método Indireto;</p>	<p>Gratuito; Código em MATLAB;</p>
14	<p>gPROMS[®] (PSE)</p> <p>site: http://www.psenterprise.com/modelbuilder.html</p>	<p>Método <i>Multiple Shooting</i>; Método <i>Single Shooting</i>;</p>	<p>Licenciado; Ambiente de modelagem e simulação com rica interface gráfica; Interface com MATLAB;</p>
15	<p>MUSCOD-II (<i>Multiple Shooting Code for Optimal Control</i>) (HOFFMAN <i>et al.</i>, 2011).</p> <p>site: http://www.iwr.uni-heidelberg.de/groups/agbock/FILES/muscod_manual.pdf</p>	<p>Método <i>Multiple Shooting</i>; Método Simultâneo: colocação em elementos finitos; Diversos pacotes: LIBLAC, DAESOL, BLAS, LAPACK, QPQPT, AMPL;</p>	<p>Licenciado; Escrito em C, FORTRAN com interface em MATLAB e NEOS; Ambiente Linux e Windows.</p>
16	<p>DYNOPC(LANG e BIEGLER, 2007)</p>	<p>Método Simultâneo: colocação em elementos finitos; Elementos finitos móveis.</p>	<p>Licenciado.</p>

Além das ferramentas listadas há outros ambientes de simulação que permitem o usuário configurar DAOPs, como o ASPEN[®] e MATLAB[®].

LOGSDON e BIEGLER (1993) desenvolveram uma estratégia para melhorar a eficiência do SQP, decompondo o modelo algébrico discretizado e aplicando uma relaxação ao sistema. Outra estratégia para códigos SQP foi apresentada por CERVANTES *et al.* (2000), a qual aproveita a estrutura esparsa do sistema algébrico resultante. Os autores denominaram o novo código de rSQP (*Reduced Successive Quadratic Programming*).

Dependendo do problema, a estimativa inicial pode ser um fator crucial para a solução numérica do problema de otimização. Alguns pesquisadores têm aplicado métodos estocásticos para otimização dinâmica. Uma das vantagens desses métodos é a independência de estimativas iniciais. YANG e YAO (2013) exemplificam diversas aplicações de algoritmos genéticos e ant colony para DAOPs. BANGA *et al.* (1998) aplicaram um algoritmo estocástico intitulado *Integrated Controlled Random Search for Dynamic Systems* (ICRS/DS) para controle ótimo de sistemas de engenharia química.

UPRETI (2004) utilizou algoritmos genéticos para solução de problemas de controle ótimo. Segundo o autor, a aplicação de algoritmos estocásticos possibilita implementar a otimização de maneira robusta, independente de estimativas iniciais ou soluções auxiliares.

Apesar de nos métodos estocásticos não haver necessidade de cômputo das derivadas da função objetivo e restrições, há necessidade de um número elevado de avaliações da função objetivo que pode resultar em elevado custo computacional, especialmente em sistemas de elevada dimensão (BALSA-CANTO *et al.*, 2001). No entanto, alguns pesquisadores vêm aplicando algoritmos globais determinísticos para solucionar DAOPs.

ESPOSITO e FLOUDAS (2000) aplicaram o método de otimização global α -BB (ADJIMAN *et al.* 1994) para solução de problemas NLP resultantes de discretização de sistemas de DAEs. Soluções locais obtidas via método sequencial foram utilizadas como limite superior do mínimo global da função objetivo. Limites inferiores foram computados através da solução de relaxações convexas do problema em sub-regiões, obtidas por meio da estimação convexa da função objetivo.

Capítulo 4

Aplicações, Avanços Recentes e Novas Propostas

Neste Capítulo

Tópico 4.1. Visão geral sobre Aplicações de DAOPs em Processos Químicos;

Tópico 4.2. Adaptação de Malhas em Otimização Dinâmica: Aplicação de Wavelets;

Tópico 4.3. Wavelets como Aproximação

O objetivo deste capítulo é resumir as principais pesquisas recentes sobre projetos de DAOPs. Procura-se retratar as principais limitações que desencorajam a aplicabilidade industrial. Sobre esse prisma, realiza-se uma revisão criteriosa sobre pesquisas envolvendo adaptação de malhas aplicada a sistemas algébrico-diferenciais. Tais aplicações têm despertado bastante interesse atualmente, principalmente mediante ao crescimento vertiginoso da computação científica e demanda por otimização em tempo real.

4.1. Visão Geral sobre Aplicações de DAOPs em Processos Químicos

A otimização dinâmica tem diversas aplicações para a área de ciências exatas, como astronomia (BETTS, 2010; LONGUSKI *et al.*, 2014), robótica (DIEHL *et al.*, 2006), economia (KAMIEN e SCHWARTZ, 1991) e teoria dos jogos (GEERING, 2007).

Na área de engenharia química, pode-se enfatizar o estudo da otimização dinâmica sobre diversos problemas onde a dinâmica do processo é significativa, tais como: reatores semi-batelada (VASSILIADIS *et al.*, 1994a; 1994b, 1994c; SRINIVASAN *et al.*, 2003a, 2003b; SCHLEGEL *et al.*, 2005; BIEGLER 2010; ASSASSA e MARQUARDT, 2014), processos de destilação (WOZNY e LI 2000; HARTWICH e MARQUARDT, 2010), camada de otimização de sistemas de controle preditivos NMPC (DIEHL *et al.*, 2006; SCHLEGEL, 2004; HUANG, 2010; WÜRTH *et al.*, 2009), transição ótima de processos contínuos (PRATA *et al.*, 2008), estimação de

parâmetros de sistemas dinâmicos (BINDER, 2002), escoamento em poços de petróleo NALUM (2013), etc.

Problemas de otimização dinâmica na engenharia química possuem, em sua maioria, restrições de trajetória (ao longo do tempo) sobre as variáveis de estado, resultantes, por exemplo, de limitações físicas ou de segurança impostas pelo sistema, o que torna o problema ainda mais complexo de ser solucionado (Figura 4.1). Em algumas unidades industriais, o perfil ótimo resultante da solução de tais problemas pode ser utilizado como, por exemplo, setpoints ótimos de controladores ou para a especificação da qualidade de produtos (SOUZA, 2007).

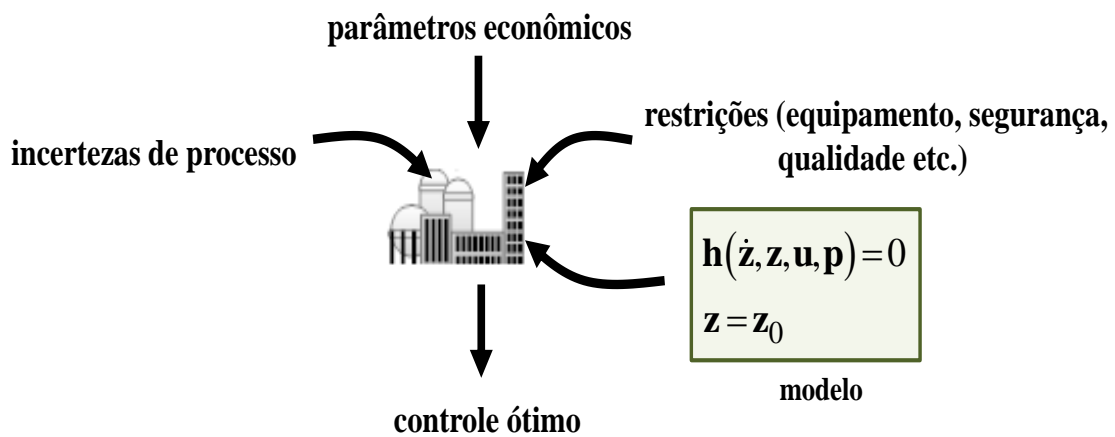


Figura 4.1. Ilustração: Aplicação de DAOP em processos químicos.

A aplicação de otimização dinâmica em tempo real (DRTO) envolve normalmente problemas de estimação de parâmetros, Controle Preditivo Não Linear (NMPC) e reconciliação de dados (DIEHL *et al.*, 2006). Para implementação em tempo real, o desempenho computacional de cálculo é primordial, pois como o objetivo é aplicar a solução ótima em um intervalo (horizonte) de tempo pré-estabelecido, no mínimo o cálculo da otimização deve ser consideravelmente inferior a esse intervalo (DIEHL *et al.*, 2006). Por exemplo, em estratégias MPC (GARCIA *et al.*, 1989), que utilizam modelos linearizados, possibilita-se quantificar as ações de controle em segundos. Já na realidade atual, DAOPs com elevada dimensão (até milhares de variáveis) apresentam um elevado custo computacional. Além disso, ao aplicar a solução ótima, sua validade pode rapidamente ser perdida, pois perturbações inerentes

ao processo podem afetar o sistema, e a solução ótima computada perder a validade. Por exemplo, DIEHL *et al.* (2006) estudou aspectos que afetam a estabilidade de sistemas de otimização dinâmica em malha fechada. Algumas estratégias (HUANG, 2010) sugerem a divisão do DRTO em duas camadas, com a DAOP realizada em malha aberta em uma camada superior (otimização econômica) acoplada a uma camada de MPC inferior para corrigir ações de controle.

Apesar de aplicação de DRTOs ser um procedimento bastante dificultoso, a própria solução de DAOPs, mesmo com modelos reduzidos onde o custo computacional não é o fator de maior preocupação, é um procedimento complexo. Uma das principais razões é o fato da solução de DAOPs consistirem de um ou mais intervalos, denominados arcos, onde os perfis de controle são contínuos e diferenciáveis, porém oscilam de um intervalo para outro entre os pontos de junção. Conseqüentemente, os perfis de controle podem apresentar uma frequência não uniforme entre diferentes arcos.

Observa-se que há uma crescente preocupação em sofisticar os algoritmos para solução de tais problemas, tornando cada vez mais viável a aplicação industrial. Nota-se que os aspectos mais preocupantes dizem respeito principalmente ao tratamento de problemas com arcos singulares, restrição de trajetória, descontinuidade no perfil de controle e convergência numérica.

Pesquisadores (SCHLEGEL, 2004; BIEGLER, 2007; BIEGLER, 2010) entendem que uma política de discretização eficaz ajuda a minimizar algumas dessas limitações. Particularmente, estratégias de adaptação de malhas ou elementos finitos móveis têm sido implementadas. Basicamente, nessas estratégias, os estágios de discretização podem variar ao longo da solução, gerando malhas representadas em diferentes níveis de resolução.

O exemplo seguinte ilustra a aplicação do método sequencial para solucionar um problema de controle ótimo singular. Segundo BIEGLER (2010), o método simultâneo apresenta resultados satisfatórios para problemas com arcos singulares. Porém, dificuldades aparecem quando um perfil de controle requer níveis de discretização mais refinados. Nesse caso, a matriz Hessiana se torna mal condicionada à medida que o nível de discretização aumenta.

Exemplo 4.1. Controle Ótimo Singular (BIEGLER, 2010)

$$\min_u x_3\left(\frac{\pi}{2}\right) \quad (4.1)$$

$$\dot{x}_1 = x_2(t) \quad x_1(0) = 0$$

$$\dot{x}_2 = u(t) \quad x_2(0) = 1$$

$$\dot{x}_3 = x_2(t)^2/2 - x_1(t)^2/2 \quad x_3(0) = 0$$

$$-1 \leq u(t) \leq 1$$

$$t \in [0, \pi/2]$$

Note que, ao aplicar o PMP sobre o problema, é possível obter a expressão para o perfil de controle. Definindo a função Hamiltoniano:

$$\bar{h}(t) = \lambda_1(t)x_2(t) + \lambda_2(t)u(t) + \lambda_3(t)\left(x_2(t)^2/2 - x_1(t)^2/2\right) \quad (4.2)$$

e aplicando as condições de otimalidade:

$$\dot{\lambda}_1 = -\frac{\partial \bar{h}(t)}{\partial x_1(t)} = x_1(t)\lambda_3(t) \quad (4.3)$$

$$\dot{\lambda}_2 = -\frac{\partial \bar{h}(t)}{\partial x_2(t)} = -\lambda_1(t) - \lambda_3(t)x_2(t) \quad (4.4)$$

$$\dot{\lambda}_3(t) = -\frac{\partial \bar{h}(t)}{\partial x_3(t)} = 0, \quad (4.5)$$

$$\frac{\partial \bar{h}(t)}{\partial u(t)} = \lambda_2(t) = 0 \quad (4.6)$$

Note que o Hamiltoniano tem dependência linear em relação à variável de controle. Nesse caso, ao diferenciar em relação ao tempo até explicitar a variável de controle, resulta em:

$$\frac{\partial}{\partial t} \frac{\partial \bar{h}}{\partial u} = \dot{\lambda}_2 = -\lambda_1(t) - \lambda_3(t)x_2(t) \quad (4.7)$$

$$\frac{d^2}{dt^2} \frac{\partial \bar{h}}{\partial u} = \ddot{\lambda}_2 = -\dot{\lambda}_1 - \underbrace{\dot{\lambda}_3 x_2(t)}_0 - \dot{x}_2 \lambda_3(t) = -x_1(t)\lambda_3(t) - u(t)\lambda_3(t) = 0 \quad (4.8)$$

Logo:

$$u(t) = -x_1(t) \quad (4.9)$$

Finalmente, as condições finais são obtidas:

$$\lambda_1(\pi/2) = \frac{\partial \varphi}{\partial x_1} = 0$$

$$(4.10)$$

$$\lambda_2(\pi/2) = \frac{\partial \varphi}{\partial x_2} = 0$$

$$(4.11)$$

$$\lambda_3(\pi/2) = \frac{\partial \varphi}{\partial x_3} = 1$$

$$(4.12)$$

Sabendo-se que $\varphi = x_3(\pi/2)$

A solução numérica obtida com 32 estágios discretização é mostrada nas Figuras 4.2 e 4.3.

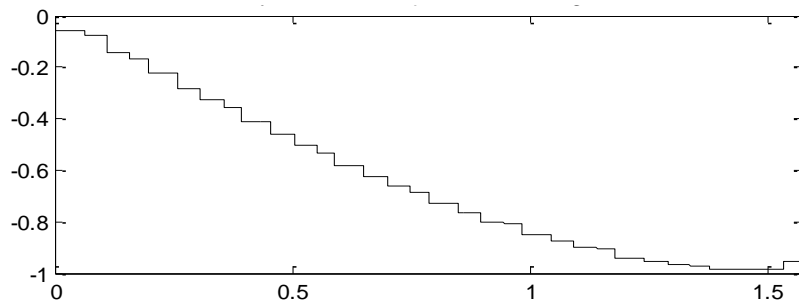


Figura. 4.2. Perfil de controle ótimo obtido com método sequencial com 32 estágios de discretização.

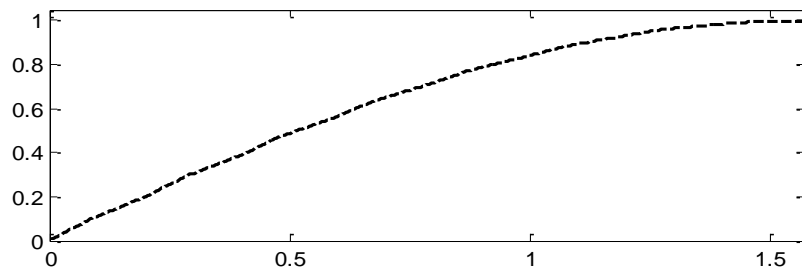


Figura. 4.3. Variável de estado.

A Figura 4.4 ilustra a trajetória de controle obtida pelo método simultâneo, com 32 elementos. BIEGLER (2010) mostra que a solução desse problema via método simultâneo apresenta trajetória de controle consideravelmente oscilatória.

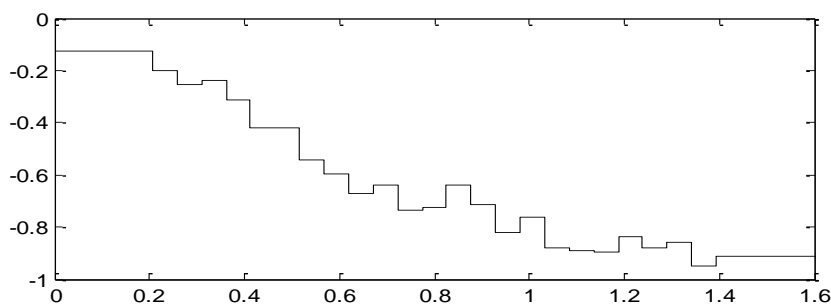


Figura. 4.4. Perfil de controle ótimo obtido com método simultâneo.

O próximo tópico apresenta os principais trabalhos que abordaram estratégias de adaptação de malhas em DAOPs.

4.2. Estratégias de Adaptação de Malhas em Otimização Dinâmica: Aplicação de Wavelets

Estratégias de adaptação de malhas em DAOPs vêm sendo aplicadas desde a década de 1990. Nitidamente, a estratégia de implementação de sucessivas otimizações, com refinamento sucessivo do perfil de discretização das variáveis e atualização da estimativa inicial tem se mostrado uma estratégia bastante promissora, sendo inclusive já aplicável em alguns dos softwares citados no Capítulo 3. Uma aplicação dessa metodologia pode ser vista no trabalho de BIEGLER (1997) em que o problema de otimização dinâmica é solucionado em um *loop* iterativo interno, enquanto outro problema de otimização com inserção da posição dos coeficientes da variável de controle como variáveis de decisão é solucionado em um *loop* iterativo externo. Tal estratégia possibilita um procedimento de discretização que explora a estrutura do perfil de controle.

BALSA-CANTO *et al.* (2001) propôs a solução via método sequencial em que a cada passo do algoritmo de adaptação dobra-se o nível de parametrização da variável de controle, ideia semelhante à desenvolvida por BETTS e HUFFMAN (1999). BINDER (2002) utiliza esse procedimento de forma mais otimizada, em que as variáveis de estado são discretizadas e novos pontos de discretização são inseridos somente se os erros locais dos estados de diferenciação forem maiores que uma tolerância pré-estabelecida heurísticamente. HADYANTO *et al.* (2008) aplicou um procedimento semelhante, onde o nível de discretização é sucessivamente refinado de acordo com um critério de desempenho baseado no gradiente da função objetivo em relação às variáveis de controle. Um valor de *threshold* heurísticamente pré-determinado é utilizado como limite para continuidade do refinamento. Os autores utilizaram um critério de *threshold* em torno de 10–20% do gradiente da função objetivo.

A ideia da adaptação de malhas evoluiu para aplicação das wavelets (detalhadas no Capítulo 5) como ferramenta matemática de adaptação seletiva (SCHLEGEL, 2004; BINDER *et al.* 2000a; 2000b). O desafio de escolher uma malha ótima para solucionar

DAOPs surgiu da necessidade de reduzir o custo computacional dos algoritmos assim como possibilitar adaptação automática de acordo com o perfil de controle.

As wavelets⁸ são funções com certas propriedades matemáticas, como ortogonalidade e localização espacial, que permitem a representação de um conjunto de dados em diferentes níveis de resolução (ou aproximação).

Por consequência, possibilita-se analisar as principais características desse conjunto, havendo a possibilidade de manipular tais dados a fim de compactá-los ou remover possíveis ruídos ou dados irrelevantes. Em linhas gerais, o critério wavelet adaptativo se baseia em representar uma variável (de controle ou estado) por sua respectiva expansão em base wavelets, por exemplo:

$$u(t) \cong \sum_n \sum_m d_{n,m} \psi_{n,m}(t) \quad (4.13)$$

e

$$\psi_{n,m}(t) = 2^{n/2} \psi(2^n t - m) \quad (4.14)$$

em que:

ψ corresponde à função wavelet, $m \in \mathbb{Z}$ é o índice que mede a localização espacial (ou no tempo) e $n \in \mathbb{Z}$ é o índice que mede a resolução da wavelet.

Uma das principais características dessa representação é a possibilidade de desconsiderar muitos coeficientes d da expansão acima, já que as wavelets se caracterizam por serem funções localizadas, e gerarem representação esparsa de um conjunto de dados (DAUBECHIES, 1992). Quando o intuito é, por exemplo, comprimir o conjunto original de dados, deve-se considerar um limiar, ou *threshold*, para remover coeficientes supostamente pouco representativos.

BINDER (2002) introduziu funções wavelets para solucionar DAOPs. Nesse trabalho os autores utilizaram um esquema wavelet-Galerkin (VON WATZDORF *et al.*, 2006; LI e FANG, 2010; OROPEZA, 2007) para discretização das equações. Para selecionar o conjunto ótimo de bases as aproximações são monitoradas por uma métrica (*threshold*) pré-especificada para seleção de bases ótimas. O pesquisador mostrou que a

⁸ As wavelets são apresentadas detalhadamente no próximo capítulo.

discretização total do sistema de DAEs com funções wavelets pode resultar em um problema de otimização dinâmica no domínio wavelets, ideia semelhante à apresentada por CHEN e HSIAO (1999) e aprimorada recentemente por HAYA (2011) para discretização de problemas variacionais mais simples.

Wavelets também foram utilizadas para discretização *single-shooting* por BINDER *et al.* (2000); SCHLEGEL (2004) e SCHLEGEL *et al.* (2005) descritos abaixo.

SCHLEGEL (2004) propôs um algoritmo de reparametrização da variável de controle acoplado a um procedimento de detecção de mudança de arcos. Segundo o pesquisador, o procedimento wavelets adaptativo ainda apresenta deficiência em representar a variável de controle em torno de descontinuidades, resultando no acúmulo de muitos pontos em torno dessas regiões, tornando a solução oscilatória e muitas vezes desvantajosa de ser implementada. Em um dos exemplos apresentados o critério wavelets adaptativo apresentou um desempenho inferior ao critério com discretização sucessiva com estágios igualmente espaçados (considerado teoricamente menos vantajoso). Além disso, dificuldades também foram encontradas para correta detecção dos pontos de junção das trajetórias de controle, com soluções apresentando oscilações em torno dessas regiões. Para contornar tais dificuldades o autor implementou estratégias de reparametrização da variável de controle de acordo com identificação *a priori* dos arcos.

O exemplo seguinte ilustra a dificuldade encontrada por SCHLEGEL (2004) em justificar o uso de adaptação wavelets sobre uma estratégia de adaptação sucessiva equidistante, a princípio menos vantajosa que a proposta pelo pesquisador.

Exemplo 4.2. Biorreator Batelada (SCHLEGEL, 2004)

Esse problema foi solucionado numericamente por PARK e RAMIREZ (1988), BALSACANTO *et al.* (2001) e foi utilizado também como estudo-de-caso por SCHLEGEL (2004). O modelo é descrito pelas equações seguintes:

$$\frac{dP_M}{dt} = \Theta(\mu_x) \cdot (P_T - P_M) - \frac{F}{V} \cdot P_M, \quad P_M(0) = P_{M0} \quad (4.16)$$

$$\frac{dP_T}{dt} = f_P(S) \cdot X - \frac{F}{V} \cdot P_T, \quad P_T(0) = P_{T0} \quad (4.17)$$

$$\frac{dX}{dt} = \mu_x(S) \cdot X - \frac{F}{V} \cdot P_T, \quad P_T(0) = P_{T0} \quad (4.18)$$

$$\frac{dS}{dt} = -7,3 \cdot \mu_x(S) \cdot X - \frac{F}{V} \cdot (m - S), \quad S(0) = S_0 \quad (4.19)$$

$$\frac{dV}{dt} = F, \quad F(0) = F_0 \quad (4.20)$$

$$\Theta(\mu_x) = \frac{4,75 \cdot \mu_x}{0,12 + \mu_x} \quad (4.21)$$

$$f_P(S) = S \cdot \frac{e^{-5,01S}}{0,1 + S} \quad (4.22)$$

$$\mu_x(S) = \frac{21,88 \cdot S}{(S + 0,4)(S + 62,5)} \quad (4.23)$$

Logo o problema de otimização é definido como:

$$J = \max_u P_M(t_f) \cdot V(t_f) \quad (4.24)$$

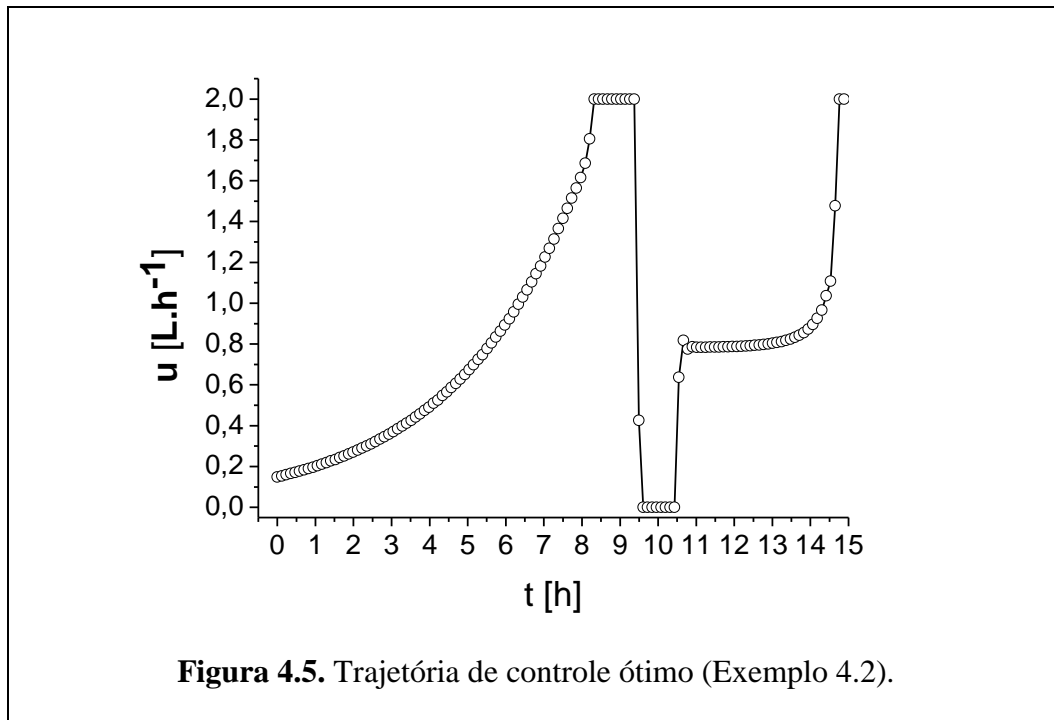
sujeito a:

$$u_{\min} \leq u \leq u_{\max} \quad (4.25)$$

em que $u = F$

O objetivo é maximizar a quantidade de proteína produzida, definida pela Equação 4.24.

O perfil de controle ótimo, com 128 estágios de discretização é mostrado na Figura 4.5. Note que o perfil de controle apresenta cinco regiões distintas. SCHLEGEL (2004) não obteve sucesso na aplicação das wavelets, obtendo um custo computacional 121,61% superior ao caso solucionado com 128 estágios (considerado como solução ótima). O pesquisador atribui tal dificuldade ao formato do perfil de controle que necessita de uma boa estimativa inicial de qualidade para rápida convergência.



A diferença da estratégia de SCHLEGEL (2004) para a estratégia de BINDER *et al.* (2000) se dá no critério de desempenho que avalia a estratégia de refinamento. Enquanto em BINDER *et al.* (2000) o gradiente da função objetivo é utilizado, em SCHLEGEL (2004) e SCHLEGEL *et al.* (2005) os valores dos coeficientes de parametrização são avaliados e rearranjados no domínio wavelets. Destaca-se que em ambos os trabalhos, critérios heurísticos são utilizados para a especificação dos valores de *threshold*.

SCHLEGEL *et al.* (2005) propuseram uma estratégia de solução dos problemas de otimização dinâmica baseada no método sequencial com funções base wavelets. Para isso apresentaram um software chamado ADOPT. Para comunicação com o modelo, o software ADOPT pode ser conectado a um padrão conhecido como ESO (do inglês *Equation Set Object*) que é capaz de comunicar todas as informações contidas do modelo de DAEs (ex.: número e valor das variáveis, resíduos, estrutura e valor da matriz Jacobiana, etc.). A estratégia, utilizada pelos pesquisadores, foi comunicar o ADOPT via plataforma CORBA *object bus*. Para a modelagem foi utilizado o software gPROMs[®], baseado em equações. Nesse trabalho, os pesquisadores evidenciaram o elevado custo computacional utilizando essa plataforma de comunicação entre os

softwares ADOPT e gPROMs[®], o que justifica a limitação para aplicações em tempo real.

HARTWICH e MARQUARDT (2010) implementaram um algoritmo paralelo para o cômputo das equações de sensibilidade no método *single-shooting*. O algoritmo foi aplicado a um modelo com mais de 13500 DAEs e 75 parâmetros. O custo computacional foi consideravelmente reduzido com a técnica proposta. FESKO (2012) e DRAG e STYCZEN (2012) também utilizaram algoritmos paralelos para solucionar DAOPs utilizando o método *multiple-shooting*.

Recentemente ASSASSA e MARQUARDT (2014) acoplaram a metodologia desenvolvida por SCHLEGEL *et al.* (2005) ao método *multiple-shooting*. Foram utilizados dois estudos de caso para comparar o desempenho computacional do mesmo algoritmo utilizado por SCHLEGEL *et al.* (2005). Os autores concluíram que o método *multiple-shooting* comprovou-se mais estável do que o método sequencial minimizando a ocorrência de instabilidades.

4.3. Wavelet-Thresholding

Em problemas onde que as soluções apresentam mudanças bruscas de comportamento, o uso de malhas regulares não é o mais conveniente. Se for usada uma malha refinada capaz de descrever com precisão as funções em zonas de mudanças bruscas, nas zonas de comportamento suave também serão utilizados muitos pontos, o que, além de não ser necessário, faz com que o custo computacional seja excessivamente elevado. Se for usada uma malha pouco refinada, o custo computacional diminui, assim como a qualidade dos resultados, devido a uma escassa representação da função próximo das irregularidades. Por essa razão, vários esquemas adaptativos têm sido propostos ao longo das últimas duas décadas (BLANCO, 2002).

Destaca-se que uma das principais dificuldades para implementação de DAOPs é o elevado custo computacional necessário para solucionar problemas resultantes de DAEs de elevada dimensão. Em engenharia química, principalmente, a modelagem de operações unitárias como torres de destilação, estimação de parâmetros ou

implementação de NMPCs pode resultar em sistemas de até milhares de variáveis de estado.

O objetivo deste estudo é propor novas estratégias para discretização adaptativa de DAOPs. Para isso, propõe-se um procedimento de discretização adaptativa baseado em bases wavelets. A inovação consiste em importar mecanismos utilizados para compressão de dados, já difundidos na área de processamento digital, estatística e tratamento de imagens, para seleção ótima das bases wavelets utilizadas para representação das variáveis do DAOP. De fato há um campo bastante específico na área de compressão de dados e remoção de ruídos que direciona precisamente como aplicar as wavelets adequadamente para tal tarefa. Há diversos estudos cuja proposta de compressão de dados sustenta-se em estratégias para estimação do nível de ruído ou informações descartáveis que possam ser eliminadas de acordo com um determinado valor de *threshold* (DONOHO, 1992; JANSEN, 2000; WIELAND, 2009). Há algumas regras para o cálculo do valor do *threshold* já bastante consolidadas. A literatura vem mostrando que as regras utilizadas para tal fim são determinantes para a seleção dos coeficientes de detalhes wavelets mais prospectivos, os quais são aptos a representar a maior parcela da energia de um determinado conjunto de dados.

Verificaremos neste trabalho que a importação desses mecanismos é essencial para a aplicação de estratégias de discretização adaptativa baseada em bases wavelets. Ademais, a importação dessas metodologias para solucionar DAOPs de engenharia química abre novas perspectivas para o desenvolvimento e ou melhoria de métodos numéricos.

A Figura 4.6 ilustra a aplicação da análise wavelets de um perfil de controle. Note que o perfil de controle original seria decomposto em duas camadas de resolução. A avaliação típica dos coeficientes da expansão wavelets do perfil ilustrado é mostrada na Figura 4.7.

Conforme mostrado na Figura 4.7, o perfil de detalhes segue uma distribuição que depende dos níveis de resolução. Intuitivamente, não é uma boa estratégia utilizar um limiar fixo para tratamento dos coeficientes. Tal seleção se baseia na escolha ótima

do *threshold*. Logo, cada faixa de resolução deve ser averiguada diferentemente, pois a significância dos coeficientes de detalhes wavelets varia com a resolução.

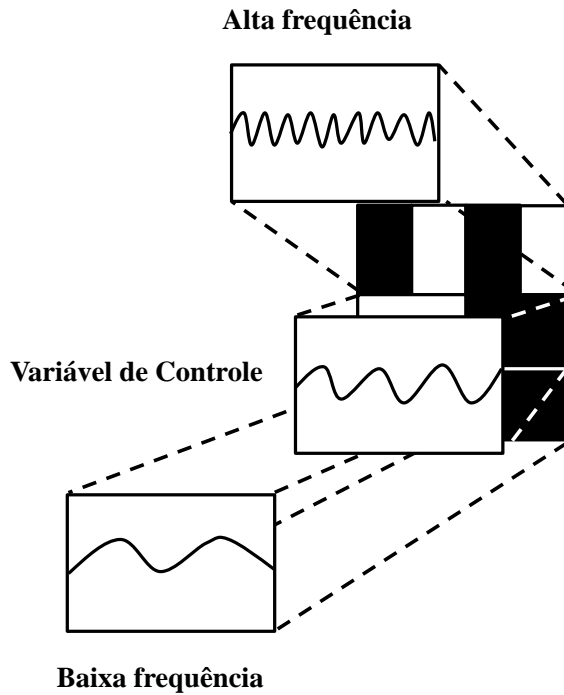


Figura 4.6. Multiresolução Wavelets: do processamento de imagens para otimização (detecção de perfis de alta e baixa frequência).

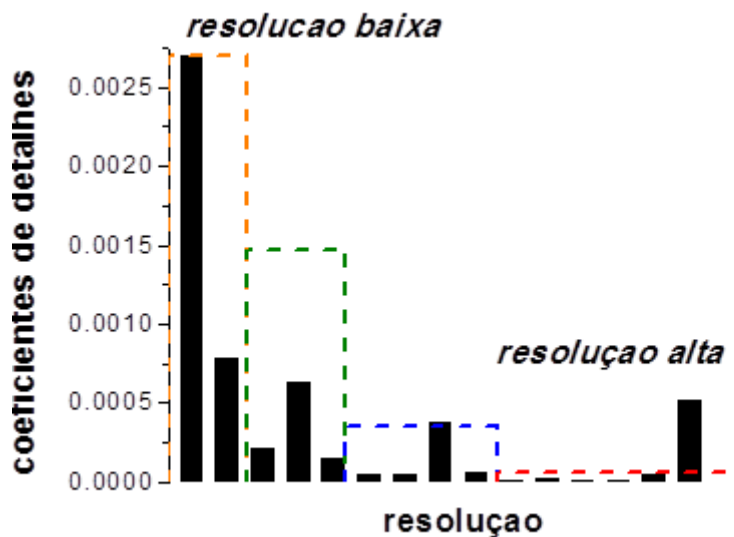


Figura 4.7. Ilustração: espectro de detalhes wavelets.

Por exemplo, nas camadas de resolução mais elevadas, coeficientes de baixa magnitude podem ser importantes. O inverso também pode acontecer nas camadas de

resolução mais baixas. Conseqüentemente, esse mecanismo deve ser executado cuidadosamente a fim de não degradar e tornar injustificável o uso desse procedimento. Percebe-se que os trabalhos reportados no Tópico 4.1 não discutem esses aspectos. Por exemplo, as dificuldades encontradas por SCHLEGEL (2004) e citadas em ASSASSA e MARQUARDT (2014) podem ser oriundas dessas limitações.

A estratégia, sugerida neste trabalho, consiste em acoplar o algoritmo *wavelet-thresholding* adaptativo (WtA) ao método *single-shooting*, no qual o nível de discretização das variáveis de controle evolui em elementos com dimensões variadas. Tal discretização se modifica a cada iteração do algoritmo para melhorar a acurácia da aproximação em determinados pontos específicos no domínio de controle (malha), mais especificamente em regiões de descontinuidade ou pontos de junção. Desse modo, o algoritmo proposto utiliza um procedimento automático para refinar ou para engrossar a malha, a fim de atingir a acurácia requerida com o objetivo de reduzir o esforço computacional do algoritmo de otimização e facilitar a etapa de integração do sistema DAE. Dessa maneira, a cada iteração a estimativa inicial do perfil de controle é aprimorada e conseqüentemente facilita-se a convergência para o perfil ótimo. A necessidade de refinar ou preservar o nível de discretização da malha depende da dinâmica do processo, a qual pode afetar a qualidade da aproximação (da variável de controle), e conseqüentemente, a estratégia de solução. O número e a posição dos pontos da discretização dependem da política adotada de compressão do vetor de pontos resultantes da discretização. Comparam-se aqui diferentes métricas para determinação do *threshold* configurado na etapa de compressão da malha.

A metodologia é consolidada em um capítulo de estudo de casos (Capítulo 7) no qual são abordados vários *benchmarks* de DAOPs, majoritariamente de engenharia química.

Capítulo 5

Introdução à Análise Wavelets

Neste Capítulo

Tópico 5.1. Introdução às wavelets;
Tópico 5.2. Evolução da transformada de Fourier para transformada wavelets;
Tópico 5.3. Principais propriedades das wavelets;
Tópico 5.4. Família de wavelets;
Tópico 5.5. Introdução ao procedimento de *thresholding*;
Tópico 5.6. Em busca do *threshold* ideal;
Tópico 5.7. Critérios *thresholdng*;
Tópico 5.8. Aplicações de Wavelets em Engenharia: da Solução de Sistemas de Equações à Otimização Dinâmica;
Tópico 5.9.Resumo do Capítulo.

O objetivo desse capítulo é apresentar as propriedades fundamentais das wavelets e levantar as suas principais características já com objetivo de introduzir as ferramentas numéricas que serão utilizadas para intervir nas etapas de discretização nos DAOPs. O **Tópico 5.1** apresenta uma breve introdução à teoria wavelets. No **Tópico 5.2** apresentam-se as características principais das wavelets e suas aplicações. No **Tópico 5.3** são introduzidos os conceitos da análise multirresolução e banco de filtros digitais que mostram como funciona o algoritmo da transformada discreta wavelets. O **Tópico 5.4** apresenta as diferentes bases (famílias) wavelets. O **Tópico 5.5** introduz os mecanismos do procedimento de wavelet *thresholding* para remoção de ruídos e compressão. O **Tópico 5.6** configura a idealização de um *threshold* ótimo. Já o **Tópico 5.7** introduz as estratégias *Sureshrink* e *Visushrink*. O **Tópico 5.8** apresenta algumas aplicações de wavelets como aproximação de funções e solução de equações diferenciais. Por fim, O **Tópico 5.9** sumariza o capítulo.

5.1. Introdução às Wavelets

5.1.1. Histórico

As wavelets são funções utilizadas como ferramentas para processamento ou manipulação de dados com o objetivo de analisar dados em diferentes escalas de resolução, visando principalmente detectar transitoriedades, remover ruídos e compactar dados. Essas funções possuem certas propriedades especiais que as tornam úteis para serem utilizadas como bases de expansões de funções e também como análise numérica para a solução de equações diferenciais (BEYLKIN, 1993). Segundo MISITI *et al.* (2007), as transformadas wavelets podem ser interpretadas como mecanismos para decompor dados em suas partes constituintes, permitindo analisá-los em diferentes faixas de frequências e espaço.

O conceito moderno de wavelets pode ser visto como uma síntese de ideias originadas durante os últimos 40 anos, proveniente de diferentes áreas da engenharia (VERTELLI & KOVAČEVIĆ, 2007) que foram contribuindo de forma independente para o avanço desse tema. Dentre vários nomes, pode-se destacar DONOHO (1992) e DAUBECHIES (1992) como grandes colaboradores e também responsáveis para a inserção das wavelets como ferramenta para solucionar problemas de engenharia. BEYLKIN (1993) foi pioneiro em usar as wavelets para compressão de operadores de integração e derivação de equações. DAUBECHIES (1992) desponta como uma grande contribuição para os avanços da teoria de Análise Wavelets e é utilizado como maior referência para a introdução teórica abordada neste capítulo.

Embora a primeira menção às wavelets tenha acontecido em 1909, por Alfred Haar, as funções Haar ficaram no ostracismo por muitos anos e, por um período muito longo, elas continuaram a ser a única base ortonormal de wavelets conhecida. Porém, em 1985, Stephane Mallat concedeu às wavelets um grande impulso através de seu trabalho em processamento digital de imagens e, inspirado nesses resultados, Meyer, em 1986 construiu a primeira wavelet com propriedades de suavidade, aspecto relevante discutido no Tópico 5.3. Ingrid Daubechies descobriu toda uma nova classe de wavelets, que é uma versão bem mais aprimorada das wavelets Haar, porém bastante simples de serem aplicadas computacionalmente.

Pode-se entender que a aproximação de uma função por funções-base wavelets é uma evolução da expansão em séries de Fourier. Nas séries de Fourier, o objetivo básico é de aproximar uma função $f(t)$ por uma combinação linear de componentes senoidais, cada um com uma determinada frequência. Os aspectos computacionais das séries de Fourier são especialmente atrativos, principalmente por possibilitar a decomposição de dados em diferentes frequências (MORETTIN, 1999).

Para o desenvolvimento dos tópicos seguintes, destacam-se o livro de DAUBECHIES (1992), MORETTIN (1999) e BURRUS *et al.* (1998). Os trabalhos de KEISER (2005), FREIKAMP (1999), BLANCO (2002), DANGETI (2003) e CAUMO (2006) também serviram de base para a confecção deste capítulo.

5.1.2. Princípios Básicos

A transformada wavelets de uma função contínua pode ser escrita como:

$$z_{a,b} = \|a\|^{-1/2} \int_{-\infty}^{\infty} f(t) \psi\left(\frac{t-b}{a}\right) dt, \quad a, b \in \mathbb{R}, \quad a \neq 0 \quad (5.1)^9$$

e de um sinal discreto (transformada wavelets discreta):

$$d_{n,m} = \|a_0\|^{n/2} \int_{-\infty}^{\infty} f(t) \psi(a_0^n t - mb_0) dt \quad (5.2)$$

em que $a = a_0^{-n}$, $b = mb_0 a_0^n$, com $a_0 > 0$ e $b_0 > 0$ fixos, com $m, n \in \mathbb{Z}$ a função ψ é conhecida como wavelet mãe e $t \in \mathbb{R}$ é a variável tempo (ou espaço).

Nota-se da Equação (5.2) que a transformada wavelets é indexada por dois parâmetros. A mudança do parâmetro n cobre diferentes níveis de frequência, enquanto a variação do parâmetro m move a localização da análise no tempo (ou espaço) (ver o esquema da Figura 5.1). Essa transformada é classificada como DWT (*Discrete Wavelets Transform*). Definindo a função wavelet indexada por esses dois parâmetros:

$$\psi_{n,m} = \|a_0\|^{n/2} \psi(a_0^n t - mb_0) \quad (5.3)$$

reescreve-se a Equação (5.2) como:

⁹ Em que $\| \cdot \|$ é a norma euclidiana.

$$d_{n,m} = \int_{-\infty}^{\infty} f(t)\psi_{n,m} dt \quad (5.4)$$

em que $d_{n,m}$ é a transformada wavelets discreta de $f(t)$, que resulta em um vetor de coeficientes \mathbf{d} . Conseqüentemente, a expansão wavelets (transformada inversa) da Equação (5.2) é escrita como:

$$f(t) = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} d_{n,m}\psi_{n,m}(t) \quad (5.5)$$

A magnitude dos coeficientes $d_{n,m}$ traduz a energia do sinal em determinada posição-escala, de acordo com o teorema de Parseval (MORETTIN, 1999):

$$\int_{-\infty}^{\infty} \|f(t)\|^2 dt = \sum_{n=-\infty}^{\infty} \sum_{m=-\infty}^{\infty} \|d_{n,m}\|^2 \quad (5.6)$$

Conforme mostrado na Equação (5.3), $\psi_{n,m} = |a_0|^{n/2} \psi(a_0^n t - mb_0)$, é possível calcular o valor da wavelet a partir do encolhimento (ou expansão) e translação de uma simples função wavelet-mãe: ψ . Ao mudar o índice m altera-se a localização da wavelet no domínio. Esse procedimento permite representar a localização dos eventos no tempo (ou espaço). Com a mudança do índice n o formato da wavelet muda de escala (encolhendo-se ou esticando-se a função), permitindo uma representação dos detalhes em resoluções diferentes. Esse comportamento é ilustrado na Figura 5.1.

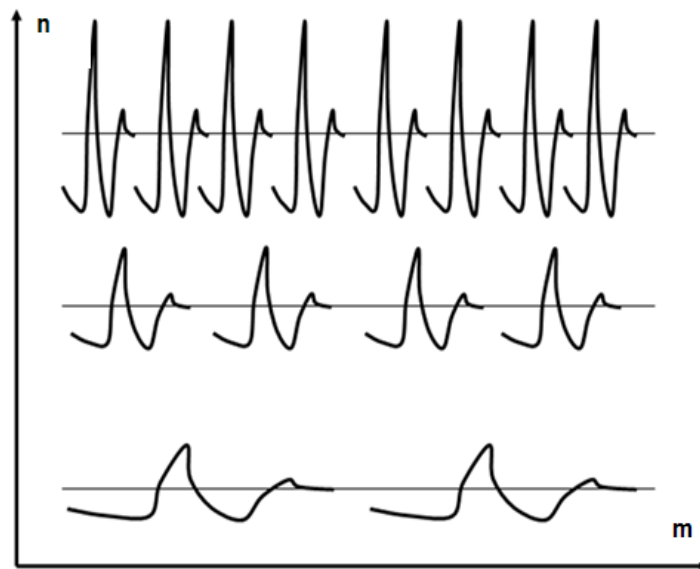
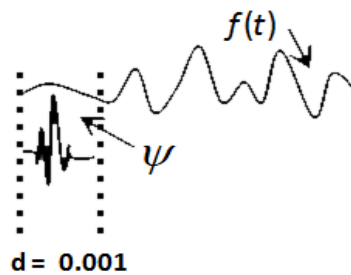


Figura 5.1. Representação de uma wavelets em diferentes conjuntos localização-escala (baseado em BURRUS *et al.*, 1998).

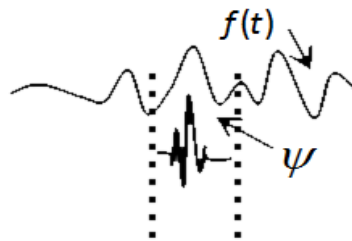
Considerando uma função contínua $f(t)$, conforme ilustrado na Figura 5.2, a primeira etapa (Figura 5.2(a)) consiste em comparar uma determinada wavelet com uma região da função, por meio do produto interno:

$$d_{n,m}(f) = \int f(t)\psi_{n,m} dt \quad (5.7)$$

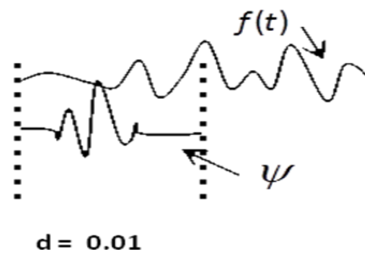
A segunda etapa (Figura 5.2(b)) é transladar a wavelet e realizar o mesmo procedimento anterior. Essa etapa é repetida até a total cobertura do domínio. A terceira etapa (Figura 5.2(c)) é repetir as etapas anteriores para uma escala diferente.



(a) Cálculo do coeficiente wavelet em uma determinada região do domínio.



(b) Translação da wavelet mãe.



(c) Mudança de escala.

Figura 5.2. Ilustração do processo de decomposição wavelets (baseada na ilustração da MATHWORKS, 2009).

Conforme ilustrado na Figura 5.2, à medida que se realiza a expansão, são obtidos os coeficientes resultantes de diferentes escalas em todas as regiões do domínio.

As wavelets de baixa escala correspondem a frequências elevadas enquanto wavelets de alta escala correspondem à baixa frequência.

5.2. Principais Propriedades das Wavelets

5.2.1 Ortogonalidade

De acordo com a escolha da função wavelet ψ e dos parâmetros a_0, b_0 , é possível construir bases ortogonais em $L^2(\mathbb{R})$ ¹⁰. Em particular, escolhendo-se $a_0 = 2$ e $b_0 = 1$, a Equação (5.3) é escrita como:

$$\psi_{n,m} = 2^{n/2} \psi(2^n t - m) \quad (5.8)$$

A escolha desses valores (de acordo com o Teorema de Shannon, MALLAT, 1989) garante a ortogonalidade das funções nas múltiplas escalas. A vantagem de se trabalhar com bases ortogonais em $L^2(\mathbb{R})$ é permitir a reconstrução perfeita (sem perda de informação) dos dados a partir dos coeficientes da transformada. Logo, $\psi_{n,m}$ é uma base ortogonal se:

$$\langle \psi_{n,m}, \psi_{j,k} \rangle = 0, \quad m \neq k, \quad n \neq j \quad (5.9)$$

em que $m, k, n, j \in \mathbb{Z}$.

5.2.2 Admissibilidade

A função ψ é caracterizada de fato como uma wavelet se for verificada a sua condição de admissibilidade, ou seja, para $\psi \in \mathbb{R}$ a condição de admissibilidade é escrita como:

$$\int_{-\infty}^{\infty} \psi(t) dt = 0 \quad (5.10)$$

A Equação (5.10) mostra que o valor médio da wavelet no domínio temporal é nulo, o que assegura o comportamento ondulatório da função ψ . O comportamento ondulatório está relacionado à wavelet ter suporte compacto, garantindo que a mesma possui localização espacial, ou seja, a função é diferente de zero somente em uma determinada região limitada.

5.2.3 Caracterização Local de Regularidade

¹⁰ Espaço de funções de quadrado integrável.

Outra importante propriedade das wavelets é a regularidade (que mede a suavidade). Para determinar a regularidade de uma função wavelet, o número de momentos nulos da mesma é crucial. Uma função wavelet tem L momentos nulos se:

$$\int_{-\infty}^{\infty} t^p \psi(t) dt = 0 \quad p = 0, \dots, L-1 \quad (5.11)$$

A vantagem de uma wavelet ter um número elevado de momentos nulos significa a mesma apresentar uma alta capacidade de compressibilidade porque os coeficientes wavelets das escalas mais finas de uma função $f(t)$ são essencialmente nulos onde a função é suave. Isso significa que quanto maior L mais suave (ou regular) é a função e maior o número de funções possíveis de serem representadas com fidelidade, ou seja, para polinômios com ordem inferior a L , os coeficientes de detalhes serão nulos. Para polinômios de ordem superior a L , os coeficientes wavelets terão valores elevados onde $f(t)$ não for suave, o que permite usar wavelets para detectar discontinuidades. Esse princípio é consequência do seguinte teorema:

$$\|d_{n,m}\| \leq C \cdot 2^{-n(L-1)/2} \int_{-\infty}^{\infty} \|f\|_{L^2} \quad (5.12)$$

em que C é uma constante que depende de ψ e de $\|f\|_{L^2}$ (DAUBECHIES, 1992).

A ordem de grandeza dos coeficientes wavelets está associada à suavidade da função $f(t)$ analisada no suporte de ψ e ao seu número de momentos nulos. Dessa maneira os coeficientes wavelets podem ser indicativos de regularidade das funções analisadas, pois tipicamente são desprezíveis em regiões de suavidade e maiores em regiões de singularidades (DAUBECHIES, 1992).

5.3. Análise Multirresolução (AMR)

5.3.1 Interpretação das Wavelets como AMR

Nessa seção deduz-se a representação das wavelets conforme a teoria de análise multirresolução (AMR). A AMR de um conjunto de dados pode ser compreendida como uma sequência de subespaços $\{V_n, n \in \mathbb{Z}\}$ que aproximam $L^2(\mathbb{R})$, de acordo com as seguintes premissas:

$$(i) \quad V_0 \subset V_1 \subset V_2 \subset \dots \subset L^2(\mathbb{R}) \quad (5.13a)$$

$$(ii) L^2(\mathbb{R}) \cup_n V_n \tag{5.13b}$$

$$(iii) \cap_n V_n = \{0\} \tag{5.13c}$$

A Equação (5.13a) indica que ao passar de um subespaço V_0 para um V_1 , novos detalhes são acrescentados à aproximação de um sinal, indicando que o subespaço que representa um nível superior contém mais informação.

A Equação (5.13b) indica que a união dos subespaços é capaz de representar as funções pertencentes ao espaço de funções de quadrado integrável $L^2(\mathbb{R})$.

A Equação (5.13c) indica que a interseção nos subespaços resulta em um espaço nulo.

O espaço V_n é gerado pela seguinte expansão:

$$V_n = span_m \{ \phi_{n,m}(t) \} \tag{5.14}$$

em que $span$ significa a expansão de uma função $\phi_{n,m}$, definida como função wavelet escala:

$$\phi_{n,m}(t) = 2^{n/2} \phi(2^n t - m) \tag{5.15}$$

A Equação (5.15) mostra que se uma função $f^v(t) \in V_n$ então:

$$f^v(t) = \sum_m c_{n,m} \phi_{n,m}(t) \tag{5.16}$$

em que f^v significa uma função expandida no espaço V .

De acordo com a definição de V_n na Equação (5.14), os subespaços satisfazem à seguinte condição:

$$f^v(t) \in V_n \Leftrightarrow f^v(2t) \in V_{n+1} \tag{5.17}$$

o que assegura que os elementos de um espaço são suas versões “dilatadas” no espaço seguinte, conforme ilustrado na Figura 5.3.

Sabendo-se que:

$$\phi_{0,m}(t) = 2^{0/2} \phi(2^0 t - m) = \phi(t - m)$$

e

$$\phi_{1,m}(t) = 2^{1/2} \phi(2t - m) = \sqrt{2} \phi(2t - m) \tag{5.18}$$

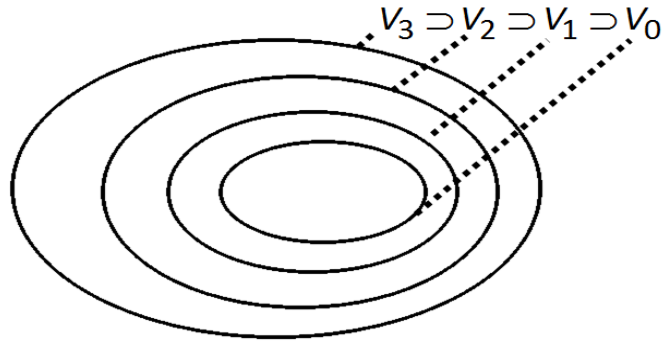


Figura 5.3. Representação da sequência de subespaços expandidos por funções escala.

De acordo com a Figura 5.3, observa-se que se $f^v(t) \in V_0$ então $f^v(t) \in V_1$.

Isso mostra que $\phi(t)$ em um subespaço V_n pode ser representado como uma soma ponderada de $\phi(2t)$ em uma escala V_{n+1} transladada por $k \in \mathbb{Z}$, ou seja:

$$\phi(t) = \sum_k \ell_f(k) \sqrt{2} \phi(2t - k) \quad (5.19)$$

em que os coeficientes $\ell_f(k)$ são uma sequência de números reais ou complexos, nomeados como coeficientes ou filtros escala e $\sqrt{2}$ mantém a norma unitária da função $\phi(t)$ em diferentes escalas (BURRUS *et al.*, 1998). Logo, o filtro $\ell_f(k)$ dependerá da expressão da função escala (no Exemplo 5.1 ilustraremos o procedimento para o cômputo do filtro $\ell_f(k)$).

Na AMR, se um conjunto de dados puder ser representado por uma soma ponderada de funções-base $\phi(t-k)$, então um conjunto maior, incluindo o original poderá ser representado por uma soma ponderada de $\phi(2t-k)$. Esse aspecto é o princípio que rege a aplicabilidade das wavelets como ferramenta para análise de sinais ou funções em múltiplas frequências. Logo, uma informação perdida de V_{n+1} para V_n é representada no subespaço de detalhes W_n que é o complemento ortogonal de V_n em V_{n+1} , ou seja, $V_{n+1} = V_n \oplus W_n$ sendo \oplus a soma direta¹¹ dos subespaços, conforme ilustra a Figura 5.4.

O subespaço W_n é gerado pela seguinte expansão:

$$W_n = \underset{m}{span} \{ \psi_{n,m}(t) \} \quad (5.20)$$

¹¹ Soma direta \oplus : Soma de dois espaços ortogonais é nula (KREYSZIG, 1989).

em que $\psi_{n,m}(t) = 2^{n/2} \psi(2^n t - m)$ (5.21)

A Equação (5.20) mostra que se $f^w(t) \in W_n$ então:

$$f^w(t) = \sum_m d_{n,m} \psi_{n,m}(t) \tag{5.22}$$

em que f^w significa uma função expandida no espaço W .

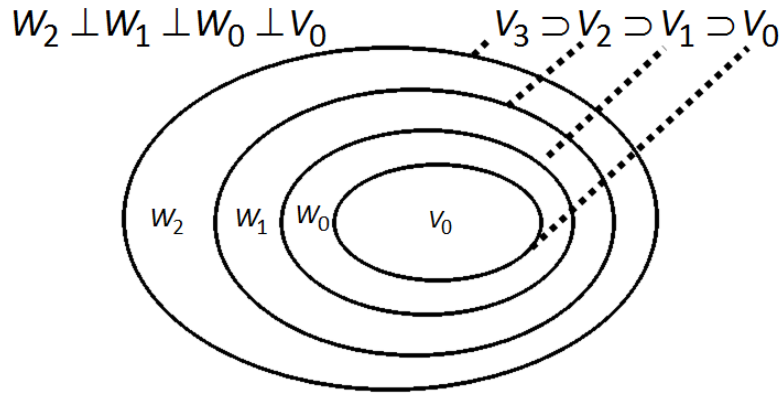


Figura 5.4. Representação da sequência de subespaços expandidos por funções escala e wavelets.

De acordo com a definição de W_n na Equação (5.20), os subespaços satisfazem à seguinte condição:

$$f^w(t) \in W_n \iff f^w(2t) \in W_{n+1}$$

Como $V_{n+1} = V_n \oplus W_n$, $W_n \subset V_{n+1}$, logo a função wavelets pode ser representada por uma soma ponderada da função escala transladada por k :

$$\psi(t) = \sum_k h_f(k) \sqrt{2} \phi(2t - k) \tag{5.23}$$

em que os coeficientes $h_f(k)$ são uma sequência de números reais ou complexos, nomeados como coeficientes de escala ou filtros wavelets. Como W_n é o complemento ortogonal de V_n em V_{n+1} surge a seguinte relação:

$$h_f(k) = (-1)^k \ell_f(1-k) \tag{5.24}$$

que conecta os filtros escala aos filtros wavelets.

Logo, a AMR mostra que:

$$f_n^e = f_{n+1}^e + f_{n+1}^w \tag{5.25}$$

e portanto:

$$f = f_n^e = f_{n+1}^w + f_{n+2}^w + \dots + f_{n+N}^w + f_{n+N}^e \quad (5.26)$$

ou seja,

$$L_2(\mathbb{R}) = V_{n_0} \oplus \bigoplus_{n \geq n_0} W_n \quad \text{para } n_0 \in \mathbb{Z} \quad (5.27)$$

em que n_0 é a resolução ou nível mais baixo e N é a resolução (escala) mais elevada.

A Equação (5.27) indica que todos os subespaços podem ser deduzidos a partir do subespaço de referência V_{n_0} adicionado a seus detalhes, caracterizador por W_n . De acordo com esse raciocínio, uma função pode ser aproximada da seguinte maneira:

$$f(t) = \sum_m c_{n_0,m} \phi_{n_0,m}(t) + \sum_{n \geq n_0} \sum_m d_{n,m} \psi_{n,m}(t) \quad (5.28)$$

em que:

$$c_{n_0,m} = \int_{-\infty}^{\infty} f(t) \phi_{n_0,m}(t) dt \quad (5.29)$$

e

$$d_{n,m} = \int_{-\infty}^{\infty} f(t) \psi_{n,m}(t) dt \quad (5.30)$$

são os coeficientes wavelets de aproximação e detalhes, respectivamente. n e m são relacionados por $m = 0, \dots, 2^n - 1$ e $n = 1, \dots, N$.

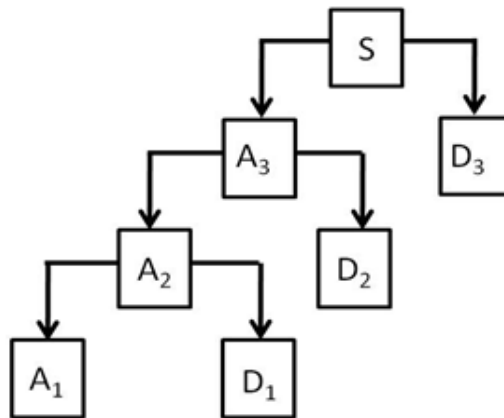


Figura 5.5. Estrutura multirresolução de um conjunto de dados S .

O processo de decomposição pode ser iterativo, com sucessivas aproximações, decompondo o sinal em componentes com resoluções cada vez maiores, conforme esquematizado na Figura 5.5.

Na Figura 5.5 o conjunto de dados S é decomposto em conjuntos de aproximação A e detalhes D . A aproximação pode ser interpretada como a faixa de baixa frequência. Já os detalhes são a parte de alta frequência. Logo:

$$S = A_3 + D_3 = A_2 + D_2 + D_1 = A_3 + D_3 + D_2 + D_1 \quad (5.31)$$

Para exemplificar o funcionamento básico da AMR, considere o Exemplo 5.1 que mostra como se obter os coeficientes de filtro da função wavelets Haar:

Exemplo 5.1. Obtenção dos coeficientes de filtro da wavelet Haar (BURRUS *et al.*, 1998)

A função wavelet escala Haar, para o nível V_0 , pode ser representada por:

$$\phi(t) = \begin{cases} 1, & 0 \leq t \leq 1 \\ 0, & \text{caso contrário} \end{cases} \quad (5.32)$$

Como $\phi(t) = \sum_k \ell_f(k) \sqrt{2} \phi(2t-k)$

$$\begin{aligned} \phi(t) &= \ell_f(0) \sqrt{2} \phi(2t-0) + \ell_f(1) \sqrt{2} \phi(2t-1) + \ell_f(2) \sqrt{2} \phi(2t-2) + \dots \\ &\dots + \ell_f(K) \sqrt{2} \phi(2t-K) \end{aligned} \quad (5.33)$$

Para $t = 0$

$$\phi(0) = \ell_f(0) \sqrt{2} \phi(0) + \ell_f(1) \sqrt{2} \phi(-1) + \ell_f(2) \sqrt{2} \phi(-2)$$

Logo:

$$1 = \ell_f(0) \sqrt{2} \cdot 1 + \ell_f(1) \sqrt{2} \cdot 0 + \ell_f(2) \sqrt{2} \cdot 0 \quad (5.34a)$$

Para $t = 1$

$$\phi(1) = \ell_f(0) \sqrt{2} \phi(2) + \ell_f(1) \sqrt{2} \phi(1) + \ell_f(2) \sqrt{2} \phi(0) \quad (5.34b)$$

Logo:

$$1 = \ell_f(0) \sqrt{2} \cdot 0 + \ell_f(1) \sqrt{2} \cdot 1 + \ell_f(2) \sqrt{2} \cdot 0 \quad (5.35)$$

Finalmente:

$$\ell_f(0) = 1/\sqrt{2}; \ell_f(1) = 1/\sqrt{2} \quad (5.36)$$

De acordo com a Equação (5.24):

$$h_f(k) = (-1)^k \ell_f(1-k) \quad (5.24)$$

os coeficientes de filtro wavelets resultam em:

$$h_f(0) = (-1)^0 \ell_f(1-0) = \ell_f(1) = 1/\sqrt{2}$$

$$h_f(1) = (-1)^1 \ell_f(1-1) = -\ell_f(0) = -1/\sqrt{2} \quad (5.37)$$

Em resumo, os coeficientes de filtro Haar \mathbf{h}_f são:

$$h_f(0) = 1/\sqrt{2} \text{ e } h_f(1) = -1/\sqrt{2} \quad (5.38)$$

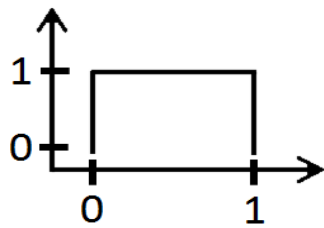
e os filtros ℓ_f são:

$$\ell_f(0) = 1/\sqrt{2} \text{ e } \ell_f(1) = 1/\sqrt{2} \quad (5.39)$$

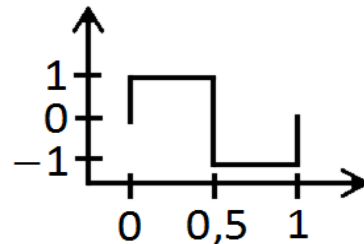
Como $\psi(t) = \sum_k h_f(k) \sqrt{2} \phi(2t-k)$, a função wavelet Haar, para o nível W_0 , pode ser escrita como:

$$\psi(t) = \begin{cases} 1, & 0 \leq t \leq 1/2 \\ -1, & 1/2 \leq t \leq 1 \\ 0, & t < 0 \text{ ou } t > 1 \end{cases} \quad (5.40)$$

O formato dessas funções é representado na Figura 5.6:



Função wavelet escala Haar.



Função wavelet mãe Haar.

Figura 5.6 Wavelet Haar.

Os filtros Haar representam como se relaciona a wavelet de um nível de resolução com outro. No próximo tópico explica-se como usar essas informações para representação em múltiplas resoluções.

5.3.2 Banco de Filtros

Devido à estrutura AMR, não é necessário trabalhar diretamente com a função-escala ou wavelet, graças às propriedades multirresolução. Dessa maneira, com o conhecimento dos coeficientes de filtro, ℓ_f e \mathbf{h}_f e os coeficientes da transformada wavelets, \mathbf{c} e \mathbf{d} , é possível mapear todo o espectro de um determinado conjunto de dados. Essa característica motivou o surgimento do algoritmo para aplicação da transformada wavelets discreta (DWT) proposto por MALLAT (1989). A estrutura básica desse algoritmo é explicada nesse tópico.

Considerando a Equação (5.19), $\phi(t) = \sum_k \ell_f(k) \sqrt{2} \phi(2t-k)$, ao dilatar e transladar a variável t temos:

$$\phi(2^n t - m) = \sum_k \ell_f(k) \sqrt{2} \phi(2(2^n t - m) - k) = \sum_k \ell_f(k) \sqrt{2} \phi(2(2^n t - m) - k) \quad (5.41)$$

que, após a mudança de variáveis $i = 2m + k$ resulta em:

$$\phi(2^n t - m) = \sum_i \ell_f(i - 2m) \sqrt{2} \phi(2^{n+1} t - i) \quad (5.42)$$

Ao considerar que:

$$V_n = \text{span}_m \left\{ 2^{n/2} \phi(2^n t - m) \right\} \quad (5.43)$$

então:

$$f(t) = \sum_m c_{n+1,m} 2^{(n+1)/2} \phi(2^{n+1} t - m) \quad (5.44)$$

Em um subespaço com resolução inferior V_n as wavelets são necessárias para representar os detalhes não disponíveis na escala n . Sabe-se que $V_{n+1} = V_n \oplus W_n$, logo:

$$f(t) = \sum_m d_{n,m} 2^{n/2} \psi(2^n t - m) + \sum_m c_{n,m} 2^{n/2} \phi(2^n t - m) \quad (5.45)$$

onde $2^n t$ mantém a norma unitária das funções-base em várias escalas (resoluções). Se as funções wavelets e escala forem ortonormais, os coeficientes escalas de uma determinada resolução n são dados por:

$$c_{n,m} = \langle f(t), \phi_{n,m}(t) \rangle = \int_{-\infty}^{\infty} f(t) 2^{n/2} \phi(2^n t - m) dt \quad (5.46)$$

Ao substituir a Equação (5.43)

$$\phi(2^n t - m) = \sum_i \ell_f(i - 2m) \sqrt{2} \phi(2^{n+1} t - i) \quad (5.43)$$

na Equação (5.46) resulta em:

$$c_{n,m} = \sum_i \ell_f(i - 2m) \underbrace{\int_{-\infty}^{\infty} f(t) 2^{(n+1)/2} \phi(2^{n+1} t - i) dt}_{c_{n+1,i}} \quad (5.47)$$

A Equação (5.47) é o produto interno da função $f(t)$ com a função escala expandida no subespaço V_{n+1} . Logo, pode-se escrever essa Equação (5.47) como:

$$c_{n,m} = \sum_i \ell_f(i-2m)c_{n+1,i} \quad (5.48)$$

Consequentemente a relação correspondente com os coeficientes wavelets é:

$$d_{n,m} = \sum_i h_f(i-2m)c_{n+1,i} \quad (5.49)$$

em que $i = 2m + k$.

MALLAT (1989) formulou um algoritmo que associa a obtenção dos coeficientes wavelets a partir de uma sequência de operações estruturadas como banco de filtros conhecidos como FIR (*Finite Impulse Response*). Esse procedimento permite relacionar a transformada wavelets à área de processamento digital, a qual usa artifícios de banco de filtros digitais para tratamento de sinais discretos. Um filtro FIR pode ser entendido como um operador linear aplicado à convolução de um conjunto de dados discretos:

$$y(t_j) = \sum_{k=0}^K \ell_f(k)x(t_j-k), \quad j=1, \dots, J \quad (5.50)$$

onde K é a ordem do filtro.

Duas operações básica de filtros FIR são os operadores *down-sampler* ou decimação e *up-sampler* ou interpolação.

No processo de *down-sampling* a dimensão de um conjunto de dados de entrada é reduzida a um determinado fator. Esse processo é simbolicamente representado na Figura 5.7 e exemplificado na Figura 5.8.

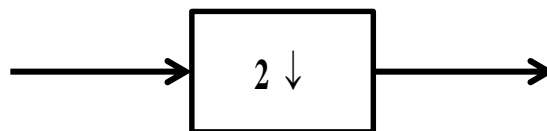


Figura 5.7. Downsampler (decimação) com fator 2.

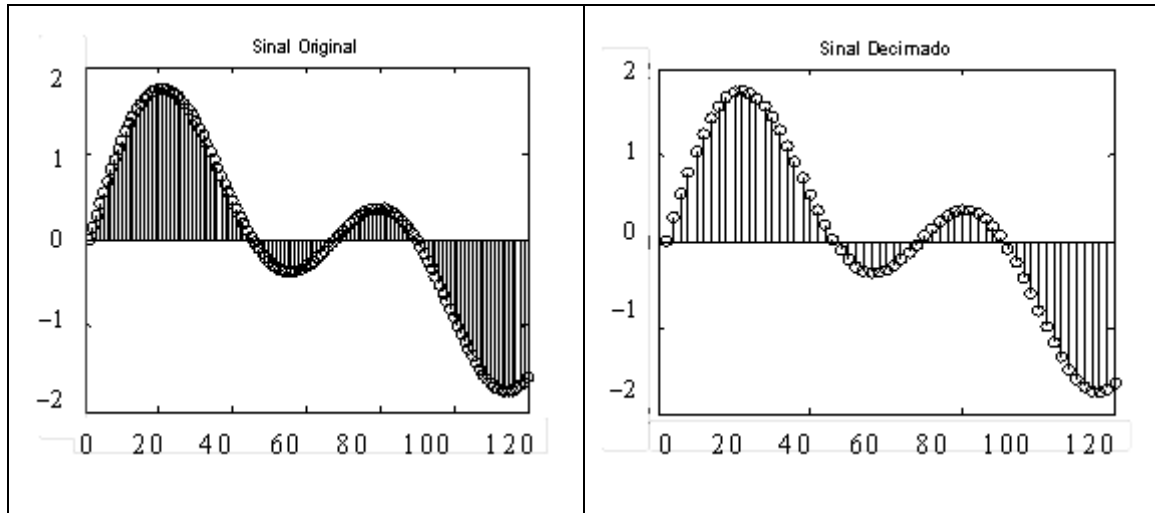


Figura 5.8. Exemplo da decimação de um sinal discreto.

É possível observar que na decimação com fator 2 metade da informação é perdida. Esse é o princípio que rege a decomposição de um conjunto de dados de um nível refinado (maior escala ou resolução) para um nível mais esparsa (menor escala ou resolução). A implementação dessas equações na estrutura de banco de filtros é ilustrada na Figura 5.9 que representa a análise ou decomposição de um conjunto de coeficientes c_{n+1} .

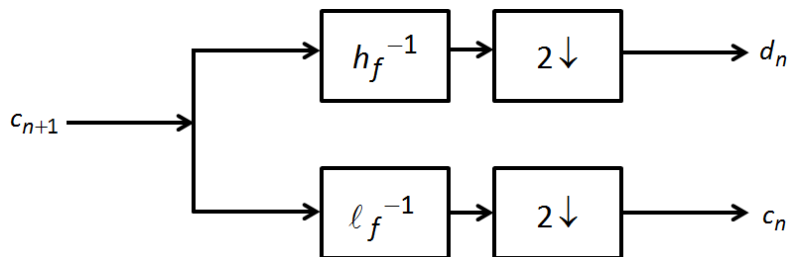


Figura 5.9. Banco de filtros de análise de c_{n+1} .

A estrutura de decomposição pode ser compreendida então por uma sequência de filtros passa baixa l_f e passa alta h_f decimados por uma fator 2, separando-se assim os detalhes não compreendidos em c_n .

De acordo com a formulação desenvolvida no início desse tópico, a equação de síntese ou reconstrução pode ser escrita como:

$$c_{n+1,m} = \sum_i c_{n,i} h_f(m-2i) + \sum_i d_{n,i} l_f(m-2i) \tag{5.51}$$

Nesse caso ocorre o processo inverso da decomposição, em que agora o objetivo é reconstruir o sinal a partir das escalas de menor resolução por meio da adição de

detalhes. A reconstrução é definida como uma sequência de interpolações (*up-sampling*). O processo *up-sampling* pode ser interpretado como o procedimento de aumentar a dimensão de um sinal de entrada ao inserir zeros entre os valores originais (para uma taxa de 2), ou seja:

$$\begin{aligned} y(2t_j) &= x(t_j) \\ y(2t_j + 1) &= 0 \end{aligned} \tag{5.52}$$

Esse processo é simbolicamente representado na Figura 10 e exemplificado na Figura 5.11.

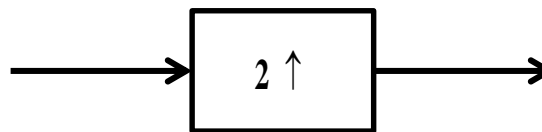


Figura 5.10. Interpolação (*up-sampling*) com fator 2.

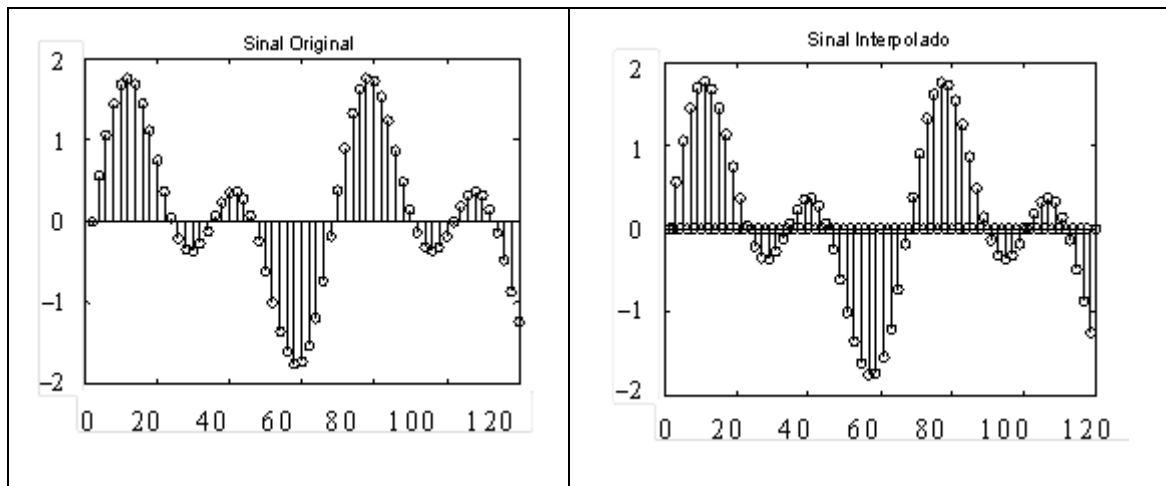


Figura 5.11. Exemplo da interpolação de um sinal.

O processo de reconstrução do sinal é representado na Figura 5.12:

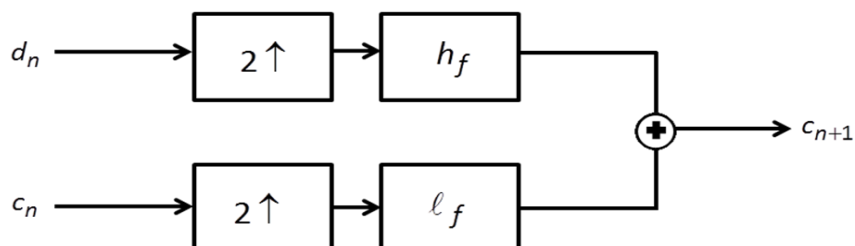


Figura 5.12. Banco de filtros de reconstrução de c_{n+1} .

Finalmente, o processo de decomposição e síntese a partir dos coeficientes da DWT é esquematizado na Figura 5.13:

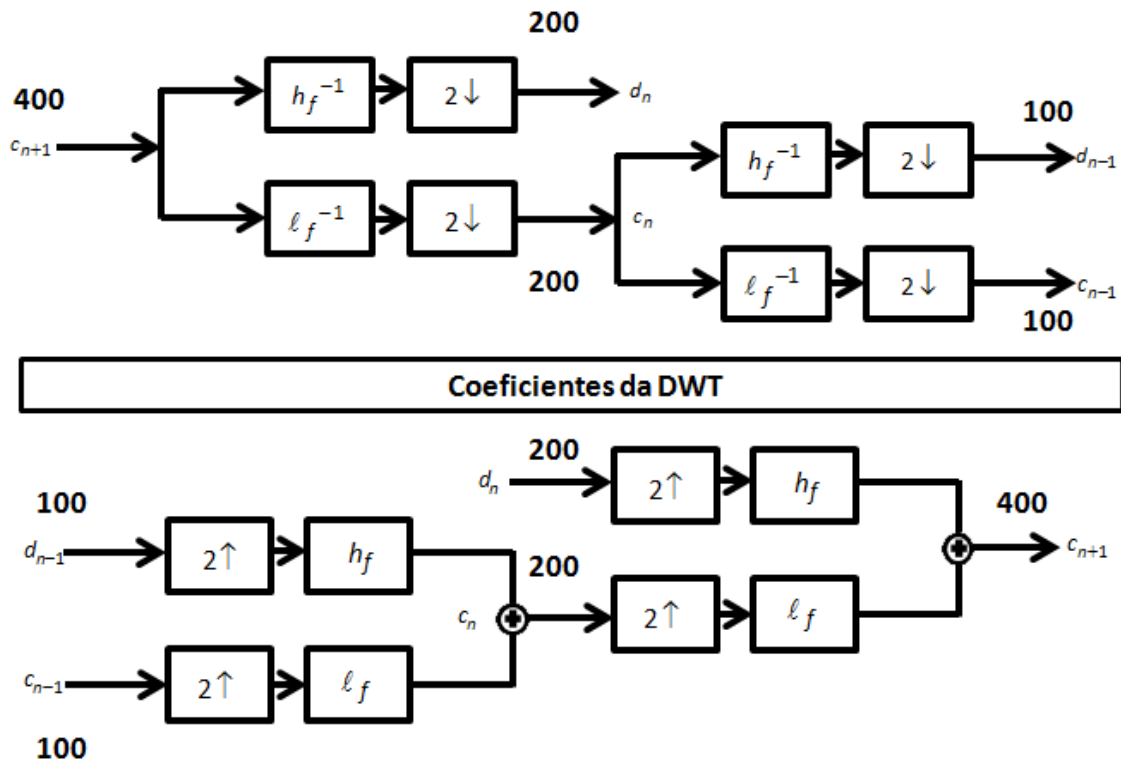


Figura 5.13. Filtros de decomposição e síntese.

Na explicação da Figura 5.13 observa-se que nenhuma informação é perdida nas etapas de decomposição e síntese. Os coeficientes na escala de maior resolução possuem tamanho 400 e a cada etapa de decomposição é decimado pela metade. Na etapa de síntese toda a informação é recuperada.

Existem diversas estratégias para estimar a escala inicial n para se obter c_{n+1} . À primeira vista nota-se que:

$$c_{n,m} = \langle f(t), \phi_{n,m}(t) \rangle = \int_{-\infty}^{\infty} f(t) 2^{n/2} \phi(2^n t - m) dt \quad (5.53)$$

Porém, nem sempre é fácil realizar esse cálculo, pois a maioria das funções ϕ_{n+1} não possui solução analítica, e por isso esse cálculo é realizado numericamente (DAUBECHIES, 1992).

Uma ideia mais simples é partir de uma resolução refinada o suficiente para que os coeficientes de detalhes da escala inferior sejam pouco representativos. Logo, os

coeficientes da DWT podem ser obtidos pela simples amostragem da função original (BURRUS *et al.*, 1998). Dessa maneira:

$$c_{N,m} = \langle f(t), \phi_{N,m}(t) \rangle \approx f(t_m), \quad m = 0, \dots, 2^N - 1 \quad (5.54)$$

sendo N a escala de maior resolução. Normalmente a escala mais refinada é $N = \log_2(n_s)$ em que n_s é o número de elementos de $f(t)$.

A obtenção dos coeficientes de detalhes é importante para detectar os principais aspectos do conjunto de dados e antes da etapa de síntese pode-se eliminar ou alterar esses coeficientes, de acordo com a necessidade. Esse procedimento é explicado no próximo Tópico 5.5.

O Exemplo 5.2 mostra como obter a DWT Haar de um vetor discreto $f(t_i)$, $i = 1, \dots, n_s$, com $n_s = 8$.

Exemplo 5.2. Exemplo Deduzido da DWT Haar

O objetivo desse exemplo é mostrar como se calcula a DWT da função $f(t_i)$, $i = 1, \dots, n_s$ usando a função wavelet Haar. Considere que:

$$f(t_i)_{i=1, \dots, n_s} = [1, 0, -3, 2, 1, 0, 1, 2] \quad (5.55)$$

Os coeficientes de filtro da função Haar (de acordo com o exemplo 5.1) são:

$$\ell_f(0) = 1/\sqrt{2}, \quad \ell_f(1) = 1/\sqrt{2}, \quad h_f(0) = 1/\sqrt{2} \quad \text{e} \quad h_f(1) = -1/\sqrt{2}.$$

O primeiro passo é calcular a escala mais refinada:

$$N = \log_2(n_s) = \log_2(8) = 3 \quad (5.56)$$

Logo, de acordo com a Equação (5.54):

$$c_{N,m} \cong f(t_m), \text{ resultando em:}$$

$$c_{3,m=0..7} = [1, 0, -3, 2, 1, 0, 1, 2] \quad (5.57)$$

Como, de acordo com a Equação (5.49):

$$c_{n,m} = \sum_i \ell_f(i - 2m) c_{n+1,i}, \text{ então:}$$

$$c_{2,0} = \ell_f(0)c_{3,0} + \ell_f(1)c_{3,1} = [(1/\sqrt{2}) \cdot 1] + [(1/\sqrt{2}) \cdot 0] = 1/\sqrt{2} \quad (5.58)$$

$$c_{2,1} = \ell_f(0)c_{3,2} + \ell_f(1)c_{3,3} = [(1/\sqrt{2}) \cdot -3] + [(1/\sqrt{2}) \cdot 2] = -1/\sqrt{2} \quad (5.59)$$

$$c_{2,2} = \ell_f(0)c_{3,4} + \ell_f(1)c_{3,5} = [(1/\sqrt{2}) \cdot 1] + [(1/\sqrt{2}) \cdot 0] = 1/\sqrt{2} \quad (5.60)$$

$$c_{2,3} = \ell_f(0)c_{3,5} + \ell_f(1)c_{3,6} = [(1/\sqrt{2}) \cdot 1] + [(1/\sqrt{2}) \cdot 2] = 3/\sqrt{2} \quad (5.61)$$

De acordo com a Equação 5.49:

$$d_{n,m} = \sum_i h_f(i-2m)c_{n+1,i}, \text{ então:}$$

$$d_{2,0} = h_f(0)c_{3,0} + h_f(1)c_{3,1} = [(1/\sqrt{2}) \cdot 1] + [(-1/\sqrt{2}) \cdot 0] = 1/\sqrt{2} \quad (5.62)$$

$$d_{2,1} = h_f(0)c_{3,2} + h_f(1)c_{3,3} = [(1/\sqrt{2}) \cdot -3] + [(-1/\sqrt{2}) \cdot 2] = -5/\sqrt{2} \quad (5.63)$$

$$d_{2,2} = h_f(0)c_{3,4} + h_f(1)c_{3,5} = [(1/\sqrt{2}) \cdot 1] + [(-1/\sqrt{2}) \cdot 0] = 1/\sqrt{2} \quad (5.64)$$

$$d_{2,3} = h_f(0)c_{3,5} + h_f(1)c_{3,6} = [(1/\sqrt{2}) \cdot 1] + [(-1/\sqrt{2}) \cdot 2] = -1/\sqrt{2} \quad (5.65)$$

Obtêm-se então os coeficientes de aproximação e detalhes no nível de resolução $n = 2$.

Essa operação pode ser representada matricialmente por:

$$\begin{bmatrix} c_{n,0} \\ c_{n,1} \\ \vdots \\ c_{n,2^n-1} \\ d_{n,0} \\ d_{n,1} \\ \vdots \\ d_{n,2^n-1} \end{bmatrix} = \begin{bmatrix} \ell_f(0) & \ell_f(1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ell_f(0) & \ell_f(1) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \ell_f(0) & \ell_f(1) \\ h_f(0) & h_f(1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_f(0) & h_f(1) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & h_f(0) & h_f(1) \end{bmatrix} \cdot \begin{bmatrix} c_{n+1,0} \\ c_{n+1,1} \\ c_{n+1,2} \\ c_{n+1,3} \\ \vdots \\ c_{n,2^n-2} \\ c_{n,2^n-1} \end{bmatrix} \quad (5.66)$$

Aplica-se o mesmo procedimento para obter os coeficientes de resolução mais baixa $n = 1$ a partir de $n = 2$:

$$\begin{bmatrix} c_{1,0} \\ c_{1,1} \\ d_{1,0} \\ d_{1,1} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} & 0 & 0 \\ 0 & 0 & 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 1/\sqrt{2} \\ -1/\sqrt{2} \\ 1/\sqrt{2} \\ 3/\sqrt{2} \end{bmatrix} = \begin{bmatrix} 0 \\ 2 \\ 1 \\ -1 \end{bmatrix} \quad (5.67)$$

E para os coeficientes de resolução mais baixa $n = 0$ a partir de $n = 1$:

$$\begin{bmatrix} c_{0,0} \\ d_{0,0} \end{bmatrix} = \begin{bmatrix} 1/\sqrt{2} & 1/\sqrt{2} \\ 1/\sqrt{2} & -1/\sqrt{2} \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 2 \end{bmatrix} = \begin{bmatrix} \sqrt{2} \\ -\sqrt{2} \end{bmatrix} \quad (5.68)$$

¶No processo inverso, ou seja, a etapa de síntese é obtida pelo inverso da operação apresentada.

Como:

$$f(t) = \sum_m c_{n_0,m} \phi_{n_0,m}(t) + \sum_{n \geq n_0} \sum_m d_{n,m} \psi_{n,m}(t) \quad (5.69)$$

é possível representar a função pelos coeficientes obtidos.

Obtendo-se os coeficientes, é possível analisar as características de $f(t_i)_{i=1,\dots,n_s}$

no espaço wavelets e então fazer a transformada inversa:

$$\begin{bmatrix} c_{n+1,0} \\ c_{n+1,1} \\ c_{n+1,2} \\ c_{n+1,3} \\ \vdots \\ c_{n,2^n-2} \\ c_{n,2^n-1} \end{bmatrix} = \begin{bmatrix} \ell_f(0) & \ell_f(1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & \ell_f(0) & \ell_f(1) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \ell_f(0) & \ell_f(1) \\ h_f(0) & h_f(1) & 0 & 0 & \cdots & 0 & 0 \\ 0 & 0 & h_f(0) & h_f(1) & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & 0 & 0 \\ 0 & 0 & 0 & 0 & \cdots & h_f(0) & h_f(1) \end{bmatrix}^{-1} \cdot \begin{bmatrix} c_{n,0} \\ c_{n,1} \\ \vdots \\ c_{n,2^n-1} \\ d_{n,0} \\ d_{n,1} \\ \vdots \\ d_{n,2^n-1} \end{bmatrix} \quad (5.70)$$

5.4 Família de Wavelets

As funções wavelets diferem substancialmente entre si, em características tais como ordem dos filtros de decomposição, existência ou não de forma analítica, tamanho de suporte, momentos nulos, ortogonalidade, possibilidade de implementação do algoritmo de Mallat, etc. Todas estas distinções obviamente fazem com que cada função seja mais ou menos adequada para um determinado tipo de análise.

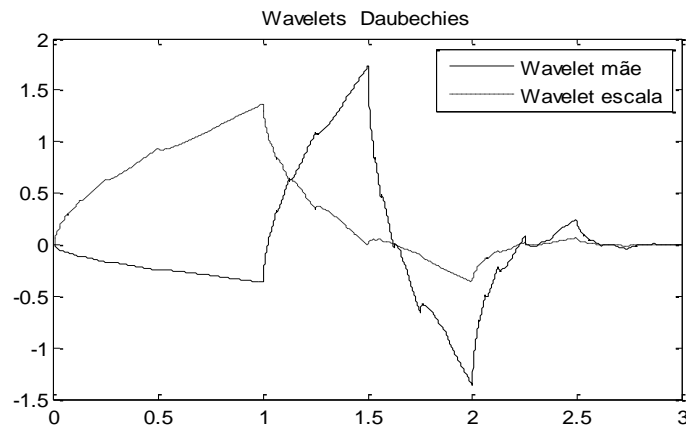
Na maioria dos casos, as funções wavelets não possuem expressão analítica, sendo necessários métodos numéricos para seu cômputo. No entanto, o algoritmo DWT

de Mallat, apresentado anteriormente, não é limitado por esse aspecto, sendo necessário apenas o conhecimento dos filtros das wavelets, conforme exemplificado no Tópico anterior.

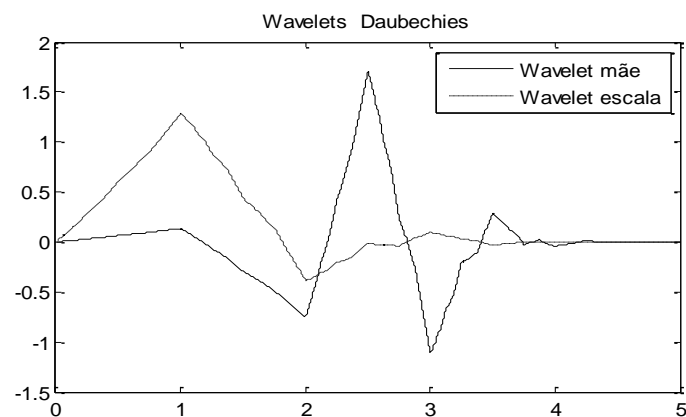
As condições necessárias e suficientes para a construção das wavelets envolvem vários aspectos como mencionados no Tópico 5.2. Uma descrição detalhada desse assunto pode ser consultada em BURRUS *et al.* (1998).

As wavelets Daubechies são bastante mencionadas na literatura e largamente utilizadas para processamento de imagens. Essa família de funções está dividida de acordo com o número de seus momentos nulos N , Figura 5.14.

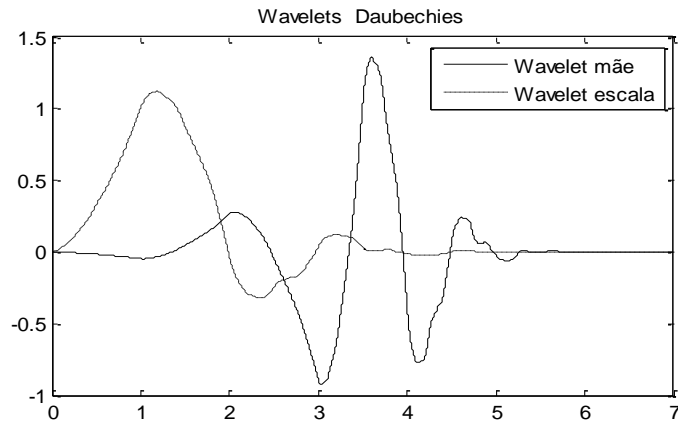
Na Figura 5.14 visualiza-se que quanto maior o número de momentos nulos, mais comprimida e suave é a função wavelet Daubechies.



(a) Daubechies 2.



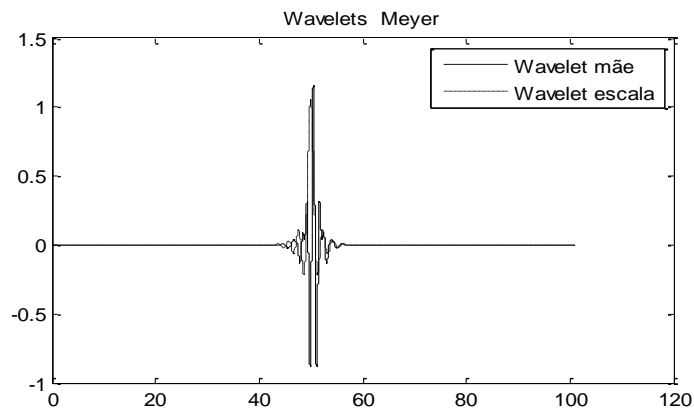
(b) Daubechies 3.



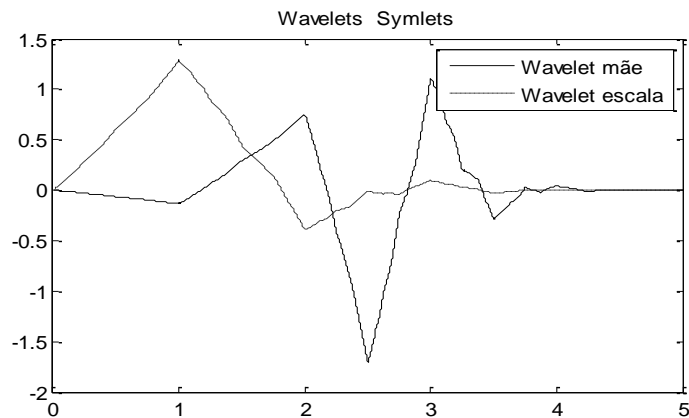
(c) Daubechies 4.

Figura 5.14. Família de funções wavelets Daubechies.

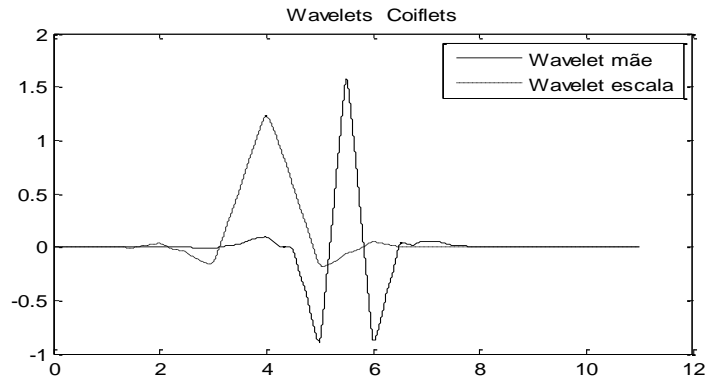
Outras funções bastante difundidas são as wavelets biortogonais, Meyer, Coiflets, Symlets, Morlet, Gaussiana, Shannon, etc. (DAUBECHIES, 1992). A Figura 5.15 ilustra algumas wavelets (wavelet mãe e função escala).



(a) Meyer.



(b) Symlets N=2



(c) Coiflets N=2

Figura 5.15. Ilustração de outras funções wavelets.

Exemplo 5.3. Análise de um sinal (usando pacote wavelets MATLAB 7.6[©]).

Considerando um sinal discreto y de 100 pontos, com as seguintes características:

$$y(x) = \begin{cases} 1, & x = 0 \\ \frac{\text{sen}(\pi x)}{\pi x}, & x \neq 0 \end{cases} \quad x \in [-5, 5] \quad (5.71)$$

A forma de $y(x)$ é mostrada na Figura 5.16.

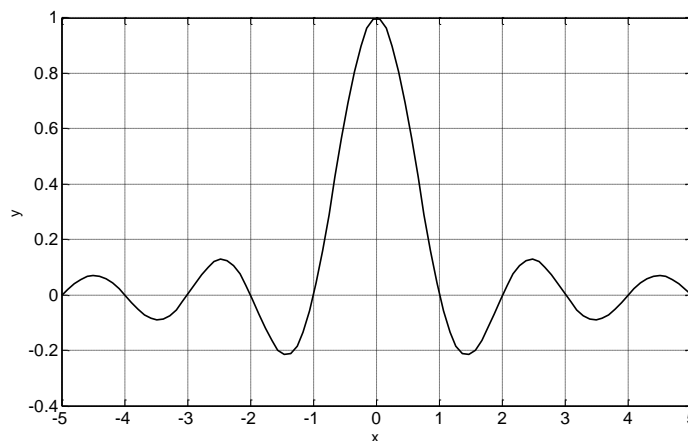


Figura 5.16. Função do problema-exemplo 5.3.

Decompõe-se $y(x)$ usando uma função wavelet Haar em três níveis de resolução, conforme mostrado na Figura 5.17.

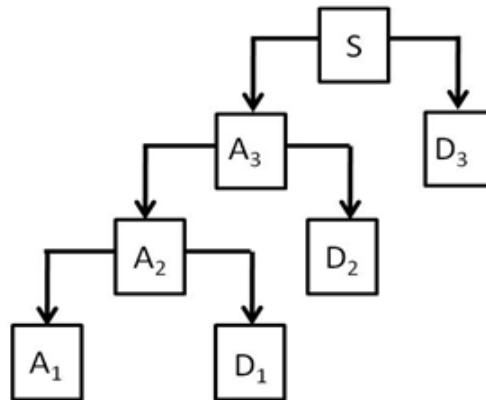


Figura 5.17. Decomposição de $y(x)$ em três níveis de resolução.

Na estrutura da Figura 5.17 os ramos da esquerda representam as aproximações enquanto os ramos da direita são os detalhes.

Na Figura 5.18 é possível observar que o sinal decomposto em três níveis, no primeiro há maior riqueza de detalhes, enquanto que no último nível o nível de detalhes é menor. É possível também analisar os locais em que os coeficientes de detalhes são mais expressivos (frequência elevada).

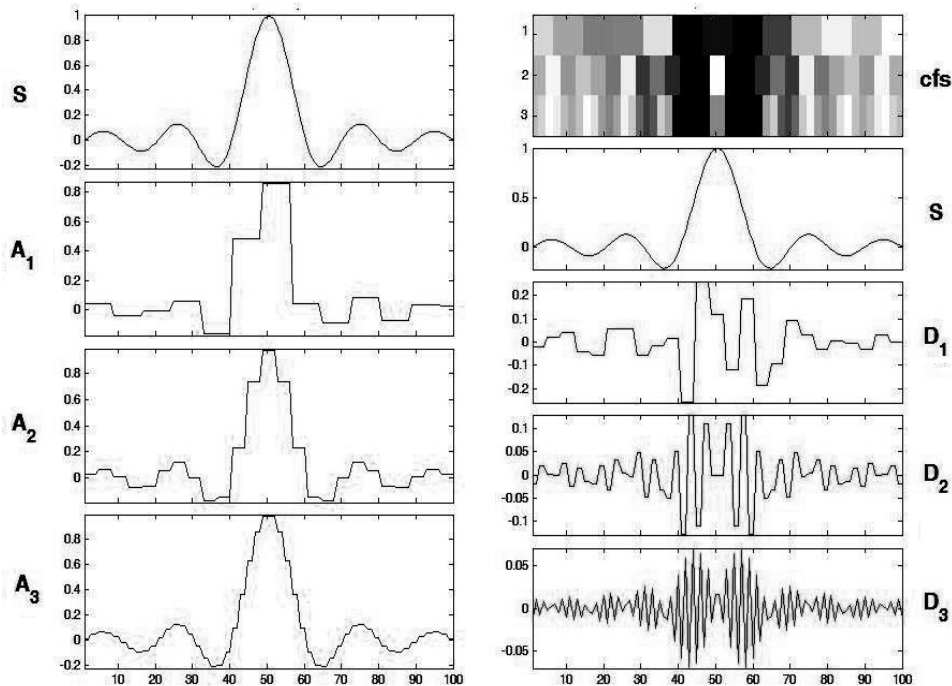


Figura 5.18. Gráficos das aproximações e detalhes do sinal S.

Na Figura 5.18, o primeiro gráfico da coluna da direita mostra o diagrama de coeficientes wavelets em cada nível de detalhes. Os pontos mais escuros representam os coeficientes mais elevados, enquanto os mais claros correspondem aos de menores valores. É possível observar que para este exemplo, os coeficientes mais expressivos estão na região de maior descontinuidade do sinal.

A vantagem de se decompor o sinal em diferentes níveis é que há diferentes maneiras de reconstruir o mesmo. Por exemplo, se retirar todos os detalhes, a reconstrução será representada apenas pela aproximação A_1 . Outra maneira é selecionar somente alguns coeficientes de detalhes, conforme discutido no próximo tópico.

5.5. Introdução ao Procedimento *Thresholding*

A ideia original da remoção de ruído de sinais, envolvendo wavelets, surgiu na década de 1990 a partir dos estudos de DONOHO (1992, 1993, 1995) e DONOHO e JONHSTONE (1994, 1995). O objetivo desses autores foi obter um procedimento de reconstrução seletiva do sinal, de modo a eliminar o ruído presente.

O maior desafio desses métodos é estimar a quantidade de ruído presente em um conjunto de dados. Para isso, os autores propuseram um critério baseado na estimação adaptativa do ruído (ver também MORETTIN, 1999; JANSEN, 2000 e MISITI *et al.*, 2007), que se baseia nos seguintes passos: **(i)** transformação dos dados para o domínio wavelets aplicando a **DWT**; **(ii)** eliminação dos coeficientes menos representativos (de acordo com um limiar ou *threshold* calculado); **(iii)** transformada inversa dos dados, **iDWT**.

Os passos **(i)** e **(iii)** foram tratados nos tópicos anteriores. Já este tópico aborda os aspectos do tratamento dos coeficientes de detalhes wavelets, passo **(ii)**. É unânime entre diversos pesquisadores da área que esta etapa é primordial para o sucesso da aplicação das wavelets, pois é na verdade o que justifica a aplicação da ferramenta. Em particular, o estudo desses critérios aplicados à análise wavelets de dados resultantes de processos de discretização ainda não foi realizado.

Supondo que a um conjunto de pontos discretos, $\mathbf{z} = \{z_1, z_2, \dots, z_{n_s}\}$ são incorporados ruídos $\boldsymbol{\eta} = \{\eta_1, \eta_2, \dots, \eta_{n_s}\}$ ¹². Ou seja,

$$\mathbf{z} = \mathbf{s} + \boldsymbol{\eta} \quad (5.72)$$

em que $\mathbf{s} = \{s_1, s_2, \dots, s_{n_s}\}$ é o sinal sem ruído. Supondo-se que o vetor de ruído tem distribuição normal $\boldsymbol{\eta} \sim N(0, \sigma^2)$ com desvio padrão σ , a **DWT** de \mathbf{z} pode ser escrita como:

$$\mathbf{DWT}(\mathbf{z}) = \mathbf{DWT}(\mathbf{s}) + \mathbf{DWT}(\boldsymbol{\eta}) \quad (5.73)$$

em que $\mathbf{DWT}(\mathbf{z}) = \mathbf{d}$, $\mathbf{DWT}(\mathbf{s}) = \boldsymbol{\theta}$, e $\mathbf{DWT}(\boldsymbol{\eta}) = \mathbf{w}$. Logo:

$$\mathbf{d} = \boldsymbol{\theta} + \mathbf{w} \quad (5.74)$$

sendo $\boldsymbol{\theta}$ o vetor de detalhes wavelets e \mathbf{w} o resíduo.

O procedimento *thresholding* pode ser escrito como:

$$\mathbf{d}_\delta = \mathbf{Thr}(\mathbf{d}, \delta) \cdot \mathbf{d} \quad (5.75)$$

em que \mathbf{d}_δ são os (*thesholded*) coeficientes filtrados e $\mathbf{Thr}(\cdot)$ é uma regra *thresholding* que depende do valor de *threshold* δ , em que $\delta \in \mathbb{R}$. Logo, a regra *thresholding* pode ser escrita como:

$$\mathbf{Thr}(d_{n,m}, \delta) = d_{n,m} - \mathbf{sgn}(d_{n,m}) \quad (5.76)$$

em que $\mathbf{sgn}(\cdot)$ é uma função suave definida como:

$$\mathbf{sgn}(d_{n,m}) = \begin{cases} -1 & \text{se } d_{n,m} < -\delta \\ 0 & \text{se } d_{n,m} \in [-\delta, \delta] \\ 1 & \text{se } d_{n,m} > \delta \end{cases} \quad (5.77)$$

Depois do procedimento de *thresholding* calcula-se a transformada inversa:

$$\mathbf{s}_\delta = \mathbf{iDWT}(\mathbf{d}_\delta) \quad (5.78)$$

Em linhas gerais, o *theshold* “ótimo” deve ser calculado de maneira a minimizar o desvio médio quadrático entre $\boldsymbol{\theta}$ e \mathbf{d}_δ , ou seja:

$$F_\delta = \frac{1}{n_s} \sum_{i=1}^{n_s} (d_{\delta_i} - \theta_i)^2 \quad (5.79)$$

O problema é que $\boldsymbol{\theta}$ é desconhecido. Logo, o F_δ deve ser estimado:

¹² Define-se n_s propositalmente para futuras aplicações de *thresholding* em vetores de n_s estágios (perfis de controle).

$$R(\mathbf{d}_\delta, \boldsymbol{\theta}) = \mathbf{E}(F_\delta) = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{E}(d_{\delta_i} - \theta_i)^2 \quad (5.80)$$

em que R é definido como a função risco e \mathbf{E} é a esperança de F_δ .

Como a **DWT** é ortogonal, $R(\mathbf{d}_\delta, \boldsymbol{\theta}) = R(\mathbf{s}_\delta, \mathbf{s})$.

5.6. Em Busca do *Threshold Ideal* (MISITI *et al.*, 2007)

5.6.1. Estratégia I

Utilizando a formulação apresentada no tópico anterior, esboça-se neste tópico a estratégia de *thresholding* ideal. Por conveniência utiliza-se a mesma notação utilizada no tópico anterior.

O objetivo é estimar \mathbf{s} (até então desconhecida) de maneira a minimizar o risco, definido como o desvio médio quadrático entre a função \mathbf{s} , e sua estimativa \mathbf{s}_δ , lembrando que $\mathbf{s}_\delta = \mathbf{iDWT}(\mathbf{d}_\delta)$. Logo:

$$R(\mathbf{s}_\delta, \mathbf{s}) = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{E}(s_{\delta_i} - s_i)^2 \quad (5.81)$$

Uma ideia de estimação de uma função \mathbf{s} consiste em propor estimadores de acordo com a seguinte expansão em bases wavelets (conforme definido nos tópicos anteriores):

$$\mathbf{s} = \sum_n c_{n_0, m} \varphi_{n_0, m}(t) + \sum_{n \geq n_0} \sum_m \alpha_{n, m} d_{n, m} \psi_{n, m}(t) \quad (5.82)$$

em que aqui $d_{\delta_{n, m}} = \alpha_{n, m} d_{n, m}$, sendo $\alpha_{n, m} \in \mathbb{R}$ um fator de ponderação.

Escolhe-se $\alpha_{n, m}$ a fim de minimizar $R(\mathbf{s}_\delta, \mathbf{s})$. Considerando uma escala n fixa, reescreve-se o risco como:

$$R(\mathbf{s}_\delta, \mathbf{s}) = \sum_m \mathbf{E}(\alpha_{n, m} d_{n, m} \psi_{n, m} - \theta_{n, m} \psi_{n, m})^2 = \sum_m \mathbf{E}(\alpha_{n, m} d_{n, m} - \theta_{n, m})^2 \quad (5.83)$$

Sabendo-se que, $\mathbf{d} = \boldsymbol{\theta} + \mathbf{w}$, a Equação (5.83) é escrita como:

$$R(\mathbf{s}_\delta, \mathbf{s}) = \sum_m \mathbf{E}(\alpha_{n, m} (\theta_{n, m} + w_{n, m}) - \theta_{n, m})^2 \quad (5.84)$$

Simplificando a notação, substituímos os índices m, n por i :

$$R(\mathbf{s}_\delta, \mathbf{s}) = \sum_i \mathbf{E}(\alpha_i (\theta_i + w_i) - \theta_i)^2 \quad (5.85)$$

Logo:

$$\sum_i \mathbf{E}(\alpha_i (\theta_i + w_i) - \theta_i)^2 = \sum_i \left(\theta_i^2 (1 - \alpha_i)^2 + \alpha_i^2 \mathbf{E}w_i^2 - 2\theta_i (1 - \alpha_i) \alpha_i \mathbf{E}w_i \right) \quad (5.86)$$

Considerando que $\mathbf{E}w_i = 0$ (média zero) e $\mathbf{E}w_i^2 = \sigma^2$ (variância σ^2), a função risco é definida como:

$$R_{ideal}(\mathbf{s}_\delta, \mathbf{s}) = \sum_i \mathbf{E}(\alpha_i (\theta_i + w_i) - \theta_i)^2 = \sum_i \theta_i^2 (1 - \alpha_i)^2 + \sigma^2 \sum_i \alpha_i^2 \quad (5.87)$$

Nota-se que a minimização de $R_{ideal}(\mathbf{s}_\delta, \mathbf{s})$ implica em:

$$\frac{\partial}{\partial \alpha_i} R_{ideal}(\mathbf{s}_\delta, \mathbf{s}) = 0 \quad (5.88)$$

O que resulta no valor ótimo de α_i :

$$\alpha_i = \theta_i^2 / (\theta_i^2 + \sigma^2) \quad (5.89)$$

Analisando a estrutura do valor ótimo de α_i , percebe-se sua dependência com θ_i que são os coeficientes da função desconhecida \mathbf{s} . Logo, é preciso conhecer os valores de θ_i para implementação do método.

5.6.2. Estratégia II

Uma segunda ideia para a busca do *threshold* ótimo segue abaixo:

Definindo $I = \{i \in \{1, 2, \dots, n_s\} \mid \alpha_i = 1\}$ o conjunto de índices selecionados (quando $\theta_i \gg \sigma$), é possível reescrever a Equação (5.87) como:

$$R_{ideal}(\mathbf{d}_\delta, \boldsymbol{\theta}) = \sum_{i \notin I} \theta_i^2 + \sum_{i \in I} \sigma^2 = \sum_{i \notin I} \theta_i^2 + \underbrace{\text{card}(I) \sigma^2}_{\theta_i \gg \sigma} \quad (5.90)$$

em que $\text{card}(I)$ é o número de elementos do vetor I .

A Equação (5.90) indica que a seleção de i em I custa σ^2 e a exclusão custa θ_i^2 . Portanto, $R(\mathbf{s}_\delta, \mathbf{s})$ é mínimo apenas se os índices $\|\theta_i\| \geq \sigma$ aparecerem em I . Isso significa que há uma estratégia ótima (ideal) de seleção dos coeficientes, caso se conheça os valores de θ_i e σ . A política de *thresholding* de uma variável θ_i , nesse caso, é preservar somente $\|\theta_i\| \geq \sigma$.

Existem diversas estratégias para a escolha do *threshold* ótimo (como *Sureshtink*, *Visushrink*, *Minimaxi*, *Cross Validation*, *Bayeshirink* etc.). Essas estratégias são constantemente utilizadas para remoção de ruídos de imagens e compressão de dados (DANGETI, 2003). Nos tópicos a seguir, explicam-se as principais estratégias apresentadas na literatura: *Visushrink* e *Sureshrink*.

5.7. Critérios *Thresholding*

5.7.1. *Visushrink*

DONOHO (1992) desenvolveu uma metodologia para estimar um nível de *threshold* de um sinal a partir do modelo de ruído branco. Os autores apresentaram algumas regras para seleção do *threshold*, entre elas a mais popular chamada "regra do *threshold* universal" ou *Visushrink*. O termo é em referência às boas propriedades visuais da reconstrução obtida pelo *thresholding* dos coeficientes wavelets resultantes da DWT aplicada ao processamento de imagens. Essa metodologia parte da ideia de seleção de certos coeficientes prospectivos, o que seria o papel do parâmetro α na Equação (5.89).

A estratégia adotada por DONOHO (1992) para estimar o valor do *threshold*, independentemente do conhecimento de θ_i foi utilizar diretamente o modelo do ruído branco, apresentado no tópico anterior:

$$\mathbf{d} = \boldsymbol{\theta} + \mathbf{w} \quad (5.74)$$

Nesse caso, o *threshold* é definido como:

$$\delta^{VS} = \sigma \sqrt{2 \ln(n_s)} \quad (5.91)$$

em que n_s é o número de elementos da amostra, δ^{VS} é o *threshold* (*Visushrink*) e σ é o desvio padrão do ruído.

Em JANSEN (2001) é possível verificar a dedução de que δ^{VS} tende minimizar $R(\mathbf{s}_\delta, \mathbf{s})$ quando $n_s \rightarrow \infty$.

O valor de $\delta^{VS} = \sigma \sqrt{2 \ln(n_s)}$ vem da Teoria de Valores Extremos (LEADBETTER *et al.*, 1983). Essa teoria especifica possíveis formas para o limite da distribuição de máximos de sequências.

Ao definir $\|w_{\max}\| = \max(w_1, \dots, w_{n_s})$, a convergência da probabilidade de $\|w_{\max}\| \leq \delta$ é definida como:

$$P\{\|w_{\max}\| \leq \delta\} = P\{w_1 \leq \delta, w_2 \leq \delta, \dots, w_{n_s} \leq \delta\} = F(\delta) \quad (5.92)$$

em que F representa a distribuição $P\{\|w_{\max}\| \leq \delta\}$. O interesse dessa teoria é estudar o comportamento de F para valores elevados de n_s . Fisher e Tippett descobriram que há três famílias para F que são classificadas como distribuições de valores extremos. Essa classificação independe da própria distribuição de \mathbf{d} , sendo diferenciadas apenas por constantes de normalização definidas abaixo:

Segundo (LEADBETTER *et al.*, 1983), para o caso de distribuição $N(0, \sigma)$:

$$P\{a_{n_s} (\|w_{\max}\| - b_{n_s}) \leq \delta\} = e^{(-e^{-\delta})} \quad (5.93)$$

As constantes são escritas como:

$$a_{n_s} = \sqrt{2 \ln(n_s)} \quad (5.94)$$

e:

$$b_{n_s} = (2 \ln(n_s))^{1/2} - \frac{1}{2} \left[2 (\ln(n_s))^{-1/2} s_{n_s} \right] \quad (5.95)$$

e

$$s_{n_s} = \ln[\ln(n_s)] + \ln(4\pi) \quad (5.96)$$

O que implica em:

$$P_{n_s \rightarrow \infty} (\|w_{\max}\| \leq \sigma \sqrt{2 \ln(n_s)}) = 1 \quad (5.97)$$

Isso significa que o *threshold* $\sigma \sqrt{2 \ln(n_s)}$ tende a remover todos os coeficientes com $N(0, \sigma)$. Uma discussão mais detalhada sobre a teoria de valores extremos pode ser conferida em KOTZ e NADARAJAH (2000).

A estratégia *Visushrink* carrega o fardo de eliminar muitos coeficientes de detalhes em níveis de resolução elevados, ou seja, um melhor desempenho poderia ser obtido se valores pequenos de *threshold* fossem utilizados em níveis de resolução maiores. Em outras palavras, dependendo dos dados a serem analisados, essa estratégia pode ser muito agressiva ao eliminar coeficientes de detalhes importantes para conservar as características principais de um sinal.

Note que o *threshold* $\delta^{VS} = \sigma\sqrt{2 \ln(n_s)}$ necessita do cálculo explícito de σ , que é o desvio padrão de possíveis ruídos presentes. Como esse valor é desconhecido, geralmente utiliza-se a mediana do desvio absoluto como estimador, ou seja:

$$\sigma \approx \kappa \mathbf{MAD} \quad (5.98)$$

em que **MAD** é a mediana do desvio absoluto, ou seja:

$$\mathbf{MAD} = \text{mediana}_i (\|w_i - \text{mediana}(\mathbf{w})\|) = \text{mediana}(\|\hat{\mathbf{w}}\|) \quad (5.99)$$

sendo $\kappa \approx 1,4826$. O valor de κ é deduzido abaixo:

Prova de que $\kappa \approx 1,4826$:

Nota-se que para distribuições normais:

$$P(\|\hat{\mathbf{d}}\| \leq \mathbf{MAD}) = 1/2 \quad (5.100)$$

ou

$$P\left(\left\|\frac{\hat{\mathbf{w}}}{\sigma}\right\| \leq \frac{\mathbf{MAD}}{\sigma}\right) = P\left(\|\hat{\mathbf{z}}\| \leq \frac{\mathbf{MAD}}{\sigma}\right) = 1/2 \quad (5.101)$$

Como $\Phi(\gamma) = P(\|\hat{\mathbf{z}}\| \leq \gamma) = c$ é a distribuição cumulativa de γ .

e também:

$$\Phi(\gamma) = P(a \leq \gamma \leq b) \quad (5.102)$$

Logo,

$$\Phi\left(\frac{\mathbf{MAD}}{\sigma}\right) - \Phi\left(-\frac{\mathbf{MAD}}{\sigma}\right) = 1/2 \quad (5.103)$$

$$\text{sendo que } \Phi\left(-\frac{\mathbf{MAD}}{\sigma}\right) = 1 - \Phi\left(\frac{\mathbf{MAD}}{\sigma}\right) \quad (5.104)$$

Logo:

$$\Phi\left(\frac{\mathbf{MAD}}{\sigma}\right) - \left\{1 - \Phi\left(\frac{\mathbf{MAD}}{\sigma}\right)\right\} = 1/2 \quad (5.105)$$

Consequentemente:

$$\Phi\left(\frac{\mathbf{MAD}}{\sigma}\right) = \frac{3}{4} \quad (5.106)$$

Como $\Phi(\gamma) = c$ e $\Phi^{-1}(c) = \gamma$, logo:

$$\frac{\mathbf{MAD}}{\sigma} = \Phi^{-1}\left(\frac{3}{4}\right) \quad (5.107)$$

Logo,

$$\kappa \approx \Phi^{-1}\left(\frac{3}{4}\right) = 1,4826 \quad (5.108)$$

Geralmente, estima-se $\sigma \approx \kappa \mathbf{MAD}$ a partir dos coeficientes do nível wavelets de maior resolução onde teoricamente as oscilações de maiores frequências são mais fáceis de serem detectadas.

5.7.2 Sureshrink

Uma estratégia de *thresholding* baseada no SURE (*Stein's Unbiased Risk Estimator*) – (STEIN, 1981) foi proposta por DONOHO e JOHNSTONE (1995) e é conhecida como *Sureshrink*. Esse método é uma combinação entre a estratégia *Visushrink*, apresentada no tópico anterior, e o estimador SURE. Os pesquisadores mostraram que o método SURE origina um estimador não tendencioso e que minimiza uma função de risco para um valor de *threshold* particular; ou seja, para um dado conjunto de dados, o *threshold* é aquele que apresenta o menor risco.

Considere novamente o risco definido pela Equação (5.80):

$$R(\mathbf{d}_\delta, \boldsymbol{\theta}) = \frac{1}{n_s} \sum_{i=1}^{n_s} \mathbf{E}\left(d_{\delta_i} - \theta_i\right)^2 \quad (5.80)$$

STEIN (1981) propôs uma estimativa não tendenciosa do desvio médio quadrático de um estimador de $\boldsymbol{\theta}$, ou seja, o SURE é uma maneira de avaliar a acurácia de um determinado estimador. Para ser mais específico, assume-se que um estimador da forma:

$$\hat{\boldsymbol{\theta}} = \mathbf{d}_\delta = \mathbf{d} + \omega(\mathbf{d}) \quad (5.109)$$

em que ω é uma função diferenciável.

O SURE é avaliado como:

$$\mathbf{SURE}(\hat{\boldsymbol{\theta}}) = \mathbf{E}(\|\mathbf{d} + \boldsymbol{\omega}(\mathbf{d}) - \boldsymbol{\theta}\|) = \hat{\sigma}^2 + \mathbf{E}(\|\boldsymbol{\omega}(\mathbf{d})\|)^2 + 2 \sum_i \frac{\partial \boldsymbol{\omega}(d_i)}{\partial d_i} \quad (5.110)$$

DONOHO e JOHNSTONE (1995) mostraram que quando:

$$\mathbf{d}_\delta = \mathbf{Thr}(\mathbf{d}_\delta, \delta) \cdot \mathbf{d} \quad (5.75)$$

em que:

$$\mathbf{Thr}(d_{n,m}, \delta) = d_{n,m} - \delta \mathbf{sgn}(d_{n,m}) \quad (5.76)$$

e $\mathbf{sgn}(\cdot)$ é uma função suave definida como:

$$\mathbf{sgn}(d_{n,m}) = \begin{cases} -1 & \text{se } d_{n,m} < -\delta \\ 0 & \text{se } d_{n,m} \in [-\delta, \delta] \\ 1 & \text{se } d_{n,m} > \delta \end{cases} \quad (5.77)$$

o SURE pode ser escrito como:

$$\mathbf{SURE}(\mathbf{d}_\delta) = n_s - 2 \# \{i: \|d_i\| < \delta\} + \sum_{i=1}^{n_s} \text{mín}(|d_i|, \delta)^2 \quad (5.111)$$

em que $\#$ é uma função indicadora definida como:

$$\# \{x\} = \begin{cases} 1, & \text{se } x \text{ é verdadeiro} \\ 0, & \text{se } x \text{ é falso} \end{cases} \quad (5.112)$$

A estratégia *Sureshrink* se baseia então em escolher δ de maneira a minimizar o SURE. Esse cálculo é realizado em cada nível e decomposição wavelets, ou seja:

$$\delta^{SS} = \text{argmin}_{0 \leq \delta \leq \delta^{VS}} \mathbf{SURE} \quad (5.113)$$

A Equação (5.113) mostra que o *threshold* ótimo é resultado da minimização do SURE com δ entre 0 e δ^{VS} . Esse cálculo é adaptativo, ou seja, é realizado para cada nível de resolução wavelet separadamente. Geralmente, δ^{SS} é multiplicado pela estimativa da variância do ruído presente em cada nível de resolução. A estimativa da variância é realizada de acordo com o procedimento explicado no tópico anterior.

A Figura 4.8, ilustrada no Capítulo 4 exemplifica o cômputo do δ^{SS} para 4 níveis de resolução diferentes.

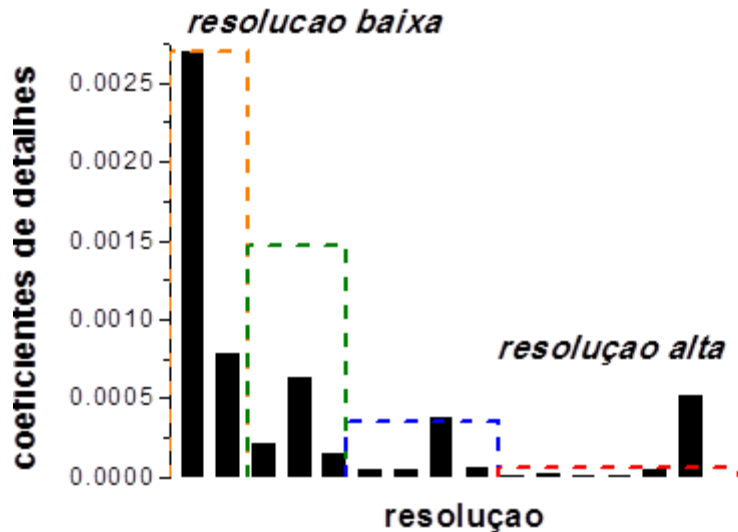


Figura 4.8. Ilustração: espectro de detalhes wavelets.

Note na Figura 4.8 que no critério adaptativo o valor de *threshold* varia com o nível de resolução. Em outras palavras, cada nível é avaliado independentemente.

Há diversos estudos sobre a política de *thresholding* para compressão de dados e remoção de ruídos de sinais. Dentre esses, pode-se citar: ABRAMOVICH e BENJAMINI (1996), HALL e PATIL (1996), AZZALINE *et al.* (2005), ROSAS-OREA *et al.* (2005), GHANBARI e KARAMI-MOLLAEI (2006), LUISIER *et al.* (2007), XU *et al.* (2009), BARTUSEK *et al.* (2011).

JANSEN (2000) comparou as estratégias *Visushrink* e *Sureshrink* aplicadas à compressão de imagens. Também foi introduzida uma nova estratégia definida como *Cross Validation* Generalizada. O pesquisador mostrou que a *General Cross Validation* a *Sureshrink* são bons estimadores do risco, sendo que a *General Cross Validation* possui a vantagem de não precisar estimar a variância do ruído desconhecido.

WIELAND (2009) comparou os três métodos para remoção de ruídos sonoros. Foi concluído que para sinais pouco ruidosos, em geral, o *Visushrink* obteve melhores desempenhos. Já para sinais muito ruidosos, o *Sureshrink* mostrou-se mais eficaz.

BARTUSEK *et al.* (2011) testou oito tipos de wavelets para remoção de ruído de imagens. Nesse estudo foram combinadas cada wavelet com a política de *hard* e *soft thresholds*. Os resultados indicaram que as wavelets Haar são mais indicadas para remoção de ruído de imagens simples (com poucos detalhes). As wavelets Daubechies e

bi-ortogonais mostraram-se úteis para imagens mais complexas, com elevado nível de detalhes.

É importante notar que as estratégias thresholding aqui apresentadas e os algoritmos baseados em discretização wavelets aplicados à solução de DAOPs, conforme introduzido no Capítulo 4 (BINDER *et al.*, 2000; SCHLEGEL, 2004; SCHLEGEL *et al.*, 2005; ASSASSA e MARQUARDT. 2004), são temas intimamente intrincados. Sob esse prisma, nos próximos dois realizaremos um estudo mais criterioso sobre essas aplicações.

No exemplo a seguir, são aplicadas diferentes métricas de avaliação do *threshold* para o mesmo sinal estudado no exemplo 5.3.

Exemplo 5.4. Comparação entre os métodos *Visushrink* (UNIVERSAL) e *Sureshrink*.

Considerando um sinal discreto y de 100 pontos, com as seguintes características:

$$y(x) = \begin{cases} 1, & x = 0 \\ \frac{\text{sen}(\pi x)}{\pi x}, & x \neq 0 \end{cases} \quad x \in [-5, 5] \quad (5.114)$$

A forma de $y(x)$ é mostrada na Figura 5.19.

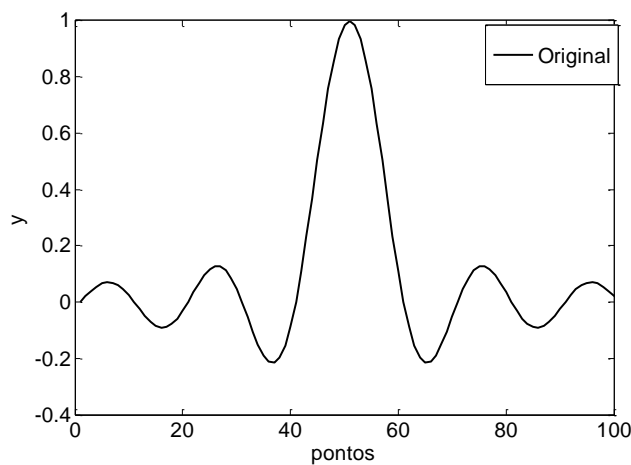


Figura 5.19. Função do problema-exemplo 6.1.

sendo que há 100 pontos discretos entre o intervalo de -5 a 5.

Decompõe-se $y(x)$ usando uma função wavelet Haar até o máximo nível de resolução 6. O espectro de detalhes dessa decomposição é mostrado na Figura 5.20.

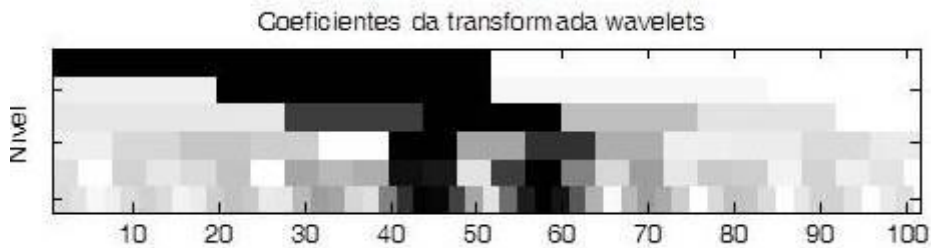


Figura 5.20. Espectro de detalhes do problema-exemplo 5.4.

Na Figura 5.20, no eixo vertical tem-se os seis níveis de decomposição em ordem crescente de cima para baixo, enquanto que no eixo horizontal tem-se o número de coeficientes de detalhes. Nota-se a concentração de detalhes (retângulos escuros) na região central do sinal, que é onde há maior nível de descontinuidade.

Na Figura 5.22 ilustra-se o resultado do *thresholding* utilizando o *Visushrink*: Ao comparar a Figura 5.23 à Figura 5.21 nota-se que o método *Sureshrink* é mais conservador que o método *Visushrink*, preservando maior quantidade de coeficientes de detalhes em regiões próximas às descontinuidades. Ao aplicar a transformada wavelets inversa, compara-se ambos os sinais comprimidos na Figura 5.24.

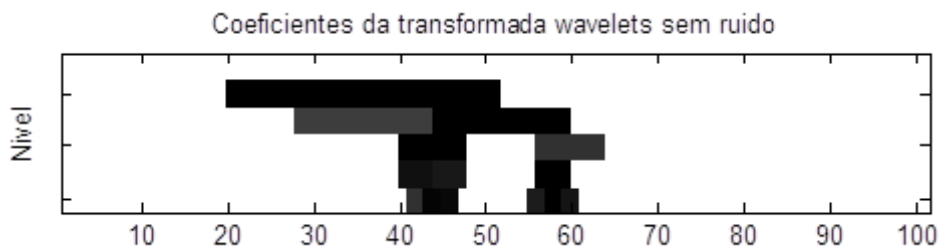


Figura 5.21. Espectro de detalhes do problema-exemplo 5.4 depois da decomposição filtragem usando *Visushrink*.

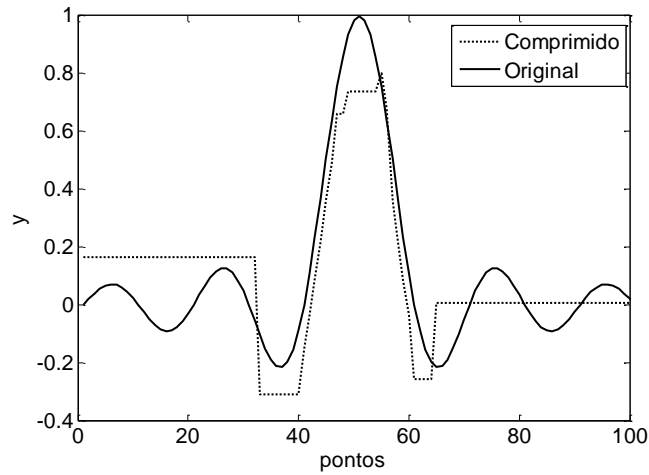


Figura 5.22. Comparação entre o sinal original e o sinal comprimido.

Comparou-se também o mesmo procedimento adotando o método *Sureshrink*:

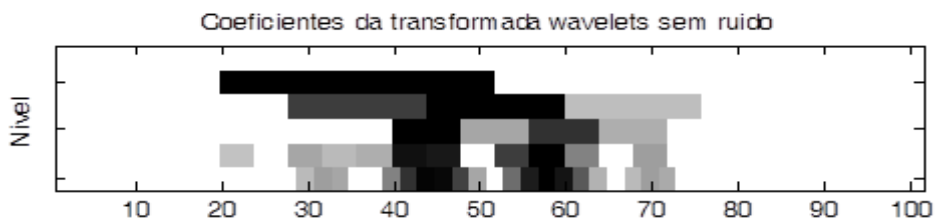


Figura 5.23. Espectro de detalhes do problema-exemplo 5.4 depois da decomposição filtragem usando *Sureshrink*.

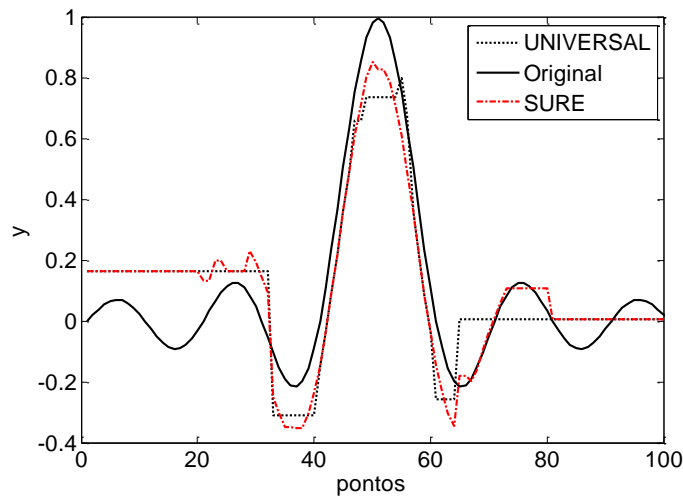


Figura 5.24. Comparação entre o sinal original e o sinal comprimido (problema-exemplo 5.4).

5.8 Aplicações de Wavelets em Engenharia: da Solução de Sistemas de Equações à Otimização Dinâmica

Este tópico expõe algumas aplicações de wavelets para soluções de sistemas de equações diferenciais parciais, PDEs e ODEs com duplo valor de contorno. Considera-se importante citar alguns trabalhos dessa área, pois a partir dessas aplicações que as wavelets passaram a ser exploradas como alternativa numérica de soluções de equações diferenciais (tanto como funções-base como ferramenta para tratar a política de discretização) e também problemas de otimização e controle ótimo.

BEYLKIN (1993) introduziram as wavelets como ferramenta numérica para representar operadores (polinomiais, exponenciais, inversa, raiz quadrada etc.). O autor descreve como representar operadores diferenciais e integrais em bases wavelets. A ideia é representar tais operadores, que são escritos por matrizes densas, em sua forma numérica, por matrizes esparsas, resultando em algoritmos mais rápidos.

Há várias aplicações das wavelets para solução de sistemas de PDEs e ODEs. Segundo FRAZIER (1999), a aproximação numérica de algumas classes de PDEs por essa ferramenta resulta em duas propriedades desejáveis para o sistema resultante da discretização: esparsividade e baixo número de condicionamento das matrizes.

Nos anos 1990, com a evolução e amadurecimento das wavelets surgiram alguns trabalhos, citando os principais: BERTOLUZZA e NALDI (1994) e VON WATZDORF *et al.* (1996) cujo objetivo seria a aplicação dessa ferramenta para solucionar sistemas de PDEs.

Tal estratégia tem sido implementada como métodos de colocação, como sugerido por BERTOLUZZA e NALDI (1994) ou por métodos Galerkin como fez JAFFARD (1992). Há também uma terceira linha de pesquisa conhecida como interpolação wavelets como descrito em KARAMI *et al.* (2008) e KUMAR e MEHRA (2007).

JAFFARD (1992) utilizaram wavelets do tipo Meyer para discretizar problemas elípticos com condições de contorno de Neumann e Dirichlet (RICE e DO, 1994). O autor elaborou um procedimento que consiste em utilizar o método de Galerkin, com a utilização de bases wavelets Meyer. Segundo o autor, essa metodologia, além de

representar o problema de forma esparsa, gera automaticamente operadores bem condicionados. FRAZIER (1999) dedica um capítulo de seu livro ao método Galerkin-wavelets. Esse método foi também estudado por BRIESEN e MARQUARDT (2003), BINDAL *et al.* (2003) e OROPEZA (2007).

BERTOLUZZA e NALDI (1994) descreveram um método de colocação baseado em funções de autocorrelação das wavelets Daubechies para solucionar sistemas de PDEs. Tais funções foram sugeridas para facilitar o tratamento das condições de contorno e também tratamento de não linearidades. CRUZ *et al.* (2001), LIU *et al.* (2001), CHEN e HSIAO (1999) e LEPIK (2005) também utilizaram um procedimento semelhante.

Apesar do avanço das wavelets para solucionar sistemas de equações diferenciais, há autores que questionam a viabilidade do método, principalmente para o tratamento de não linearidades, tornando o problema desnecessariamente complicado quando tratado com expansão em série wavelets, além disso, os autores citam também dificuldades quanto o tratamento das condições de contorno dos problemas. Esse aspecto foi verificado em BINDER (2002) ao solucionar problemas de otimização dinâmica com aplicação de expansão wavelets, fato que o desencorajou a usar tal procedimento.

Outra maneira de explorar a característica das wavelets é utilizá-las somente para atualizar ou intervir na discretização de métodos como diferenças finitas. Logo, o procedimento consiste em utilizar as wavelets apenas como ferramenta de identificação de pontos de descontinuidade. Por exemplo, BLANCO (2002) considerou um esquema adaptativo para a solução numérica de PDEs. Em cada passo de tempo, a malha espacial e irregular é escolhida dinamicamente de maneira a apresentar mais pontos nas zonas onde o comportamento da solução seja brusco, e menos pontos onde seja suave. Para a detecção de tais regiões utilizou-se os coeficientes wavelets como indicadores de regularidade local.

Indubitavelmente, mesmo percorrendo caminhos diferentes, todos esses trabalhos do início da década de 1990 contribuíram de modo substancial para aplicações das wavelets para solução de problemas de cálculo variacional e otimização dinâmica.

Verifica-se entretanto duas abordagens diferentes para essa aplicabilidade: (i) aplicação para discretização total e (ii) aplicação para etapa complementar à discretização.

De acordo com esta revisão, CHEN e HSIAO (1997) foram os primeiros a explorar as funcionalidades das wavelets para solucionar problemas de cálculo variacional. Os autores utilizaram as funções Haar para estabelecer uma matriz operacional para o operador de integração, utilizando uma espécie de método de quadratura numérica para integração. Nesse âmbito, foi possível representar o problema no próprio domínio wavelets, calculando os coeficientes de detalhes ótimos. Esse procedimento também foi desenvolvido para resolver problemas de controle ótimo lineares por HSIAO e WANG (1999) e HSIAO (2004). DAI e COCHRAN Jr. (2009) usaram um método de colocação-wavelets Haar para solução de problemas de controle ótimo. Os autores expandiram tanto as variáveis de controle como as variáveis de estado em wavelets Haar, resultando em um sistema NLP de múltipla resolução. Mais recentemente, HAYA (2011) implementou com sucesso essa estratégia para solucionar problemas de controle ótimo com maior nível de dificuldade: com restrições de trajetória e múltiplas variáveis de controle.

BINDER (2002) resgatou a metodologia de HSIAO e WANG (1999) RIC, ao aproximar as funções por séries wavelets Haar. O seu objetivo seria resolver problemas de controle ótimo, já vislumbrando aplicação em tempo real, onde a velocidade de cálculo pode vir a se tornar uma limitação. No entanto, Binder concluiu que em sua metodologia que a discretização wavelets é computacionalmente mais custosa devido principalmente ao fato de que o cálculo derivadas dos resíduos das equações algébricas que representam as restrições do problema de otimização dinâmica ser muito oneroso computacionalmente e desnecessariamente complexo, comparando-se a aproximação por funções locais em único nível de resolução. O autor, diante dessa aparente dificuldade, absteve-se da ideia de representação wavelets de todo o sistema, aplicando somente sobre a variável de controle.

5.9 Resumo do Capítulo

Neste capítulo foram introduzidos os principais fundamentos das wavelets e propriedades multirresolução. Foi destacado que as wavelets, por serem localizadas e

serem bases ortogonais podem representar um conjunto de dados em múltiplas resoluções, possibilitando localizar regiões com transitoriedades, descontinuidades ou ruídos. Alguns exemplos básicos foram expostos com o objetivo de esclarecer o comportamento das wavelets. Viu-se que a partir de uma única função-escala e conhecimento dos coeficientes de filtro é possível calcular todos os coeficientes de detalhes da decomposição via algoritmo Mallat. Mostrou-se também que esses coeficientes de filtro podem ser interpretados como filtros passa-baixa passa-alta, e dessa maneira a DWT pode ser representado como Banco de filtros digitais.

O Tópico 5.7.evidenciou que as wavelets prestam-se bem para compressão de dados e remoção de ruídos. Há diversos estudos, oriundos da área de estatística, cuja proposta de compressão de dados se sustenta em estratégias para estimação do nível de ruído ou informações descartáveis que possam ser eliminadas de acordo com um determinado nível de *threshold*. Há algumas regras para o cálculo do nível de *threshold* já bastante consolidadas como o *Sureshrink* e o *Visushrink*. Essas ferramentas têm sido largamente utilizadas para tratamento de imagens, remoção de ruídos etc. A literatura vem mostrando que as regras utilizadas para tal fim são determinantes para a seleção dos coeficientes de detalhes wavelets mais prospectivos, que possam representar a maior parcela de informação de um determinado conjunto de dados. Acresce-se a isso a utilização de wavelets para solução de sistemas de equações diferenciais culminando também em aplicações para solução de problemas de otimização dinâmica.

No próximo capítulo descreveremos o algoritmo Wavelet Thresholding Adaptativo utilizado para solução DAOPs.

Capítulo 6

Algoritmo Wavelet- *Thresholding* Adaptativo (WtA)

Neste Capítulo

Tópico 6.1. Apresentação do Algoritmo Wavelet *Thresholding* Adaptativo;

Tópico 6.2. Etapa de *thresholding*.

O objetivo deste capítulo é apresentar o algoritmo wavelet-*thresholding* adaptativo proposto. A primeira parte do capítulo, **Tópico 6.1** introduz o algoritmo e detalha as principais rotinas implementadas em MATLAB[®]. O **Tópico 6.2** descreve o novo procedimento wavelet-*thresholding*: CVPshrink. No final do capítulo ilustra-se uma aplicação do WtA a um problema de controle ótimo.

6.1. Algoritmo Adaptativo para Otimização Dinâmica

6.1.1 Introdução

O algoritmo proposto neste capítulo tem como objetivo propor melhorias quanto a critérios de adaptação de malhas, a fim de desonerar a etapa de otimização. Em outras palavras, pretende-se construir um procedimento no qual a discretização se adapte automaticamente à própria solução do problema. Tal mecanismo não só reduz a dimensão no sistema NLP como também possibilita a identificação *a priori* dos pontos de junção dos perfis de controle. O resultado final possibilita maximizar o desempenho computacional garantindo a convergência para o ótimo teórico (ou ótimo de referência).

A estrutura central do algoritmo se baseia em ciclos de otimização sucessivos, com malhas sequencialmente ajustadas por análise wavelets. A inovação se baseia na inserção de critérios *thresholding*, apresentados no Capítulo 5, nas etapas de compressão dos coeficientes wavelets. Um novo critério *thresholding*, chamado CVPshrink (*Control Vector Parameterization Shrink*) foi desenvolvido. Tal critério

caracteriza-se por um cálculo intermediário de otimização multi-objetivo, o qual balanceia a taxa de compressão e a parametrização da variável de controle.

6.1.2 Estrutura do Algoritmo Wavelet-*Thresholding* Adaptativo

Este tópico apresenta a estrutura geral do algoritmo adaptativo¹³.

Etapa 1. O algoritmo inicia com um número de estágios *default*, $n_s = n_{s,o}$ (geralmente 4 ou 8 estágios). A partir do número de estágios *default*, o domínio temporal τ é discretizado uniformemente, com estágios de tamanho fixo.

$$\boldsymbol{\tau}^{\langle \ell=1 \rangle} = \left\{ \tau_1^{\langle \ell \rangle}, \dots, \tau_{n_{s,o}}^{\langle \ell \rangle} \right\} \quad (6.1)$$

onde

$$\left(\tau_{i+1}^{\langle \ell \rangle} - \tau_i^{\langle \ell \rangle} \right)_{i=1, n_s-1} = \frac{t_f - t_o}{n_{s,o}} \text{ é o tamanho de cada estágio e}$$

$$\tau_o^{\langle \ell \rangle} = t_o \quad (6.2)$$

em que ℓ indica a iteração do WtA (nessa primeira etapa $\ell = 1$).

Em cada estágio a variável de controle $u(t)$ é parametrizada de acordo com alguma regra específica. Particularmente utiliza-se parametrização constante por partes:

$$u(t) = \wp(t, \boldsymbol{\nu}) \begin{cases} u(t) = \nu_1^{\langle \ell \rangle} & \text{se } t \leq \tau_1^{\langle \ell \rangle} \\ u(t) = \nu_i^{\langle \ell \rangle} & \text{se } \tau_{i-1}^{\langle \ell \rangle} < t \leq \tau_i^{\langle \ell \rangle}, i = 2, \dots, n_s^{\langle \ell \rangle} \end{cases} \quad (6.3)$$

em que:

$$\boldsymbol{\nu}^{\langle \ell=1 \rangle} = \left\{ \nu_1^{\langle \ell \rangle}, \dots, \nu_{n_s}^{\langle \ell \rangle} \right\} \text{ é o vetor de coeficientes de controle (variáveis de decisão).}$$

Etapa 2. Em sequência ocorre o primeiro ciclo de otimização. Tal etapa resulta no valor ótimo da função objetivo, $J^{*\langle \ell \rangle}$, e trajetória ótima de controle, $\boldsymbol{\nu}^{*\langle \ell \rangle}$.

Etapa 3. Em seguida, depois do primeiro ciclo de otimização, uma nova etapa de discretização é realizada. Nessa etapa, a malha é refinada por um fator de 2, com a inserção de uma nova variável de decisão no interior de cada estágio. A parametrização da variável de controle é realizada de maneira semelhante à Etapa 2;

¹³ Considera-se para efeitos de apresentação da metodologia somente uma variável de controle $u(t)$.

Etapa 4. A partir da segunda parametrização, ocorre o segundo ciclo, $\ell = 2$, de otimização. O número de estágios dessa etapa é logicamente o dobro, $2n_{s,o}$, do número de estágios da etapa inicial;

Etapa 5. A partir da segunda etapa de otimização, inicia-se a fase adaptativa do algoritmo. A **DWT** é realizada sobre $\mathbf{v}^{*\langle\ell\rangle}$ gerando os coeficientes de detalhes equivalentes, $\mathbf{d}^{\langle\ell\rangle}$. Em seguida, o mecanismo de *thresholding* é realizado sobre $\mathbf{d}^{\langle\ell\rangle}$, gerando o vetor de coeficientes comprimidos, $\mathbf{d}_\delta^{\langle\ell\rangle}$. Esse esquema é sumarizado a seguir:

Wavelet *thresholding* (rotina *wav_thresh*)

- | | | |
|----------|---|---|
| 1 | $\mathbf{v}^* = \{v_1^*, v_2^*, \dots, v_{n_s}^*\}$ | perfil de controle ótimo |
| 2 | $\mathbf{d} = \text{DWT}(\mathbf{v}^*)$ | DWT |
| 3 | $\delta \leftarrow \text{thresh}(Tr, \mathbf{d})$ | cálculo do <i>threshold</i> (<i>Tr</i> é a regra <i>thresholding</i>) |
| 4 | $\mathbf{d}_\delta = \text{Thr}(\mathbf{d}, \delta) \cdot \mathbf{d}$ | <i>Thresholding</i> |
-

Etapa 6. Após a etapa de *thresholding*, a etapa de seleção dos estágios é então realizada. Basicamente, somente os detalhes \mathbf{d}_δ que não foram alterados são preservados. Os estágios correspondentes aos detalhes selecionados são armazenados no vetor Ω .

Seleção de estágios prospectivos (rotina *select*)

- | | | |
|-----------|--|---|
| 1 | for $n = K$ to $K - n_o$ | do nível wavelet mais elevado para o mais baixo, n_o . |
| 2 | for $m = 1$ to $2^{n-1} - 1$ | índice de posição wavelets |
| 3 | if $d_{\delta, n-1, m} \neq d_{n-1, m}$ | se o detalhe comprimido é diferente do original |
| 4 | $\Upsilon_{n, m} = m 2^{K-n+1}$ | Índice de representação desses coeficientes no nível mais baixo (<i>downsampling</i>) |
| 5 | else | |
| 6 | $\Upsilon_{n, m} = 0$ | estágio não é selecionado |
| 7 | end | |
| 8 | end | |
| 9 | end | |
| 10 | $\Omega = \bigcup(\Upsilon)$ | vetor com índice dos estágios selecionados |
-

Nota-se que o vetor Υ contém os índices de detalhes representados na escala mais baixa. Em outras palavras, um coeficiente de detalhe em um nível wavelet mais elevado pode representar um mesmo ponto representado por outro coeficiente de detalhe em um

nível wavelet diferente. Desse modo, o vetor Ω descarta os índices que se repetem eventualmente. Observa-se que essa etapa poderia simplesmente ser realizada computando-se a transformada wavelets inversa (**iDWT**) dos coeficientes comprimidos. Porém, esse procedimento quase sempre modifica os coeficientes de controle de maneira a prejudicar a nova estimativa inicial para a iteração seguinte.

Etapa 7. Em sequência, os estágios selecionados em Ω sofrem um processo de refinamento. Tal procedimento se baseia na inserção de uma nova variável de decisão em uma posição específica, $\theta \in [0,05 \ 0,95]$, do estágio (ver Figura 6.1). Dessa maneira, a malha é refinada por um fator de 2 somente nos estágios selecionados. Conseqüentemente, após essa etapa de refinamento a variável de controle é parametrizada novamente e um novo ciclo de otimização é realizado. Nesse contexto, esse ciclo adaptativo (da Etapa 4 à Etapa 7) se repete até que o critério de parada seja atendido.

Refinamento (ver Figura 6.1) (rotina *refine*)

1	$\Lambda = 1$	índice de refinamento
2	for $i = 1$ to n_s	ciclo do estágio 1 até n_s
3	for $j = 1$ to $length(\Omega)$	para cada índice de Ω
4	if $\Omega_j = i$ and $(\tau_i - \tau_{i-1}) \geq \mathcal{G}$	se o atual estágio foi selecionado e for maior qe uma tolerância mínima \mathcal{G}
5	$\mu_{\Lambda-1} = \nu_{i-1}, \mu_{\Lambda} = \nu_{i-1}$	dobra-se número de pontos
6	$\pi_{\Lambda-1} = \tau_i \theta + \tau_{i-1}(1-\theta), \pi_{\Lambda} = \tau_i$	posição dos novos pontos
7	$\Lambda = \Lambda + 2$	atualização do índice
8	else	
9	$\mu_{\Lambda} = \nu_{i-1}$	mantém ponto original
10	$\pi_{\Lambda} = \tau_i$	mantém posição original
11	$\Lambda = \Lambda + 1$	sem refinamento
12	end	
13	end	
14	end	
15	$n_s = n_s + length(\Omega)$	atualização do número de estágios
16	$\tau = \{\pi_1, \pi_2, \dots, \pi_{n_s-1}, \pi_{n_s}\}$	atualização das posições
17	$\mathbf{v} = \{\mu_1, \mu_2, \dots, \mu_{n_s-1}, \mu_{n_s}\}$	atualização dos coeficientes

A etapa de refinamento é ilustrada na Figura 6.1.

O parâmetro $\mathcal{G} \in \mathbb{R}$ é utilizado como restrição para a etapa de refinamento, a fim de garantir uma limitação para o tamanho dos estágios. O intuito dessa restrição é limitar uma aglomeração de pontos excessiva.

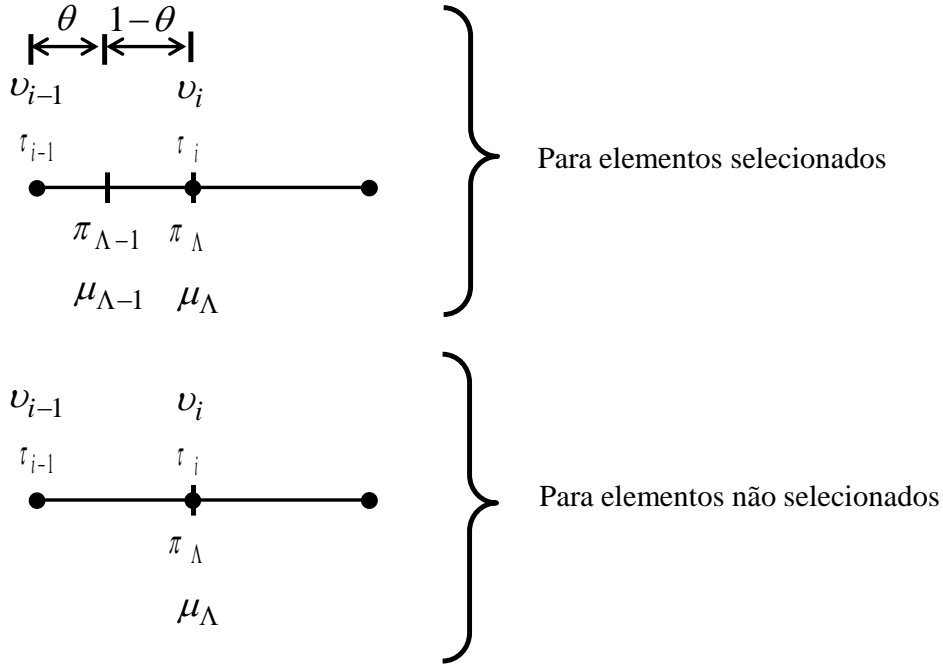


Figura 6.1. Etapa de refinamento da malha.

6.1.3 Critério de Convergência

O critério de parada do WtA, $\phi^{(\ell)}$, mede a variação relativa entre a função objetivo de duas iterações sucessivas:

$$\phi^{(\ell)} = \left\| \frac{J^{*(\ell)} - J^{*(\ell-1)}}{J^{*(\ell)}} \right\| \quad (6.4)$$

O critério de parada requer que $\phi^{(\ell)} \leq tol$, onde $tol \in \mathbb{R}$ é especificada pelo usuário. Tal critério leva a indagar se há garantia de convergência para a solução ótima analítica, ou seja, se $\lim_{\ell \rightarrow \infty} J^{*(\ell)} = J_T^*$, onde J_T^* é a solução ótima analítica. J_T^* é obtido a partir de uma solução numérica utilizando-se uma malha bastante refinada, ou seja, com $n_s = n_s^{\max}$, sendo $n_s^{\max} \in \mathbb{Z}$ um número elevado de estágios. Consequentemente, a solução obtida com a malha refinada é considerada como uma solução de referência para o algoritmo.

O algoritmo principal é detalhado a seguir. O algoritmo WtA é ilustrado na Figura 6.2.

Algoritmo principal		
1	$u_L, u_U, u_o, t_0, t_f, tol, \varphi, \theta, n_{s,o}, \ell_{max}, Tr$ $\tau^{\langle \ell=1 \rangle} = \left\{ \tau_1^{\langle \ell \rangle}, \dots, \tau_{n_{s,o}}^{\langle \ell \rangle} \right\}, \tau_o^{\langle \ell \rangle} = t_o,$	parâmetros iniciais
2	$\left(\tau^{\langle \ell \rangle}_{i+1} - \tau^{\langle \ell \rangle}_i \right)_{i=1, n_s-1} = \frac{t_f - t_o}{n_{s,o}}$	discretização equidistante inicial
3	for $\ell = 1, \dots, \ell_{max}$	iteração principal
4	$\begin{cases} u(t) = v_1^{\langle \ell \rangle} & \text{se } t \leq \tau_1^{\langle \ell \rangle} \\ u(t) = v_i^{\langle \ell \rangle} & \text{se } \tau_{i-1}^{\langle \ell \rangle} < t \leq \tau_i^{\langle \ell \rangle}, i = 2, \dots, n_s^{\langle \ell \rangle} \end{cases}$	parametrização
5	$\left(J^{*\langle \ell \rangle}, v^{*\langle \ell \rangle} \right) \leftarrow \text{NLP} \left(u(t), u_L, u_U \right)$	NLP
6	if $\ell = 1$	se é primeiro ciclo
7	$\Omega = \{1, \dots, n_{s,o}\}$ $\left(\tau^{\langle \ell+1 \rangle}, v^{\langle \ell+1 \rangle}, n_s^{\langle \ell+1 \rangle} \right) \leftarrow \dots$	seleção de todos os estágios
8	$\text{refine} \left(\Omega, \theta, \vartheta, \tau^{*\langle \ell \rangle}, v^{*\langle \ell \rangle}, n_s \right)$	etapa de refinamento
9	else	se não for primeiro ciclo
10	if $\phi^{\langle \ell \rangle} < tol$	critério de parada, Equação (6.4)
10	FIM	
12	else	
13	$\mathbf{d}_\delta^{\langle \ell \rangle} \leftarrow \text{wav_thresh} \left(v^{*\langle \ell \rangle}, Th_rule \right)$	<i>thresholding</i>
14	$\Omega \leftarrow \text{select} \left(\mathbf{d}_\delta^{\langle \ell \rangle} \right)$ $\left(\tau^{\langle \ell+1 \rangle}, v^{\langle \ell+1 \rangle}, n_s^{\langle \ell+1 \rangle} \right) \leftarrow \dots$	seleção de estágios
15	$\text{refine} \left(\Omega, \theta, \vartheta, \tau^{*\langle \ell \rangle}, v^{*\langle \ell \rangle}, n_s \right)$	refinamento
16	end	
17	end	
18	end	

Note que o algoritmo pode ser aplicado aos métodos sequencial, *multiple-shooting* e simultâneo. Isso se deve ao fato de o algoritmo intervir na etapa de transição entre dois ciclos sucessivos de otimização, independentemente do método utilizado. Esta tese deu ênfase ao método sequencial, porém com algumas aplicações ao método *multiple shooting* com paralelização do código.

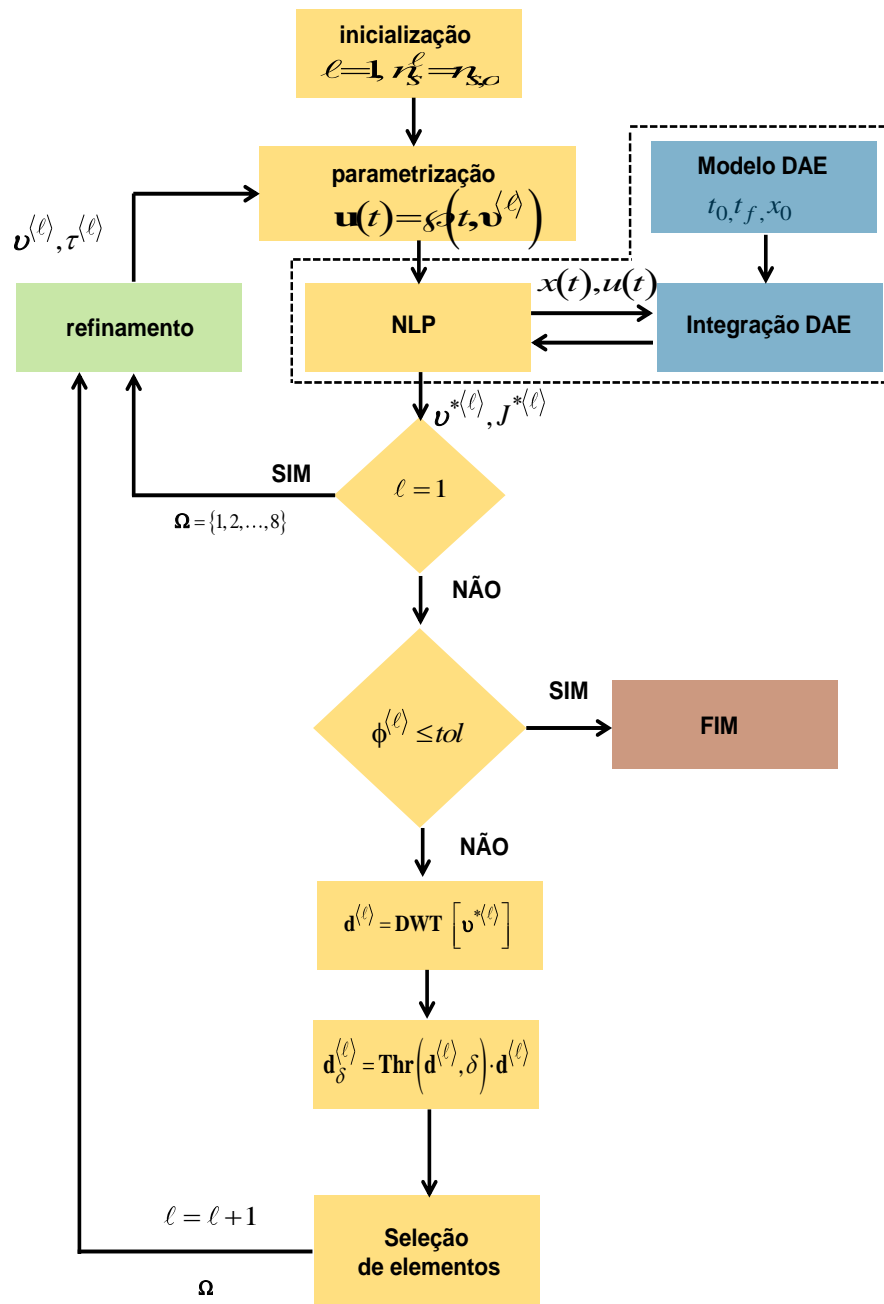


Figura 6.2. Estrutura geral do algoritmo Wavelet-*thresholding* Adaptativo (WtA).

6.1.3 Rotinas Implementadas

O algoritmo WtA foi implementado em MATLAB 7.6[®] gerando uma *toolbox* com um ambiente de simulação para facilitar a implementação de problemas de controle ótimo com diferentes características: tempo final livre, restrições de trajetória, restrições finais e múltiplas variáveis de controle. Foram desenvolvidas rotinas para o cômputo automático das equações de sensibilidade, baseada em computação simbólica.

As principais rotinas do MATLAB[®] utilizadas no desenvolvimento são listadas na tabela a seguir.

Tabela 6.1. Rotinas utilizadas.

<i>fmincon</i>	Rotinas de otimização do Matlab (ponto interior e SQP)
<i>waverec</i>	Rotina de decomposição wavelets
<i>wavedec</i>	Rotina de reconstrução wavelets
<i>dasslc</i> (COSTA <i>et al.</i> , 2002)	Rotina de integração de DAE
<i>ode45</i> , <i>ode23s</i>	Rotina de integração de DAE

A Figura 6.4 ilustra a estrutura geral do algoritmo implementado MATLAB[®].

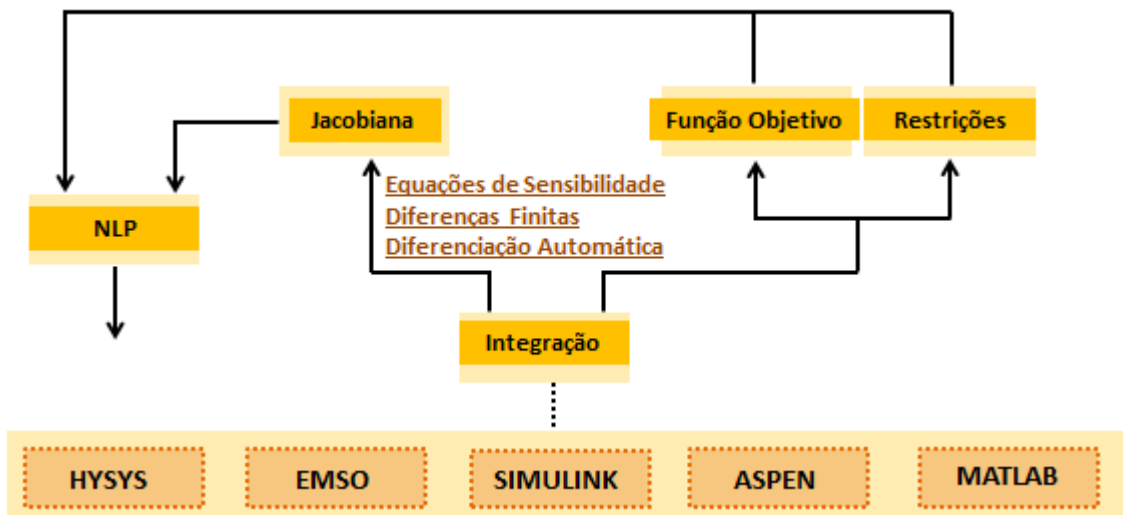


Figura 6.4. Estrutura da *toolbox* MATLAB.

Conforme ilustrado na figura o algoritmo possibilita implementação de modelos em softwares externos como o EMSO (SOARES e SECCHI, 2003), HYSYS[®] e ASPEN[®]. Também é possível a implementação de modelos no próprio MATLAB[®] e também no ambiente SIMULINK.

Ao longo da tese diversas implementações foram testadas no código, como integração com os softwares Dynopt (CIZNIAR *et al.*, 2005), WAVELAB (JANSEN, 200) e TOMLAB (RUTQUIST e EDVALL, 2010). Porém, Neste trabalho ilustraremos apenas casos com modelos desenvolvidos em MATLAB[®]. Tal escolha permitiu a análise de um conjunto maior de exemplos, devido à maior facilidade de implementação e robustez.

6.2. Etapa de *Thresholding*

Os principais critérios *thresholding* utilizados no algoritmo WtA foram apresentados no Capítulo 5: *Visushrink* e *Sureshrink*. São regras desenvolvidas a partir de perspectivas diferentes. Estudos aplicados ao tratamento de imagens (WIELAND, 2009; JANSEN, 2000), evidenciam que diferentes regras *thresholding* possam interferir sobre o desempenho do algoritmo WtA. No entanto, baseado em observações do funcionamento de tais critérios, propõe-se a implementação de uma nova regra *thresholding* definida como *Control Vector Parameterization Shrink* (CVPS).

A ideia central do CVPS caracteriza-se por dois objetivos principais da etapa de *thresholding* do WtA:

- (i) Maximizar a taxa de compressão, ou seja, minimizar o número de coeficientes de detalhes wavelets após a etapa de *thresholding*;
- (ii) Minimizar o desvio médio quadrático entre o vetor de parâmetros de controle comprimido e o vetor original. Em outras palavras, deseja-se minimizar o efeito da compressão sobre o perfil de controle original. Logo, para cada nível de resolução wavelets n o algoritmo calcula:

$$\delta_n^{cvp} = \underset{\delta}{\mathbf{argmin}} \mathfrak{I}_n$$

s.a

$$\delta_{\min} \leq \delta \leq \delta_{\max} \tag{6.5}$$

em que \mathfrak{I}_n é a função objetivo definida como:

$$\Gamma_n = \underbrace{\frac{\|\mathbf{v} - \mathbf{v}_\delta\|}{\|\mathbf{v} - \mathbf{v}_{\delta_{\max}}\|}}_{\text{parametrização}} + \underbrace{\frac{\mathbf{card}(\mathbf{d}_n) - \mathbf{card}(\mathbf{d}_{n,\delta})}{\mathbf{card}(\mathbf{d}_n)}}_{\text{compressão}} \tag{6.6}$$

e:

$$\delta_{\min} = \mathbf{mín} \left(\left\| \mathbf{d}_{n,m=0,\dots,2^n-1} \right\| \right) \text{ e } \delta_{\max} = \mathbf{máx} \left(\left\| \mathbf{d}_{n,m=0,\dots,2^n-1} \right\| \right) \tag{6.7}$$

em que $\mathbf{v}_\delta = \mathbf{iDWT}(\mathbf{d}_\delta)$ e $\mathbf{d}_\delta = \mathbf{Thr}(\mathbf{d}, \delta_n^{cvp}) \cdot \mathbf{d}$.

Na Equação (6.6), $\mathbf{card}(\mathbf{d}_{n,\delta})$ é a cardinalidade da versão comprimida de \mathbf{d}_n . Note que a função objetivo é normalizada para garantir que os dois termos estejam entre

0 e 1. Logo, o primeiro termo é 1 quando δ é máximo. O segundo termo é 1 quando δ é mínimo.

Após o cálculo do *threshold*, o procedimento de *soft thresholding* é realizado:

$$\mathbf{Thr}(d_{n,m}, \delta_n^{cvp}) = d_{n,m} - \mathbf{sgn}(d_{n,m}) \quad (6.8)$$

em que:

$$\mathbf{sgn}(d_{n,m}) = \begin{cases} -1 & \text{if } d_{n,m} < -\delta^{cvp} \\ 0 & \text{if } d_{n,m} \in [-\delta^{cvp}, \delta^{cvp}] \\ 1 & \text{if } d_{n,m} > \delta^{cvp} \end{cases} \quad (6.9)$$

A Figura 6.3 ilustra um típico CVPS *thresholding* de um vetor \mathbf{v} :

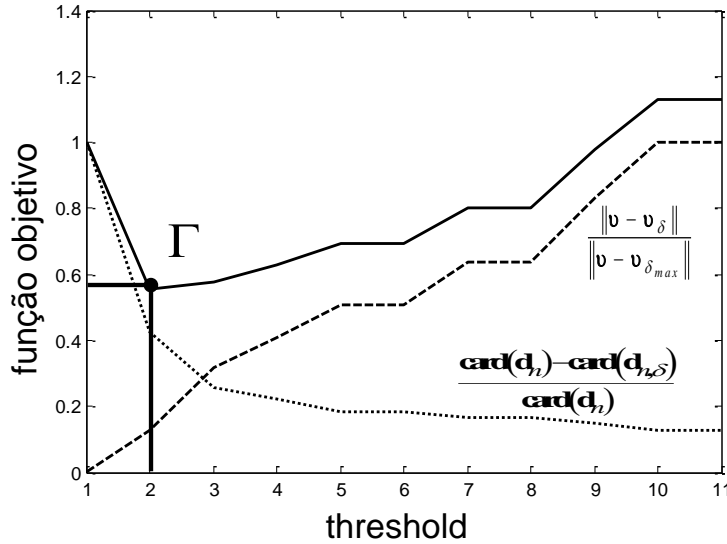


Figura 6.3. Ilustração: CVPS em um nível wavelet.

Na Figura 6.3 nota-se que o menor valor da função objetivo ocorre no valor de $\delta_n^{cvp} = 2$. A função definida na Equação (6.6) possui uma estrutura semelhante à estrutura definida pela definição do risco ideal introduzido no Capítulo 5.

$$R_{ideal} = \sum_i \theta_i^2 (1 - \alpha_i)^2 + \sigma^2 \sum_i \alpha_i^2 \quad (5.87)$$

sendo que:

$$\mathbf{d} = \boldsymbol{\theta} + \mathbf{w} \quad (5.74)$$

e o valor ótimo de cada α_i é:

$$\alpha_i^* = \theta_i^2 / (\theta_i^2 + \sigma^2) \quad (5.89)$$

Na função R_{ideal} observa-se que o primeiro termo pondera a contribuição dos coeficientes wavelets sem ruído (desconhecidos), enquanto o segundo termo pondera a contribuição da variância do suposto ruído presente.

No próximo exemplo simula-se um esquema de *thresholding* de um conjunto de dados com nível de ruído conhecido adicionado ao mesmo. O objetivo é comparar o CVPS com o *threshold* ideal.

Exemplo 6.1. Análise Wavelet *thresholding*

Ao considerar um vetor com 32 pontos conforme ilustrado na figura seguinte:

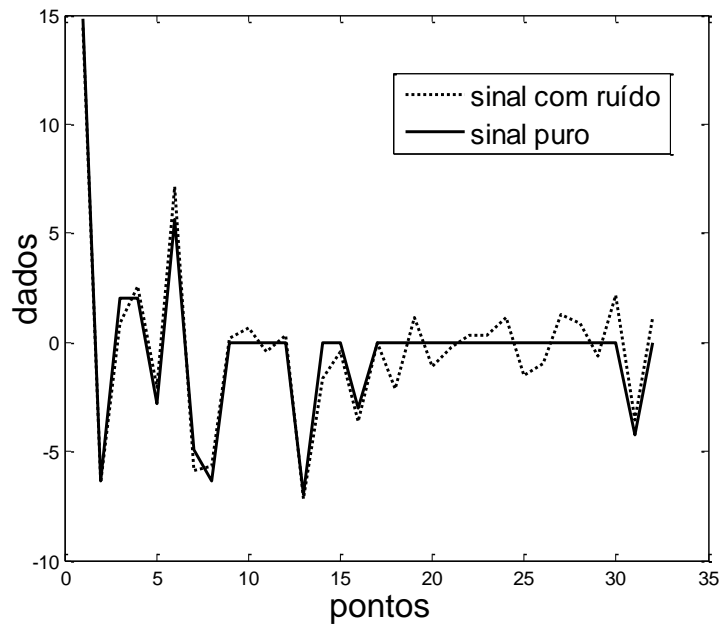


Figura 6.4. Coeficientes wavelets.

Observa-se que o sinal original \mathbf{d} é resultante da soma do sinal sem ruído (puro), θ , com a parcela com ruído \mathbf{w} ($\sigma = 1$), conhecidos. Como todos os componentes são conhecidos, computa-se o valor de R_{ideal} para diversos valores de α_i , variando-se de 0 a 1. Note que há várias combinações possíveis, logo consideramos que $\alpha_i, i = 1, \dots, 32$, é constante: α . À primeira vista tal suposição é suficiente para avaliar o comportamento de R_{ideal} em função de α , que pode ser visualizado na Figura 6.5.

Na Figura 6.5, no eixo horizontal quantifica-se o valor de α multiplicado pelo

valor máximo de *threshold*, 2. Observa-se que R_{ideal} tende a crescer para valores extremos de α , apresentando um valor mínimo em torno de 0.6. O valor ótimo de R_{ideal} , de acordo com a Equação (5.89) e (5.87) foi de 5.55.

Na Figura 6.6. ilustra-se a variação do CVPS para cada valor de *threshold*, variando de 0 a 2. Nesse caso, manipulou-se somente os coeficientes de níveis mais elevados (de 24 a 32). Observa-se que também há uma região ótima de *threshold*, em torno de 0.6.

De acordo com essa avaliação, espera-se que o CVPS apresente-se como uma estratégia alternativa ao *Visushrink* e *Sureshrink* introduzidos no capítulo anterior. Para esse exemplo específico, deduz-se que o CVPS computa um valor de *threshold* mínimo próximo ao mínimo de R_{ideal} . No próximo capítulo será possível analisar o desempenho desse método acoplado ao WtA para solução de DAOPs.

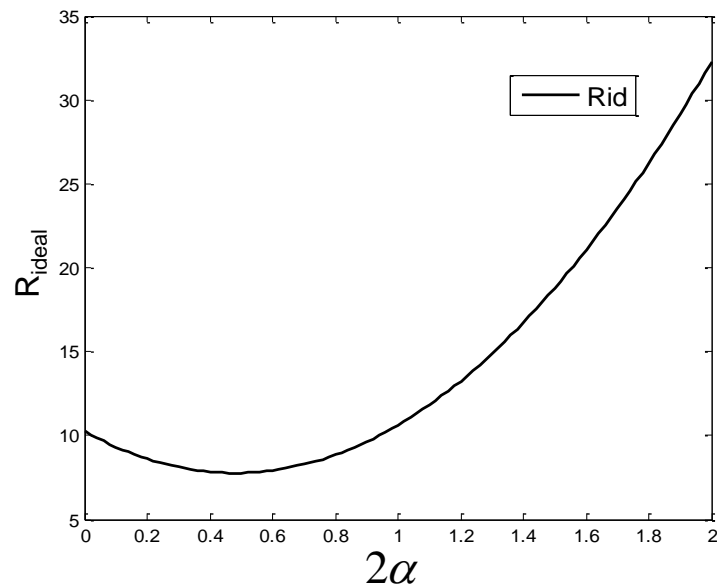


Figura 6.5. R_{ideal} em função de α multiplicado pelo valor máximo de *threshold*, 2.

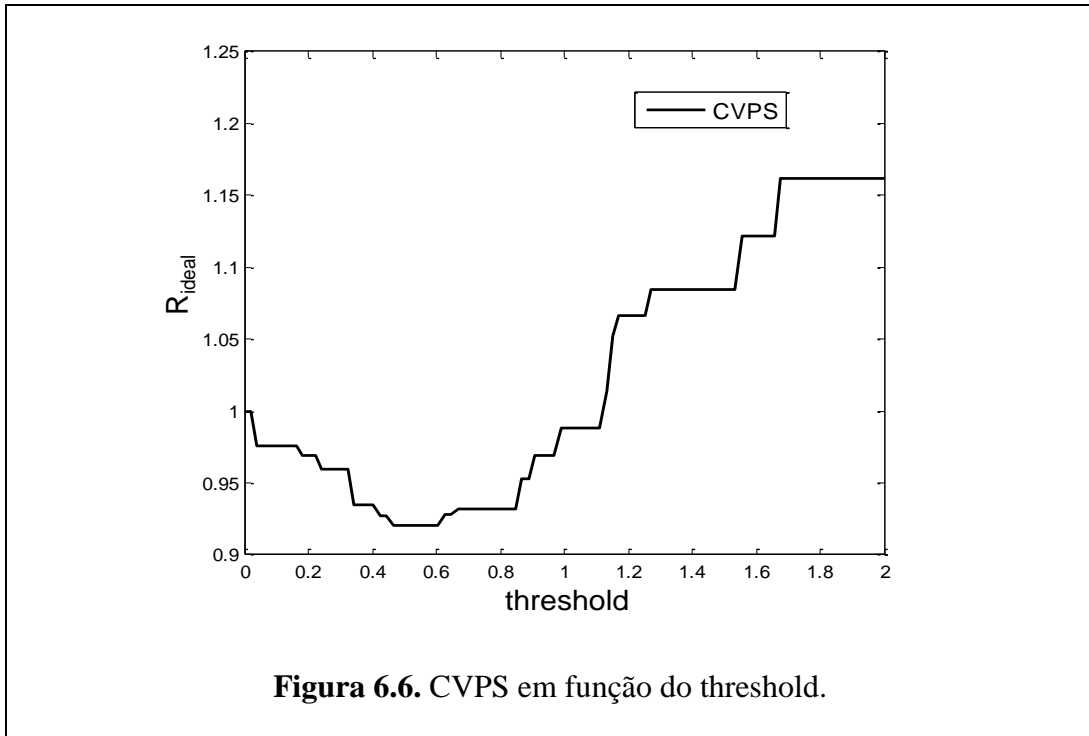


Figura 6.6. CVPS em função do threshold.

O CVPS apresentou valores muito próximos à regra de Validação Cruzada (NASON, 1996; JANSEN, 2000), também testada no algoritmo em um trabalho recente (SANTOS *et al.*, 2014). Ambos os algoritmos são resultantes de uma etapa de otimização visando encontrar um *threshold* ótimo.

Já preluçando o próximo capítulo, o exemplo seguinte ilustra uma aplicação do WtA em um problema de controle ótimo de um biorreator.

Exemplo 6.2. Aplicação do WtA a um problema de controle ótimo.

Esse exemplo também foi solucionado por SRINIVASAN *et al.* (2003a) e QUINTO (2010). Trata-se de um biorreator batelada alimentada onde a adição controlada de nutrientes afeta diretamente a taxa de crescimento da cultura celular e permite controlar fluxo metabólico. Assim, a operação pode ser definida como a busca pela política ótima de adição de um dos reagentes (QUINTO, 2010).

A taxa de crescimento da cultura é escrita como:

$$q(S) = \frac{q_m \cdot S}{K_m + S + \frac{S^2}{K_i}} \quad (6.10)$$

Já o modelo matemático do biorreator pode ser escrito como:

$$\frac{dX}{dt} = q(S) \cdot X - \frac{u}{V} \cdot X, \quad X(0) = X_0 \quad (6.11)$$

$$\frac{dS}{dt} = -\frac{q(S)}{Y_x} \cdot X - \frac{v}{Y_p} \cdot X + \frac{u}{V} \cdot (S_{in} - S), \quad S(0) = S_0 \quad (6.12)$$

$$\frac{dP}{dt} = mi \cdot X - \frac{u}{V} \cdot P, \quad P(0) = P_0 \quad (6.13)$$

$$\frac{dV}{dt} = u, \quad V(0) = V_0 \quad (6.14)$$

em que S é a concentração de substrato, X é a concentração de biomassa, P a concentração de produto, V o volume, u a taxa de alimentação, S_{in} a concentração do substrato na entrada, q_m, K_m, K_i e mi parâmetros cinéticos e Y_x, Y_p coeficientes de produção.

Utilizou-se aqui a simplificação do modelo proposta em QUINTO (2010) e SRINIVASAN *et al.* (2003a).

$$\frac{dx_1}{dt} = q(S) \cdot x_1 \quad (6.15)$$

$$\frac{dx_2}{dt} = v \cdot x_1 \quad (6.16)$$

$$\frac{dx_3}{dt} = u \quad (6.17)$$

com $x_1 = X \cdot V$, $x_2 = P \cdot V$ e $x_3 = V$, onde a concentração de substrato é obtida por balanço de massa:

$$S = \frac{1}{x_3} \cdot (S_0 \cdot V_0 + S_{in} \cdot (x_3 - V_0) - \frac{1}{Y_x} (x_1 - X_0 \cdot V_0) - \frac{1}{Y_p} \cdot (x_2 - P_0 \cdot V_0)) \quad (6.18)$$

Logo o problema de otimização é definido como:

$$J = \underset{u}{\text{máx}} [P(t_f)] \quad (6.19)$$

sujeito a:

$$\begin{cases} u_{\text{mín}} \leq u \leq u_{\text{máx}} \\ X(t) \leq X_{\text{máx}} \end{cases} \quad (6.20)$$

As variáveis e parâmetros podem ser visualizados a seguir:

Variável	Descrição	Unidade	Valor
q_m	Parâmetro cinético	$L \cdot h^{-1}$	0,53
k_m	Parâmetro cinético	$g \cdot L^{-1}$	1,2
k_i	Parâmetro cinético	$g \cdot L^{-1}$	22
mi	Parâmetro cinético	$L \cdot h^{-1}$	0,5
Y_x	Coefficiente de produção	-	0,4
Y_p	Coefficiente de produção	-	1
S_{in}	Concentração de substrato na entrada	$g \cdot L^{-1}$	20
u_{min}	Concentração mínima de u	$L \cdot h^{-1}$	0
u_{max}	Concentração máxima de u	$L \cdot h^{-1}$	1
X_{max}	Concentração máxima de X	$g \cdot L^{-1}$	3
t_f	Instante final	h	8
X_0	Concentração inicial de X	$g \cdot L^{-1}$	1
S_0	Concentração inicial de S	$g \cdot L^{-1}$	0
P_0	Concentração inicial de P	$g \cdot L^{-1}$	0
V_0	Concentração inicial de V	L	2

Segundo SRINIVASAN *et al.* (2003) esse problema de otimização pode ser dividido em quatro fases (de acordo com resultados analíticos obtidos pelo autor):

- (i) Primeira Fase: deve-se ter $u = u_{max}$ para aumentar a produção de S o mais rápido possível;
- (ii) Segunda fase: uma vez que a concentração de substrato S chega a seu valor ótimo então $u = u_{ótimo}$ é aplicado para manter S nesse valor e então aumentar a concentração de X e P o mais rápido possível.
- (iii) Terceira fase: a variável de controle é reduzida ao seu valor mínimo $u = u_{min}$ de maneira a obter o mais rápido possível o valor de equilíbrio do substrato.
- (iv) Quarta fase: quando a biomassa alcança seu valor máximo, $X = X_{max}$, a variável de controle u deve ser manipulada para manter $X = X_{max}$ e S no seu valor de equilíbrio.

A variação dos reagentes e produtos (variáveis de estado) pode ser visualizada nas Figuras 6.7 (a)-(d).

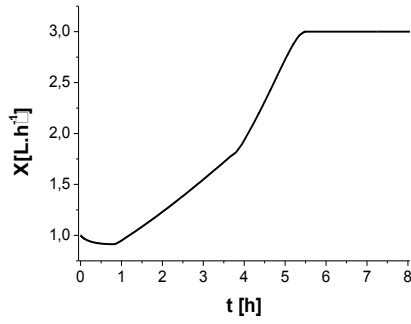


Figura 6.7 (a) Concentração de X com o tempo

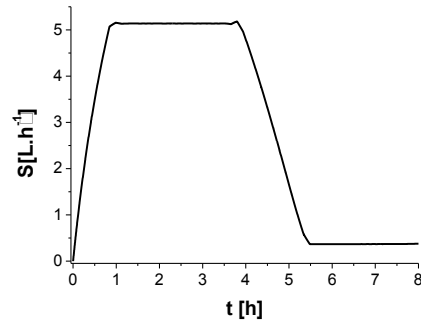


Figura 6.7 (b) Concentração de S com o tempo

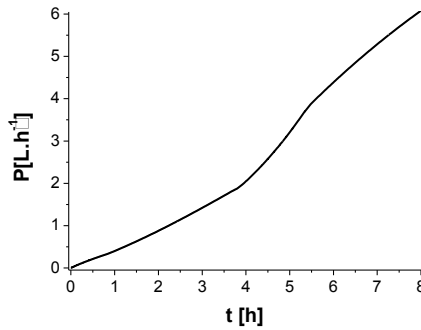


Figura 6.7 (c) Concentração de P com o tempo

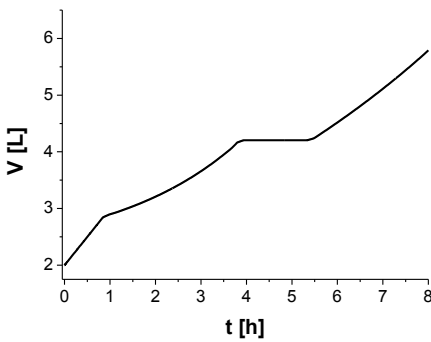


Figura 6.7 (d) Concentração de V com o tempo

QUINTO (2010) obteve o valor ótimo de J de $6,085 \text{ g} \cdot \text{L}^{-1}$. A solução numérica, Figura 6.8, obtida com 64 pontos foi suficiente para obter esse mesmo valor. O custo computacional, nesse caso, foi de 6342 segundos¹⁴.

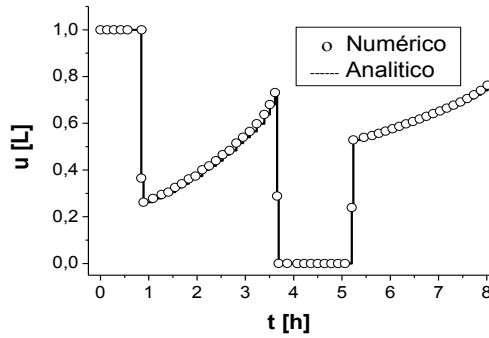
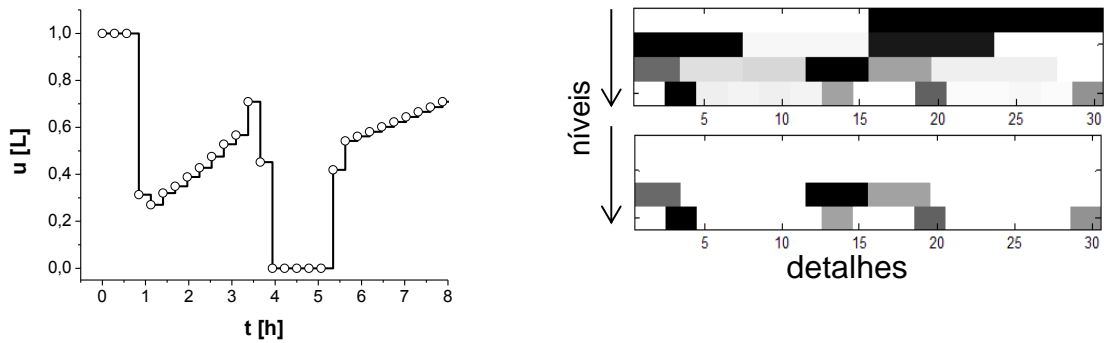


Figura 6.8. Trajetória de controle com 64 estágios.

As Figuras 6.9 (a)-(c) mostram o perfil temporal da variável de controle em cada estágio, assim como a matriz de coeficientes de detalhes, onde os quadros mais escuros correspondem a coeficientes mais representativos e os quadros mais claros correspondem a coeficientes menos representativos, sendo que o eixo vertical representa o nível de resolução crescente (de cima para baixo) e o eixo horizontal representa a posição da discretização da variável de controle (conforme exemplificado no Capítulo 5).

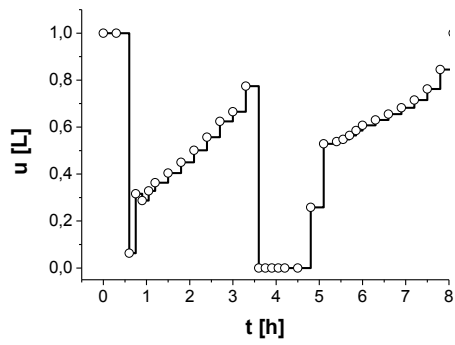


Evolução temporal da variável de controle

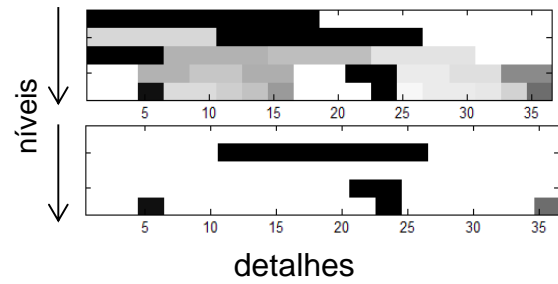
Coefficientes de detalhes antes e depois da compressão

Figura 6.9 (a) Iteração 1

¹⁴ Em ambiente MATLAB 7.6[®], integrador *dasslc* (VIEIRA e SECCHI, 2001), com tolerância relativa de 10^{-7} e tolerância absoluta de 10^{-7} , rotina *fmincon* do MATLAB 7.6[®] (SQP) com tolerâncias (função objetivo, restrições e variável de estado) de 10^{-6} .

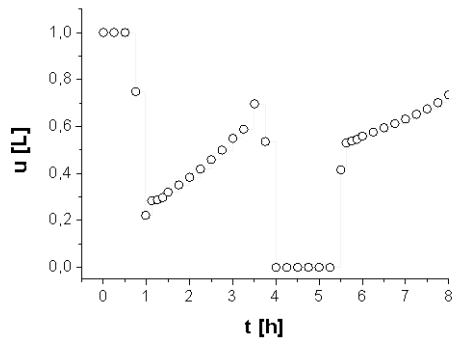


Evolução temporal da variável de controle

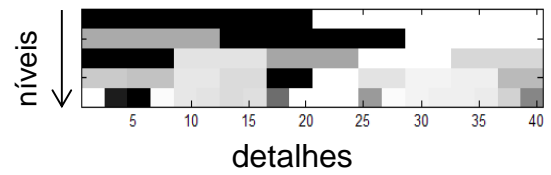


Coeficientes de detalhes antes e depois da compressão

Figura 6.9 (b) Iteração 2



Evolução temporal da variável de controle



Coeficientes de detalhes antes e depois da compressão

Figura 6.9 (c) Iteração 3

Conforme se observa no espectro de detalhes, os coeficientes de detalhes das camadas mais refinadas são preservados a cada iteração do WtA. Dessa maneira, o algoritmo tende a refinar somente em torno dos estágios de discretização detectados a cada iteração.

Capítulo 7

Aplicações

Neste Capítulo

Tópico 7.1. Reator Semi-batelada Isotérmico;

Tópico 7.2. Reator CSTR Não Linear;

Tópico 7.3. Análise de Desempenho;

Tópico 7.4. Resultados: Trajetória de Controle

Nesse capítulo aplica-se o algoritmo WtA sobre alguns *benchmarks* de controle ótimo. Nos **Tópicos 7.1** e **7.2** dois exemplos são utilizados para ilustrar a aplicabilidade do WtA. No **Tópico 7.3** 22 problemas são solucionados e analisados para as três estratégias WtA implementadas.

7.1. Reator Semi-Batelada Isotérmico (SRINIVASAN *et al.*, 2003a)

7.1.1 Apresentação do Problema

O problema estudado neste tópico é um reator semi-batelada isotérmico, estudado por SRINIVASAN *et al.* (2003a) e também por SCHLEGEL (2004) e QUINTO (2010).

O modelo desse reator é representado por:



$$\frac{dx_a}{dt} = -k_1 x_a x_b - \frac{u}{V} x_a \quad (7.2)$$

$$\frac{dx_b}{dt} = -k_1 x_a x_b - 2k_2 x_b^2 + \frac{u}{V} (x_{b,in} - x_b) \quad (7.3)$$

$$\frac{dV}{dt} = u \quad (7.4)$$

$$x_c = \frac{1}{V} (x_{a,o} V_o - x_a V) \quad (7.5)$$

$$x_d = \frac{1}{2V} \left[(x_a + x_{b,in} - x_b) V - (x_{a,o} + x_{b,in} - x_{b,o}) V_o \right] \quad (7.6)$$

em que a Equação (7.1) descreve as reações envolvidas e as Equações (7.2) a (7.6) os balanços de massa com as seguintes condições iniciais:

$$x_a(0) = x_{a_0}; x_b(0) = x_{b_0}; V(0) = V_0 \quad (7.7)$$

O objetivo do problema é maximizar a seguinte função:

$$J = \max_u [x_c(t_f)V(t_f)] \quad (7.8)$$

sujeita às seguintes restrições:

$$\begin{cases} u_{\min} \leq u \leq u_{\max} \\ x_b(t_F) \leq x_{b,\max} \\ x_d(t_F) \leq x_{d,\max} \end{cases} \quad (7.9)$$

Considerando que o objetivo é a maximização da função descrita pela Equação (7.8), o algoritmo de otimização irá manipular a vazão volumétrica do reagente B , a fim de maximizar a produção de C no instante final t_F , satisfazendo as restrições. As variáveis e parâmetros desse problema podem ser visualizados na Tabela 7.1.

Tabela 7.1. Parâmetros e variáveis do Problema 7.1

1	Descrição	Unidade	Valor
k_1	Constante de reação de A	$L \cdot \text{mol}^{-1} \cdot \text{min}^{-1}$	0,053
k_2	Constante de reação de B	$L \cdot \text{mol}^{-1} \cdot \text{min}^{-1}$	0,128
$x_{b,in}$	Concentração de alimentação de B	$\text{mol} \cdot L^{-1}$	5
u_{\min}	Limite inferior de $u(t)$	$L \cdot \text{min}^{-1}$	0
u_{\max}	Limite superior de $u(t)$	$L \cdot \text{min}^{-1}$	0,001
$x_{b,\max}$	Concentração máxima de B	$\text{mol} \cdot L^{-1}$	0,025
$x_{d,\max}$	Concentração máxima de D	$\text{mol} \cdot L^{-1}$	0,15
x_{a_0}	Concentração inicial de A	$\text{mol} \cdot L^{-1}$	0,72
x_{b_0}	Concentração inicial de B	$\text{mol} \cdot L^{-1}$	0,05
V_o	Volume inicial	L	1
t_F	Instante final	min	250

De acordo com SRINIVASAN *et al.* (2003) e QUINTO (2010), o problema pode ser descrito em 3 fases, devido às restrições de desigualdade nas variáveis de estado e controle. Estas fases são obtidas através da análise física do problema e podem ser descritas conforme abaixo:

- (i) Primeira Fase: deve-se configurar $u = u_{m\acute{a}x}$ para aumentar a concentração do produto B o mais rápido possível;
- (ii) Segunda fase: limitar a quantidade de D produzida através da solução da trajetória ótima de u ;
- (iii) Terceira fase: Deve-se configurar $u = u_{m\acute{i}n}$ para se obter $x_b(t_f) \leq x_{b,m\acute{a}x}$.

A variação dos reagentes e produtos (variáveis de estado) pode ser visualizada nas Figuras 7.1-(a)-(c). A Figura 7.1-a representa a variação da concentração do reagente A com o tempo. Observa-se que ao longo do tempo o reagente A é consumido quase que totalmente para produzir C . Analisando-se o comportamento do reagente B , Figura 7.1b é possível visualizar que a concentração desse reagente aumenta a partir da concentração inicial e se mantém em certo patamar em torno de $0,075 \text{ L} \cdot \text{min}^{-1}$, até decrescer abruptamente no instante, de 200 min, devido ao corte da vazão de B . Na Figura 7.1c visualiza-se a concentração do produto C a qual cresce à medida que os reagentes são consumidos.

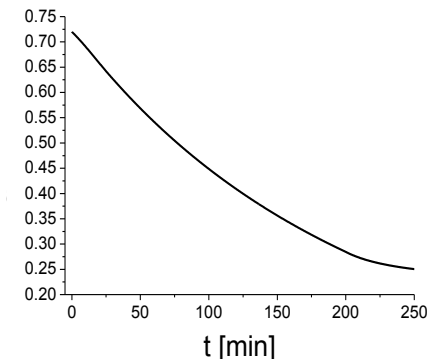
7.1.2. Aplicação do WtA

O perfil temporal da variável de controle pode ser visualizado na Figura 7.2. Os pontos representam a solução numérica (variáveis de decisão do problema de otimização) e a trajetória contínua representa a solução analítica, obtida por PMP. Pode-se observar que a vazão $u(t)$ decresce do valor inicial $10^{-3} \text{ L} \cdot \text{min}^{-1}$ até o valor de $0,6 \cdot 10^{-3} \text{ L} \cdot \text{min}^{-1}$ no instante aproximado de 200 min. A partir desse momento a vazão é interrompida. Esse comportamento pode ser explicado se observamos o modelo do processo. Como o objetivo é maximizar a produção do produto C no instante final t_f , a otimização tende a favorecer a produção de C e evitar que a produção de D atinja o limite máximo $x_{d,m\acute{a}x}$.

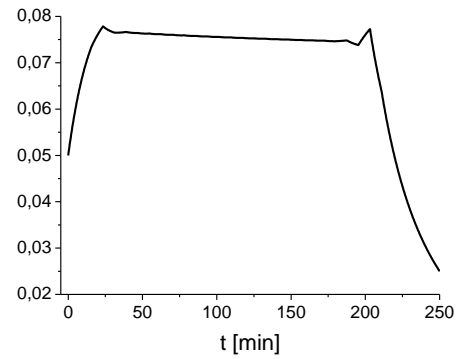
A solução ótima obtida com 128 estágios de discretização corresponde a 0,431721 moles do produto C , que é a própria solução analítica (SRINIVASAN *et al.*, 2003a).

A Figura 7.3 ilustra agora a evolução do perfil de controle, adotando-se o WtA *Sureshrink*, partindo-se de uma malha equidistante com oito estágios de discretização.

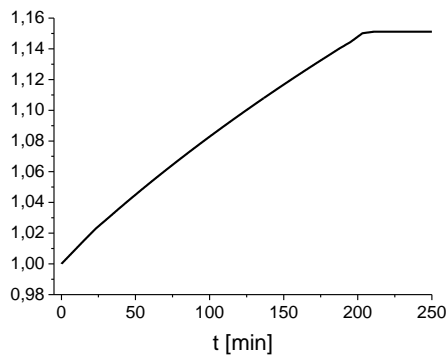
Nota-se que a cada iteração do WtA novos pontos (linhas tracejadas) são adicionados em torno das regiões de descontinuidade.



(a) Concentração de A (L/min).

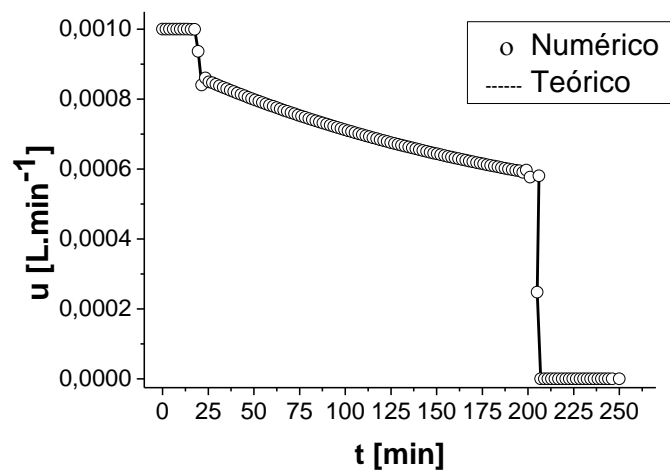


(b) Concentração de B (L/min).



(c) Evolução temporal de V (L).

Figura 7.1. Evolução temporal das variáveis de estado (Reator Semi-Batelada Isotérmico).



$$n_s = 128$$

Figura 7.2. Evolução temporal da variável de controle com 128 pontos de discretização.

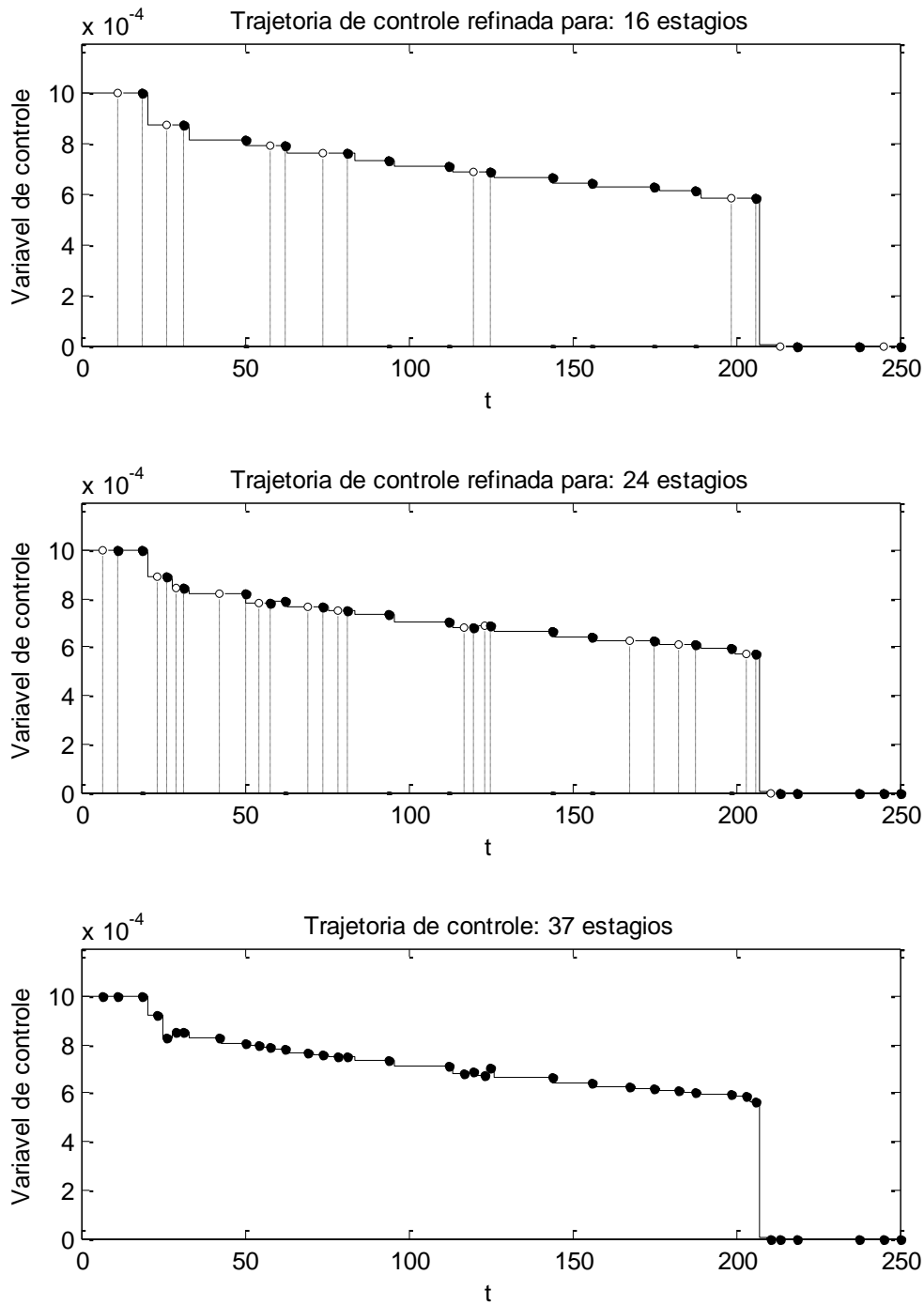


Figura 7.3. WtA *Sureshrink* (Reator Semi-Batelada Isotérmico).

A Tabela 7.1 compara as estratégias *Sureshrink* (SS), *Visushrink* (VS) e CVPshrink (CVPS). Como referência, adota-se a solução refinada com 128 estágios de discretização. Conforme explicado no Capítulo 7, o critério de parada é definido previamente de acordo com uma tolerância que avalia a variação relativa da função objetivo em cada iteração do WtA. Na Tabela 7.2 mostra-se, a cada iteração, o desempenho computacional (CPU) adimensionalizada em relação à solução de

referência, número de estágios n_s , o valor da função objetivo, J^* e taxa de convergência, ϕ .

(i) Solução de referência (equidistante)

O algoritmo é iniciado a partir de uma malha equidistante (com parametrização constante por partes) dobrando-se sucessivamente o número de estágios com atualização das estimativas iniciais a cada nível de discretização. Claramente, observa-se uma redução no valor da função objetivo para um perfil de controle cada vez mais refinado. No entanto, conforme esperado, a CPU por iteração também aumenta, devido ao maior número de variáveis de decisão.

É interessante observar que a variação relativa da função objetivo ótima, J^* em relação ao pior valor, é de aproximadamente 0,12%, ou seja:

$$f_{R,o} = \left\| \frac{J^{*\langle \ell_{máx} \rangle} - J^{*\langle \ell_{mín} \rangle}}{J^{*\langle \ell_{mín} \rangle}} \right\| \cdot 100 = 0,12\% \quad (7.10)$$

Define-se $f_{R,o}$ como fator de redução. Para todos os problemas analisados neste capítulo veremos que o $f_{R,o}$ é um parâmetro importante para avaliar a viabilidade da aplicação do WtA.

(ii) *Sureshrink* (SS)

Conforme resumido na Tabela 7.2, o método *Sureshrink* satisfaz o critério de parada após quatro iterações do algoritmo WtA, com CPU acumulada de 0,1 e 37 estágios. Logo, nota-se que quatro problemas de otimização são necessários para a convergência, porém, mesmo assim o custo computacional é notavelmente inferior. Ademais, conforme observado, a função objetivo final converge para a função objetivo de referência: 0,431721.

(iii) *Visushrink* (VS)

A Tabela 7.2 mostra que o método *Visushrink* satisfaz o critério de parada após seis iterações do algoritmo WtA, com CPU acumulada de 0,15 e 39 estágios. O desempenho computacional do método foi relativamente superior ao SS, porém ainda permite significativa redução do custo computacional.

(iii) *CVPS*

O método CVPS apresenta um resultado relativamente próximo ao SS, com quatro iterações e CPU ligeiramente inferior: 0,08. A semelhança entre os métodos é a característica adaptativa, onde cada valor de *threshold* é calculado em cada nível de resolução wavelets.

A Tabela 7.3 ilustra os valores calculados de *threshold* para cada método. Nota-se que o VS computa um valor único para todos os níveis de decomposição, enquanto os métodos SS e CVPS calculam diferentes *thresholds* para cada nível. À primeira vista, nota-se que maiores valores de *threshold* são calculados em níveis de resolução mais baixos, enquanto que nos níveis de resolução mais altos, esses valores tendem a ser menores, proporcionalmente à magnitude dos coeficientes.

Por conseguinte, uma boa justificativa para a superioridade dos métodos SS e CVPS para esse caso seria devido à capacidade dos mesmos de se ajustarem aos diferentes níveis de resolução. No entanto, tal afirmação é prematura e um conjunto maior de problemas deveria ser analisado para conclusões mais precisas.

Um estudo avaliando a sensibilidade do desempenho computacional em relação a diferentes valores de *threshold* foi realizado. A Figura 7.4 mostra dois perfis nomeados como caso A e caso B. Para ambas as curvas avalia-se o desempenho computacional (CPU) para cinco valores de *thresholds*. Para cada ponto, mostra-se também o número de estágios resultante. O caso A é justamente o modelo original apresentado (Equações 7.1.-7.9). O caso B é uma versão modificada, com $k_1 = 1,153$, meramente ilustrativo, para avaliar se a influência do valor de *threshold* se mantém. De fato, nota-se que os dois perfis se distinguem, o que indica manter um fator de *threshold* fixo pode comprometer o desempenho do método. A figura mostra que (para o caso A) um valor de *threshold* de $5 \cdot 10^{-6}$ resulta no CPU mínimo de aproximadamente 0,12 e com 36 estágios de discretização.

Tabela 7.1. Desempenho das diferentes estratégias (Reator Semi-Batelada Isotérmico) (tolerância de $0,14 \cdot 10^{-6}$)

ℓ	Equidistante				VS				SS				CVPS			
	$J^* (\cdot 10^{-2})$	CPU	n_s	$\phi (\cdot 10^6)$	$J^* (\cdot 10^{-2})$	CPU	n_s	$\phi (\cdot 10^6)$	$J^* (\cdot 10^{-2})$	CPU	n_s	$\phi (\cdot 10^6)$	$J^* (\cdot 10^{-2})$	CPU	n_s	$\phi (\cdot 10^6)$
1	43,1192	0,032	8	-	43,1192	0,032	8	-	43,1192	0,032	8	-	43,1192	0,032	8	-
2	43,1720	0,014	16	1230	43,1720	0,014	16	1230	43,1720	0,014	16	1230	43,1720	0,014	16	1230
3	43,1720	0,061	32	1,749	43,1720	0,038	22	2,989	43,1720	0,044	24	2,686	43,1720	0,034	22	2,871
4	43,1721	0,266	64	0,292	43,1720	0,068	27	2,043	43,1721	0,100	37	0,121	43,1721	0,07	30	0,087
5	43,1721	1,000	128	0,143	43,1720	0,111	31	0,841								
6					43,1721	0,157	39	0,130								

Tabela 7.2. Valores de *threshold* (Reator Semi-Batelada Isotérmico)

ℓ	VS	SS					CVPS				
	Escala única	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$	$n=1$	$n=2$	$n=3$	$n=4$	$n=5$
2	0,062	0,00	31,05	35,32	19,06		0,44	0,00	0,00	0,20	
3	0,054	0,00	29,65	0,00	18,09		0,22	0,00	0,00	0,05	
4	0,052	0,00	0,00	0,00	0,41	7,88	0,45	0,00	0,00	0,12	
5	0,044										
6	0,039										

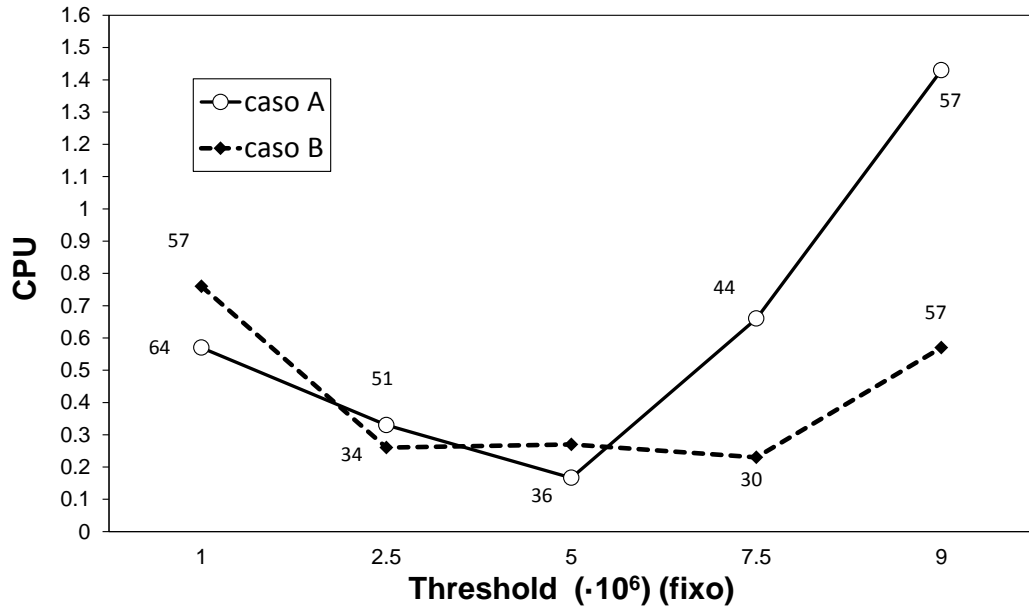


Figura 7.4. Sensibilidade do desempenho computacional em relação a diferentes valores de *threshold*.

7.2. CSTR Não Linear (BALSA-CANTO *et al.*, 2001)

Este exemplo ilustra a aplicação do algoritmo WtA sobre um problema de controle ótimo com três variáveis de controle. O problema consiste em determinar três trajetórias ótimas de controle em um reator CSTR isotérmico de maneira a obter o máximo benefício econômico. O modelo dinâmico é descrito pelas seguintes equações:

$$\dot{x}_1 = u_4 - (u_1 + u_2 + u_4)x_1 - 17,6x_1x_2 - 23x_1x_6u_3 \quad (7.11)$$

$$\dot{x}_2 = u_1 - (u_1 + u_2 + u_4)x_2 - 17,6x_1x_2 - 146x_2x_3 \quad (7.12)$$

$$\dot{x}_3 = u_2 - (u_1 + u_2 + u_4)x_3 - 73x_2x_3 \quad (7.13)$$

$$\dot{x}_4 = -(u_1 + u_2 + u_4)x_4 + 35,2x_1x_2 - 51,3x_4x_5 \quad (7.14)$$

$$\dot{x}_5 = -(u_1 + u_2 + u_4)x_5 + 219x_2x_3 - 51,3x_4x_5 \quad (7.15)$$

$$\dot{x}_6 = -(u_1 + u_2 + u_4)x_6 + 102,6x_4x_5 - 23x_1x_6u_3 \quad (7.16)$$

$$\dot{x}_7 = -(u_1 + u_2 + u_4)x_7 + 46x_1x_6u_3 \quad (7.17)$$

$$\begin{aligned} \dot{x}_8 = & 5,8 \left[(u_1 + u_2 + u_4)x_1 - u_4 \right] - 3,7u_1 - 4,1u_2 + \\ & (u_1 + u_2 + u_4)(23x_4 + 11x_5 + 28x_6 + 35x_7) - 5u_3 - 0,09 \end{aligned} \quad (7.18)$$

com as seguintes condições iniciais:

$$\mathbf{x}_o = [0,1883 \quad 0,2507 \quad 0,0467 \quad 0,0899 \quad 0,7804 \quad 0,1394 \quad 0,1046 \quad 0,000]^T \quad (7.19)$$

O objetivo do problema é maximizar a seguinte função:

$$\max_u x_8(t_f) \quad (7.20)$$

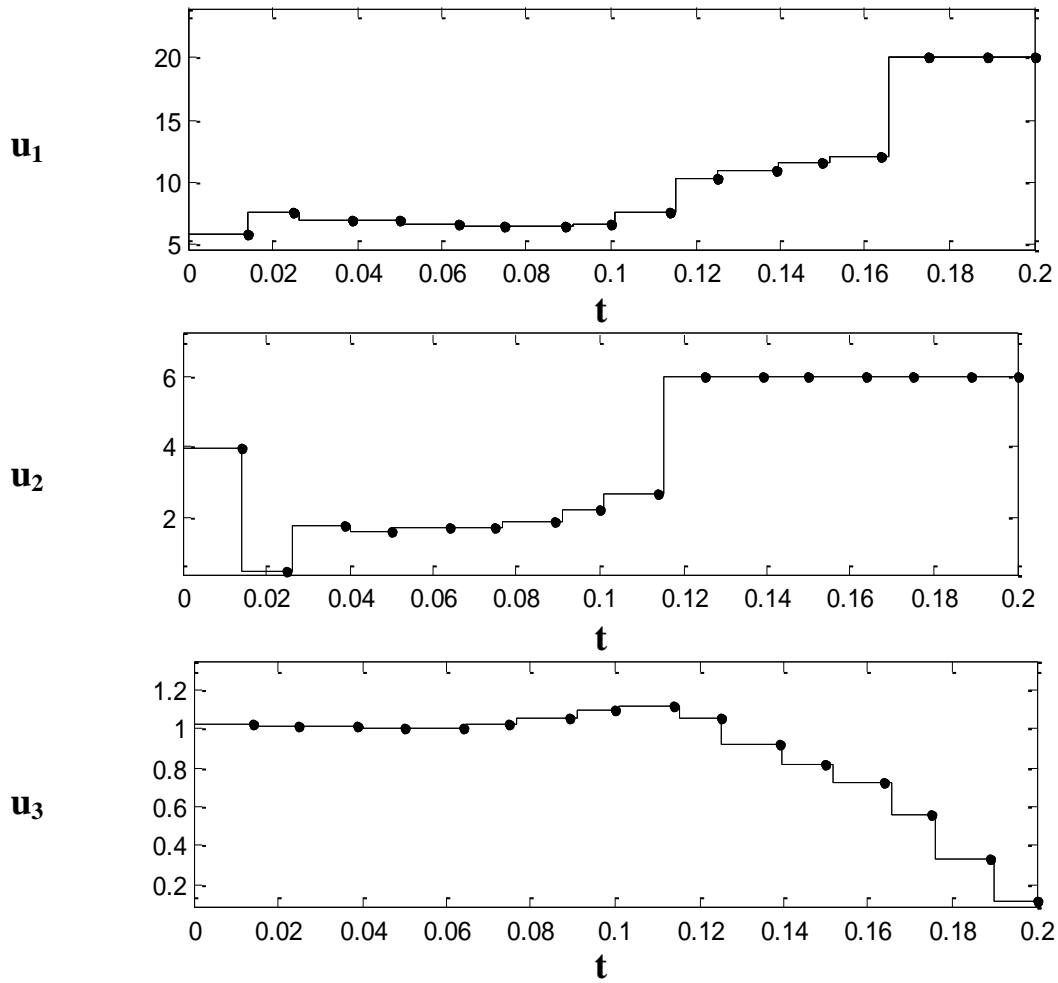
sujeita às seguintes restrições:

$$\begin{cases} 0 \leq u_1 \leq 20 \\ 0 \leq u_2 \leq 6 \\ 0 \leq u_3 \leq 4 \end{cases} \quad (7.21)$$

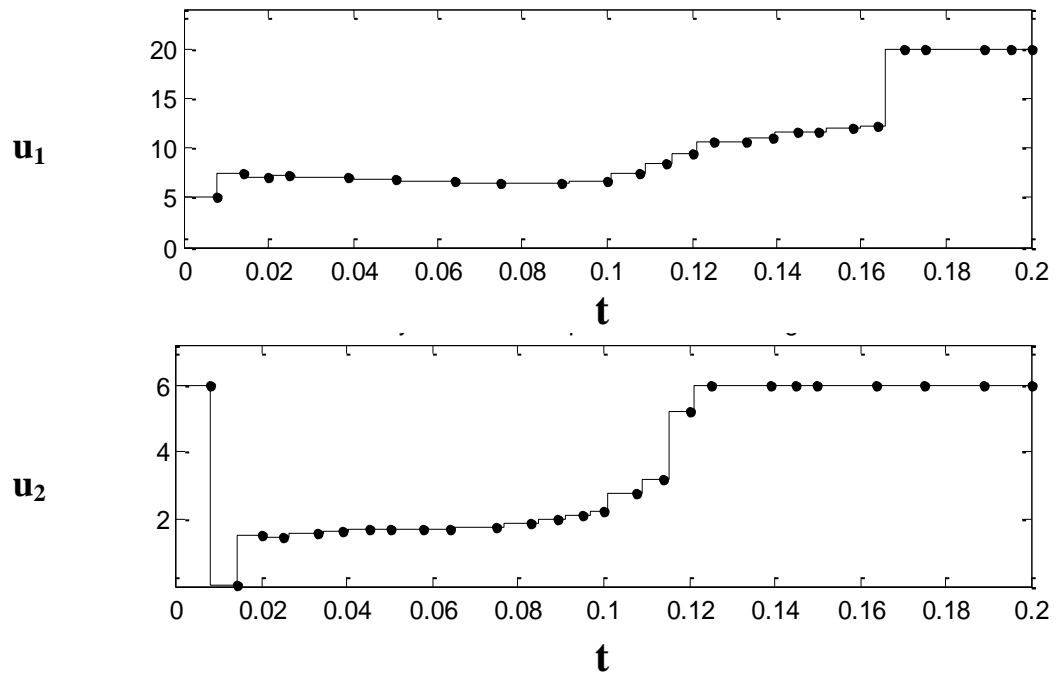
sendo que o instante final de integração, t_f , é fixo em 0,2 e a variável de controle u_4 tem um valor fixo de 6,0. As variáveis de controle são vazões de três correntes de alimentação.

A Figura 7.5 ilustra as trajetórias de controle ótimo a cada iteração ℓ do WtA com a estratégia Sureshrink. Nota-se que em alguns instantes da trajetória, as variáveis de controle u_1 e u_2 atingem as restrições. Claramente, a discretização se concentra nas regiões de maiores descontinuidades, mostrando que o algoritmo WtA capta de maneira eficiente as perturbações que ocorrem em alguns instantes específicos. Observa-se que cada variável de controle apresenta uma evolução diferente, resultando em diferentes números de estágios de discretização. Por exemplo, na iteração 2, o número de estágios de discretização é o mesmo para todas as variáveis, ou seja, 16 estágios. Em seguida, depois de uma etapa da NLP, os números de estágios de discretização das três variáveis são 25, 26 e 25, respectivamente. Já na quarta e última iteração, esses números são 41, 40 e 33 estágios, respectivamente. Nota-se também que a variável de controle u_3 apresenta um perfil de controle mais suave, resultando em menos estágios de discretização. Por outro lado, as variáveis de controle u_1 e u_2 apresentam trajetórias mais descontínuas, resultando em maior aglomeração de estágios em torno dos *switching points*.

A Figura 7.6 ilustra o espectro de detalhes wavelets da evolução do WtA, para cada variável de controle. O espectro de detalhes é ordenado em ordem crescente de níveis wavelets no eixo horizontal. O eixo vertical corresponde aos valores dos coeficientes de detalhes e *thresholds* calculados.



7.5-(a) $\ell = 2$



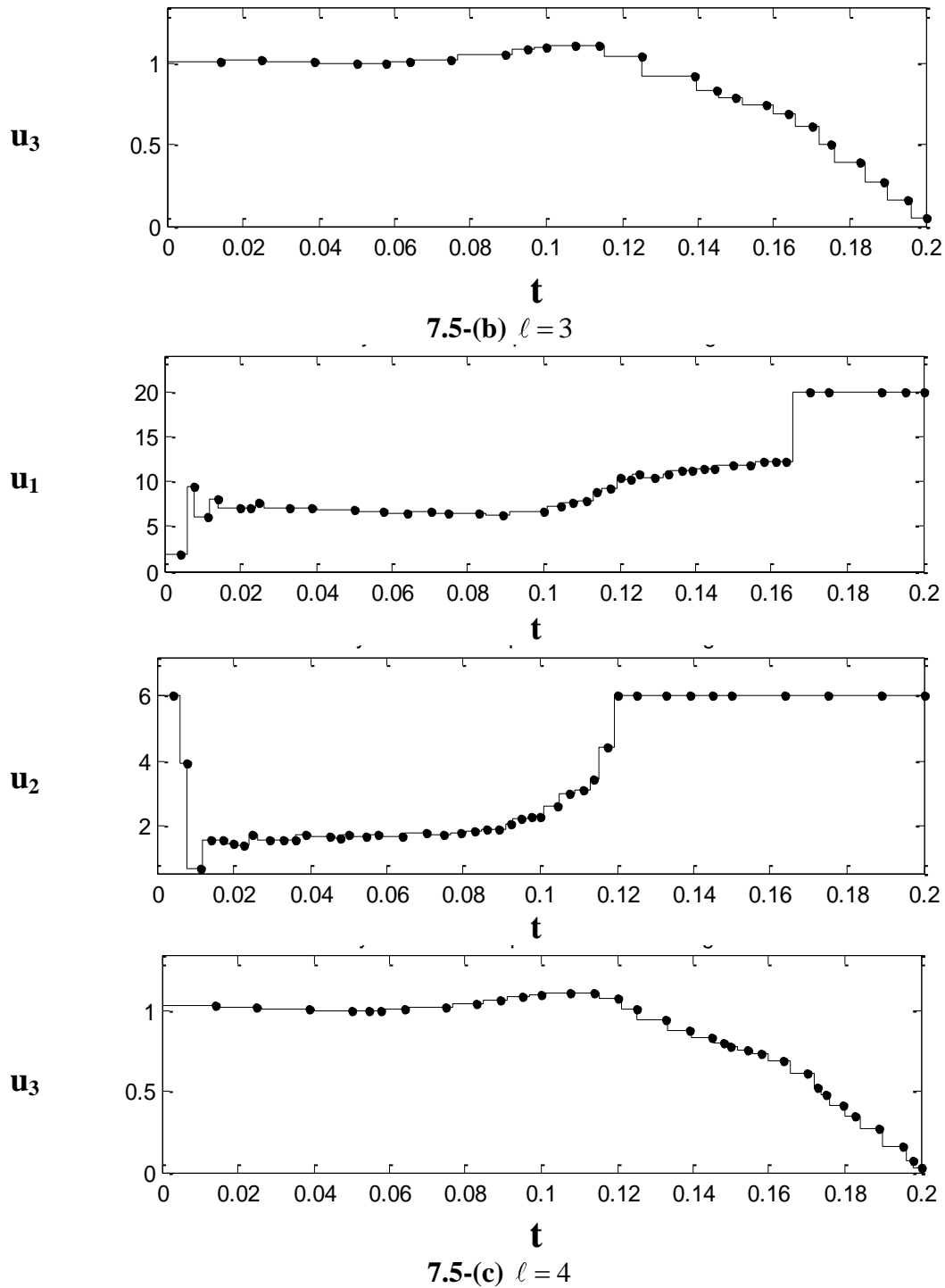
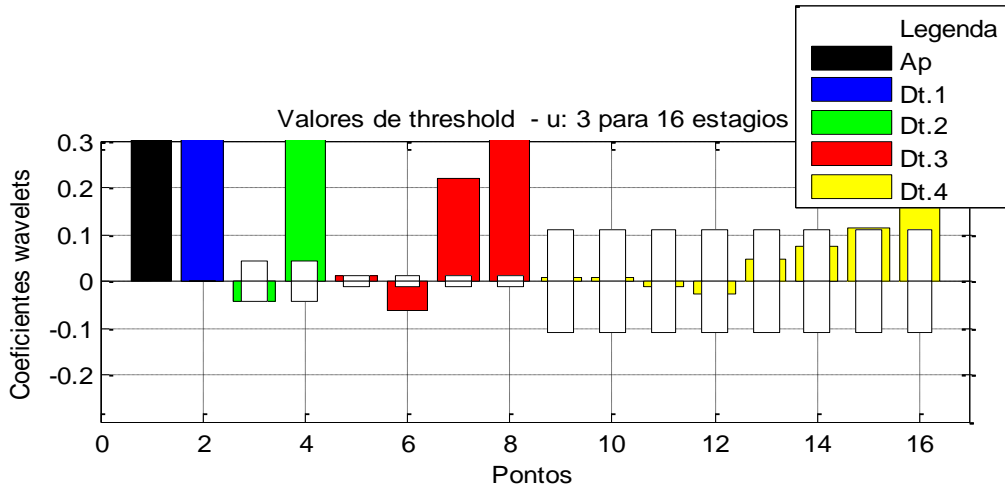
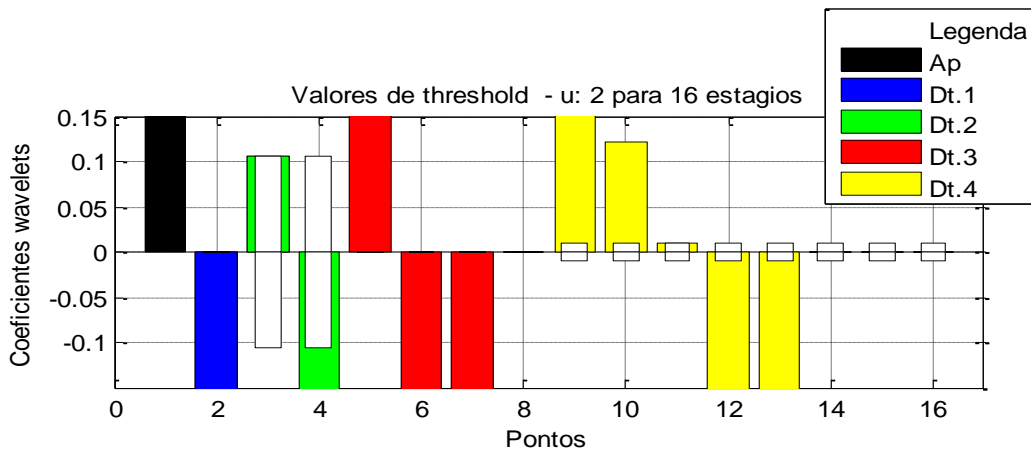
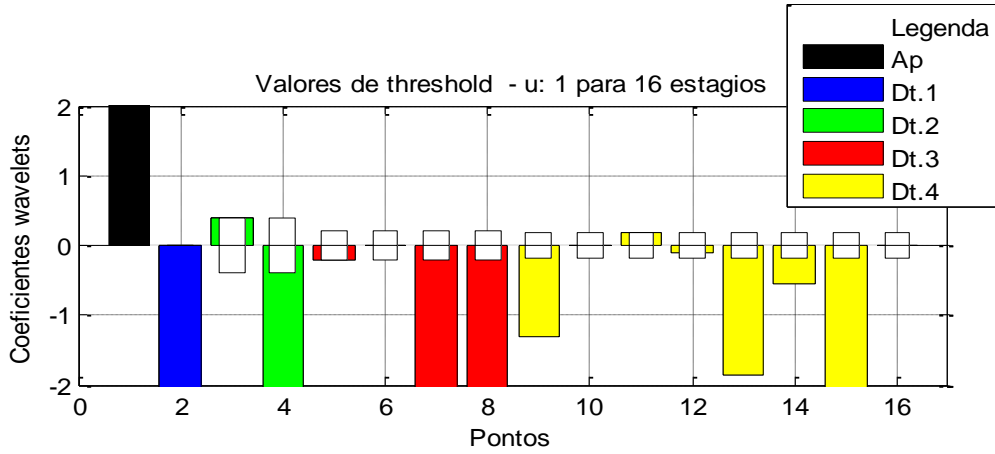


Figura 7.5. Trajetórias de controle em relação ao tempo (a) iteração 2; (b) iteração 3; (c) iteração 4.

Observa-se na Figura 7.6 que os valores de *threshold* variam de acordo com os níveis de resolução wavelets. Note na figura que cada nível de resolução é representado por uma cor diferente. As barras de cor preta representam os coeficientes de aproximação. As barras coloridas representam os coeficientes de detalhes. Já as barras de cor branca representam os valores de *threshold*.



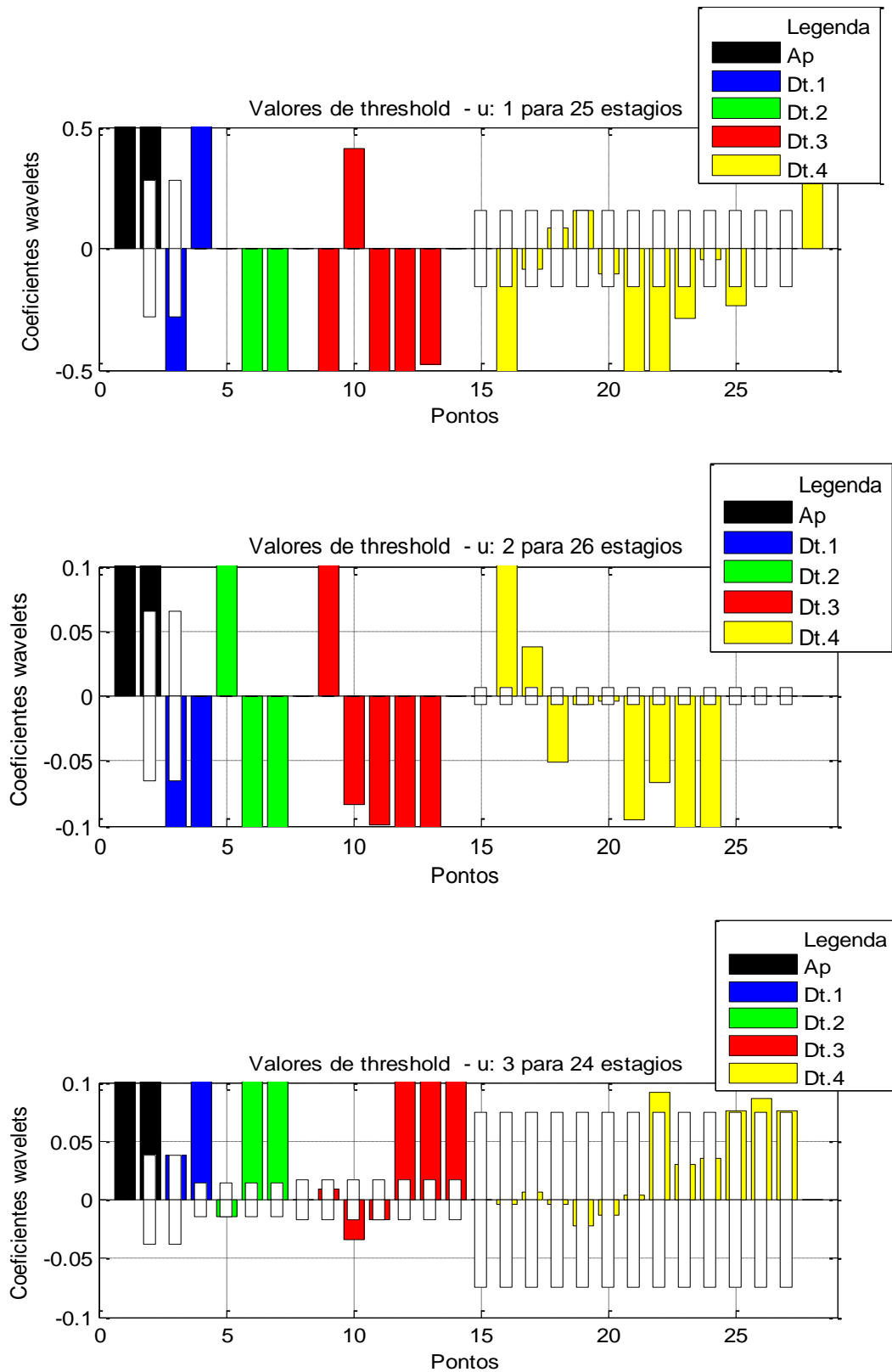


Figura 7.6. Espectro de detalhes wavelets *Sureshrink*.

Antes de comparar os métodos, ilustra-se o histórico do espectro de detalhes do método *Visushrink* na Figura 7.7.

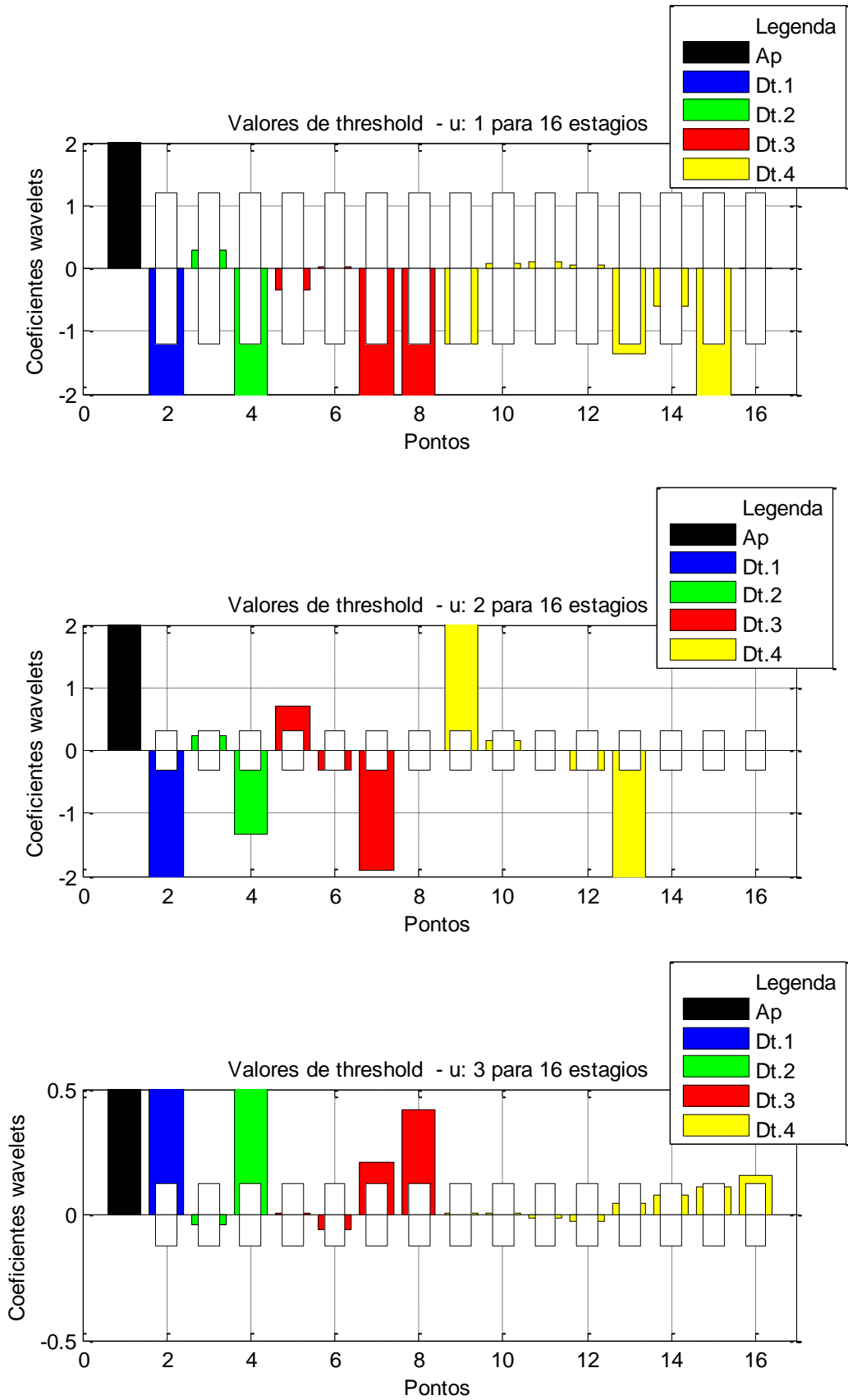


Figura 7.7. Espectro de detalhes wavelets *Visushrink*.

Na Figura 7.7 na estratégia VS um único valor de *threshold* fixo é calculado para todos os níveis de decomposição para cada variável de controle, variando somente com o próprio número de coeficientes.

O desempenho dos métodos é mostrado na Tabela 7.3. Nota-se que de fato o método resulta no CPU mínimo de 0,467. A estratégia VS obteve o pior desempenho com CPU de 3,403. Já o CVPS apresentou um valor intermediário com CPU de 0,66.

É nítido que um menor número de estágios de discretização demandará menor esforço computacional. Porém, o resultado atual do VS mostra que nem sempre isso é verdade. A Tabela 7.3 evidencia que a relação entre o número total de estágios de discretização do VS e o equivalente do equidistante é 0,54. Porém, o desempenho foi consideravelmente inferior. Os resultados também levantam questionamentos, como por exemplo, se há uma estratégia mais vantajosa perante as outras ou se há como configurar a *priori* a estratégia de discretização a ser utilizada antes mesmo de o problema ser solucionado. Com a finalidade de avaliar de maneira mais abrangente o desempenho das diferentes estratégias WtA, foram solucionados 22 problemas de controle ótimo. Os resultados dessa análise são mostrados no próximo tópico.

7.3. Avaliação de Desempenho

7.3.1 Visão Geral

Neste tópico foram avaliados 22 problemas de controle ótimo. Todos os exemplos possuem soluções conhecidas e são referenciados na literatura citada. Com objetivo de estabelecer uma métrica para avaliar o grau de dificuldade dos problemas e sua influência sobre as estratégias utilizadas, foi proposto um fator de dificuldade, definido como:

$$f_d = I_d + f_R \omega_R \quad (7.22)$$

em que o primeiro termo avalia a estrutura do problema quanto a diferentes características, como restrições de trajetória, restrições finais, instante final livre, número de arcos de controle, número de variáveis de estado e número de variáveis de controle. O segundo termo avalia o fator de redução da função objetivo.

O índice de discretização, I_d é definido como:

$$I_d = \left(I_{do} - I_{d,mín} \right) / \left(I_{d,máx} - I_{mín} \right) \quad (7.23)$$

em que:

$$I_{do} = n_u + n_x + n_g \omega_g + n_e \omega_e + n_a + \#(i) \quad (7.24)$$

$$I_{d,mín} = n_u + n_x + n_g \cdot 0 + n_e \cdot 0 + n_a + \#(i) \quad (7.25)$$

$$I_{d,máx} = n_u + n_x + n_g \cdot 1 + n_e \cdot 1 + 5n_u + \#(i) \quad (7.26)$$

onde n_u é o número de variáveis de controle, n_x é o número de variáveis de estado, n_g é o número de restrições de trajetória, n_e é o número de restrições finais, n_a é o número de arcos de controle, ω_g é o fator que avalia a possibilidade de ativação da restrição de trajetória e ω_e é o fator que avalia a possibilidade de ativação da restrição final. O termo $\#(i)$ é 1 se o problema for de instante final livre ou zero caso contrário. Note que $I_{d,mín}$ considera que não há restrições ativas e apenas um arco em cada variável de controle. Já o $I_{d,máx}$ considera que todas as possíveis restrições do problema estão ativas e que também há um número máximo de cinco arcos em cada variável de controle. I_{do} é o próprio índice de discretização do problema que é normalizado, resultando em I_d , conforme escrito na Equação (7.23). A normalização garante que I_d esteja entre 0 e 1. Para todos os problemas avaliados considerou-se que ω_g e ω_e possuem valores iguais a 0,5.

Já o fator de redução, presente na Equação (7.22), é definido como:

$$f_R = (f_{R,o} - f_{R,mín}) / (f_{R,máx} - f_{R,mín}) \quad (7.27)$$

em que

$$f_{R,o} = \left\| \frac{J^{*\langle \ell_{máx} \rangle} - J^{*\langle \ell_{mín} \rangle}}{J^{*\langle \ell_{mín} \rangle}} \right\| \cdot 100 \quad (7.10)$$

e $f_{R,mín}$ e $f_{R,máx}$ são valores fixos obtidos via experimentos numéricos. No conjunto de problemas analisados, observou-se $f_{R,mín} \cong 0,005\%$ e $f_{R,máx} \cong 5\%$, dentro o conjunto de problemas analisados. O fator de ponderação ω_R de f_R na formulação do fator de dificuldade, $f_d = I_d + f_R \omega_R$, é definido como:

$$\omega_R = n_u / n_x \quad (7.28)$$

e define a importância do fator de redução sobre o fator de dificuldade. Logo, este fator é mais importante quanto maior a relação entre o número de variáveis de controle n_u e

estado n_x do problema. A Tabela 7.4 mostra o resultado do f_d para os 22 problemas, em ordem crescente. A Figura 7.8 ilustra os diferentes tipos de problemas analisados.

Conforme mostrado na Tabela 7.4, o fator de dificuldade mínimo foi de 0,01 enquanto o fator máximo foi de 1,11. Observe que não necessariamente este fator cresce linearmente com o número de variáveis de controle ou arcos. Por exemplo, problemas com mais de uma variável de controle encontram-se em uma região intermediária, com f_d em torno de 0,5.

A Tabela 7.5 sumariza o desempenho das três estratégias WtA aplicadas. Para cada caso, disponibiliza-se o número de iterações do WtA, número total de estágios de discretização, custo computacional, além da função objetivo final. Os problemas são ordenados em ordem crescente de f_d . Adicionalmente, destaca-se o mínimo valor de CPU (em relação à CPU obtida no caso equidistante) para cada estratégia, sendo que a cor azul representa a estratégia CVPS, a cor preta representa a estratégia SS e a cor laranja a estratégia VS. A estrutura do problema de otimização de cada exemplo e a trajetória ótima de controle (obtida com o método mais eficiente) para cada problema são visualizadas no Tópico 7.4, no final deste tópico.

A Figura 7.10 mostra que para todo o conjunto de problemas, o VS é indicado para 55% dos casos, enquanto o CVPS para apenas 18%. Em princípio, nota-se que para valores baixos de f_d o VS apresenta menores valores de CPU. Já o SS apresenta valores menores para maiores valores de f_d . Essa característica é mais evidente na Figura 7.10 onde é ilustrado o desvio relativo entre a CPU do SS e do VS que apresentam desempenhos simétricos. Claramente, para valores menores de f_d esse desvio é positivo e tende a se tornar negativo para valores elevados de f_d .

A Tabela 7.5 evidencia que a estratégia CVPS não aparenta seguir alguma tendência dependente do valor de f_d . Porém, observa-se que o CVPS obtém melhores desempenhos para problemas com múltiplas variáveis de controle, como os problemas 9, 12 e 13. No entanto, ao avaliar a CPU média de cada estratégia, observa-se que o VS apresenta o menor valor, em torno de 0,48, enquanto o SS e CVPS apresentam os valores 0,66 e 0,53, respectivamente.

Tabela 7.3. Desempenho das diferentes estratégias (CSTR não linear)

ℓ	Equidistante						VS					
	J^*	CPU	$n_{s,1}$	$n_{s,2}$	$n_{s,3}$	$\phi(\cdot 10^6)$	J^*	CPU	$n_{s,1}$	$n_{s,2}$	$n_{s,3}$	$\phi(\cdot 10^6)$
1	20,0762	0,028	8	8	8	-	20,0762	0,028	8	8	8	-
2	20,0886	0,127	16	16	16	617,64	20,0886	0,127	16	16	16	617,64
3	20,0904	0,520	32	32	32	89,59	20,0900	0,370	22	25	21	69,68
4	20,0906	0,582	64	64	64	9,94	20,0901	0,394	32	40	24	4,92
5	20,0907	1	128	128	128	4,96	20,0906	2,237	56	59	34	20,48
6							20,0907	3,403	78	82	48	0.81

ℓ	SS						CVPS					
	J^*	CPU	$n_{s,1}$	$n_{s,2}$	$n_{s,3}$	$\phi(\cdot 10^6)$	J^*	CPU	$n_{s,1}$	$n_{s,2}$	$n_{s,3}$	$\phi(\cdot 10^6)$
1	20,0762	0,028	8	8	8	-	20,0762	0,028	8	8	8	-
2	20,0886	0,127	16	16	16	617,64	20,0886	0,127	16	16	16	617,64
3	20,0906	0,407	25	26	25	98,59	20,0906	0,327	25	26	24	98,65
4	20,0907	0,467	41	40	33	0,88	20,0907	0,660	41	43	33	0,91

Tabela 7.4. Fator de dificuldade calculado para 22 problemas (em ordem crescente)

		nu	nx	ng	ne	#i	na	fR	nu/nx	Id	fd
1	Batch Reactor Problem (CIZNIAR et al., 2005)	1	2	0	0	0	1	0,03	0,50	0,00	0,01
2	Temperature Control (RUTQUIST e EDVALL, 2010)	1	2	0	1	0	1	0,10	0,50	0,10	0,15
3	Van der Pol Oscillator (VASSILIADIS et al., 1994a)	1	3	0	0	0	2	0,01	0,33	0,25	0,25
4	Linear Bang Bang Control (LUUS, 2002)	1	3	0	0	0	2	0,05	0,33	0,25	0,26
5	Pressure-constrained Batch Reactor (HUANG, 2002)	1	3	1	0	0	2	0,00	0,33	0,30	0,30
6	Singular Control 5 (LUUS, 2002)	1	3	0	0	0	2	0,18	0,33	0,25	0,31
7	Lee & Ramirez Bioreactor Q = 2.5 (BALSA-CANTO et al, 2001)	2	9	0	0	0	5	0,01	0,22	0,38	0,38
8	Time Free Bang Bang Control (LIANG et al. 2003)	1	2	0	2	1	2	0,15	0,50	0,33	0,41
9	Non-linear CSTR Four Control Variables (BALSA-CANTO et al., 2001)	4	8	0	0	0	10	0,10	0,50	0,38	0,43
10	Drug Displacement Problem (LUUS, 2002)	1	2	0	2	1	2	0,24	0,50	0,33	0,45
11	Continuous State Constraint Problem (JACOBSON e LELE, 1969)	1	3	1	0	0	3	0,01	0,33	0,10	0,50
12	Lee & Ramirez Bioreactor Q = 0 (BALSA-CANTO et al., 2001)	2	8	0	0	0	4	0,00	0,25	0,50	0,50
13	Non-linear CSTR Three Control Variables (BALSA-CANTO et al., 2001)	3	8	0	0	0	9	0,01	0,38	0,50	0,50
14	CSTR Reactor (SRINAVASAN et al., 2003a)	1	3	0	2	0	3	0,02	0,33	0,50	0,51
15	Batch Fermentator Penicilin (BANGA et al., 2005)	1	4	3	0	0	3	0,04	0,25	0,50	0,51
16	The Mixed Catalyst Problem (BELL e SARGENT, 2000)	1	3	0	0	0	3	0,08	0,33	0,50	0,53
17	Benoit Chachuat Example (CHACHUAT, 2006)	1	3	0	2	0	3	0,10	0,33	0,50	0,53
18	Singular Control 4 (LUUS, 2002)	1	5	0	0	0	3	0,17	0,20	0,50	0,53
19	Quadruple Integral (RUTQUIST e EDVALL, 2010)	1	4	0	4	1	4	0,04	0,25	0,63	0,64
20	Singular Control 2 (LUUS, 2002)	1	3	0	0	0	4	0,59	0,33	0,50	0,89
21	Singular Control 3 (LUUS, 2002)	1	4	0	0	0	5	0,24	0,25	0,75	1,08



Figura 7.8. Características dos problemas analisados (restrições de trajetória e variáveis de controle).

Tabela 7.5. Desempenho das estratégias SS, VS e CVPS

	fd	SS			VS			CVPS			Equidistante			
		CPU	l	ns,t	CPU	l	ns,t	CPU	l	ns,t	l	ns,t	J*	
1	Batch Reactor Problem (CIZNIAR et al., 2005)	0,12	6	23	0,06	6	53	1,9	6	52	5	128	0,0695	
2	Temperature Control (RUTQUIST e EDVALL, 2010)	1,64	7	93	0,28	7	32	0,69	7	52	5	128	203,8656	
3	Van der Pol Oscillator (VASSILIADIS et al., 1994a)	2,94	6	76	0,1	6	29	1,08	5	118	5	128	2,9647	
4	Linear Bang Bang Control (LUUS, 2002)	1,68	9	82	0,3	14	99	0,28	6	57	4	64	41,3488	
5	Pressure-constrained Batch Reactor (HUANG, 2002)	0,25	5	35	0,18	4	27	0,25	5	39	4	64	11,7270	
6	Singular Control 5 (LUUS, 2002)	0,27	5	47	0,18	5	35	0,21	5	40	5	128	0,7540	
7	Lee & Ramirez Bioreactor Q = 2.5 (BALSA-CANTO et al, 2001)	0,39	5	120	0,36	5	123	0,55	6	140	5	256	5,7571	
8	Time Free Bang Bang Control (LIANG et al. 2003)	0,98	4	53	0,41	4	31	0,35	4	29	5	128	30,1400	
9	Non-linear CSTR Four Control Variables (BALSA-CANTO et al., 2001)	0,61	6	518	0,5	8	454	0,28	5	324	4	256	21,8169	
10	Drug Displacement Problem (LUUS, 2002)	0,94	5	55	0,22	5	27	0,53	5	38	4	64	221,2858	
11	Continuous State Constraint Problem (JACOBSON e LELE, 1969)	0,16	5	53	0,03	7	25	0,69	6	97	5	128	0,1728	
12	Lee & Ramirez Bioreactor Q = 0 (BALSA-CANTO et al., 2001)	0,50	0,54	4	81	0,59	5	87	0,23	5	65	4	128	6,1515
13	Non-linear CSTR Three Control Variables (BALSA-CANTO et al., 2001)	0,50	0,46	4	114	3,4	6	208	0,46	5	113	5	384	20,0907
14	CSTR Reactor (SRINAVASAN et al., 2003a)	0,51	0,1	4	37	0,15	6	39	0,07	4	30	5	128	0,4317
15	Batch Fermentator Penicilin (BANGA et al., 2005)	0,51	1,248	4	39	0,94	4	38	0,95	5	40	6	128	88,0364
16	The Mixed Catalyst Problem (BELL e SARGENT, 2000)	0,53	0,1	4	32	0,15	4	40	0,16	4	27	5	128	0,0480
17	Benoit Chachuat Example (CHACHUAT, 2006)	0,53	0,29	5	64	0,32	6	51	0,29	6	49	5	128	4,5859
18	Singular Control 4 (LUUS, 2002)	0,53	0,6	5	64	0,49	4	28	0,86	4	42	5	128	0,1194
19	Quadruple Integral (RUTQUIST e EDVALL, 2010)	0,64	0,12	3	32	0,2	4	34	0,16	4	35	5	128	1,0000
20	Singular Control 2 (LUUS, 2002)	0,89	0,15	5	51	0,26	5	46	0,22	5	53	5	128	0,2684
21	Singular Control 3 (LUUS, 2002)	1,08	0,2	6	62	0,28	7	63	0,5	6	36	5	128	1,2523
22	Park and Ramirez Bioreactor (BALSA-CANTO et al., 2001)	1,11	0,86	20	91	1,23	22	80	0,93	10	86	5	128	32,6860

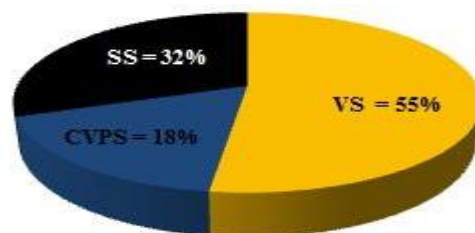


Figura 7.9. Desempenho das estratégias (percentual de sucesso de cada estratégia frente ao número total de problemas).

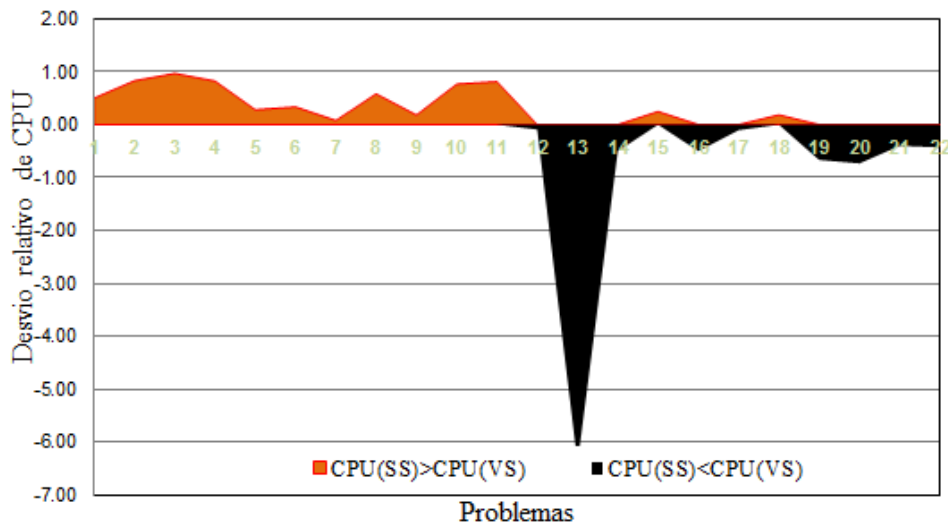


Figura 7.10. Desempenho das estratégias (confronto entre SS e VS).

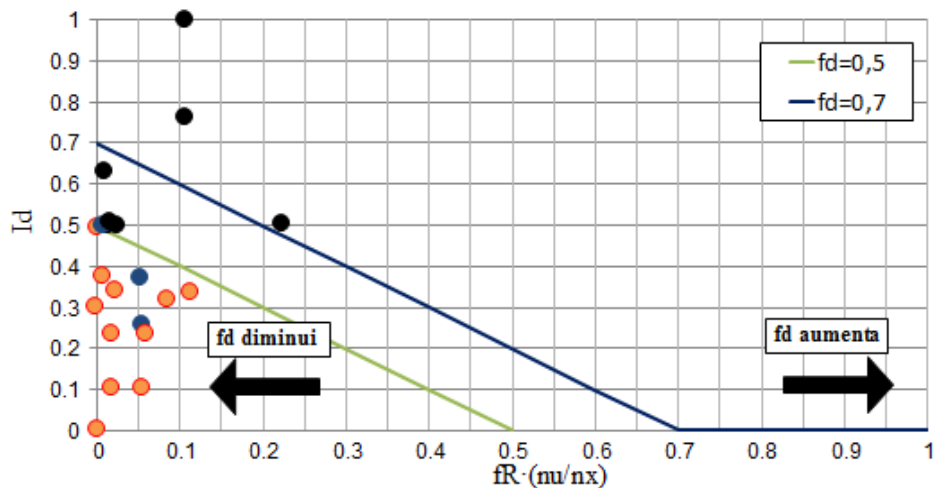


Figura 7.11. Zonas de fator de dificuldade, em função do índice de discretização e fator de redução.

Os resultados induzem-nos a dividir o *range* de f_d em duas regiões: $f_d < 0,5$ e $f_d \geq 0,5$, conforme ilustrado na Figura 7.11. Na Figura 7.11 divide-se o fator de dificuldade em duas regiões: $f_d < 0,5$ e $f_d \geq 0,5$, indicadas pela reta de cor verde. No eixo x computam-se os valores do fator de redução multiplicados pela razão das variáveis de controle e estado (ver Equação 7.22). Já no eixo y computam-se os valores do índice de discretização. Os pontos coloridos representados na figura ilustram os valores de f_d obtidos para cada problema (as cores seguem a legenda original da

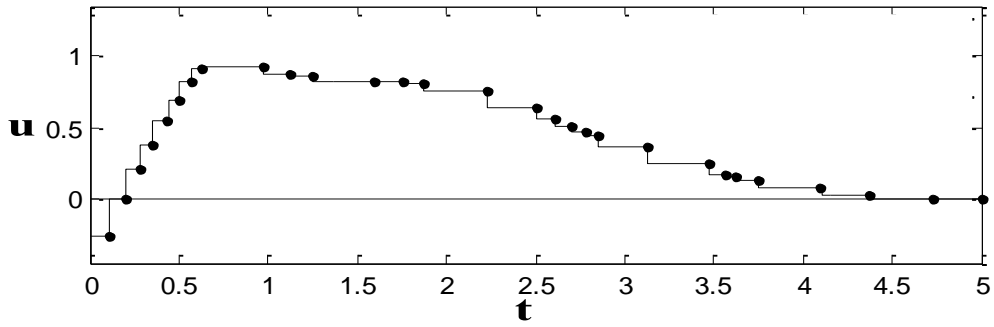
Tabela 7.5). É possível observar um aglomerado de pontos de cor laranja situados na zona de baixo fator de dificuldade 0.5. Já os pontos pretos (SS) encontram-se na zona de elevado fator de dificuldade (acima de 0.5).

Uma explicação plausível para o insucesso do SS para zonas de baixo f_d é o fato de essa região concentrar problemas com trajetórias de controle suaves, ou com poucas ocorrências de descontinuidades, conforme sumariza o Tópico 7.4.

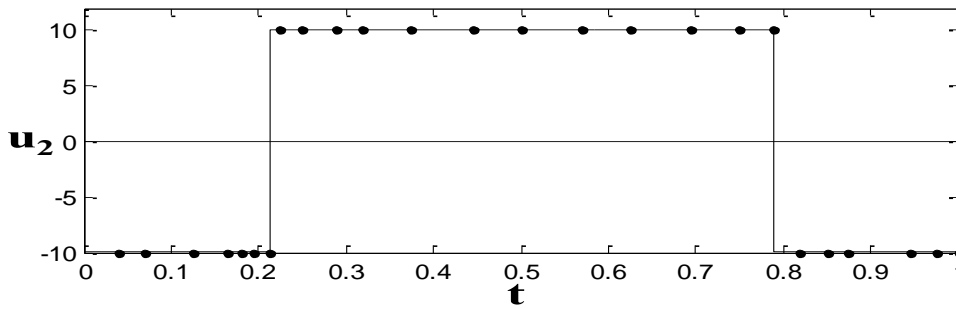
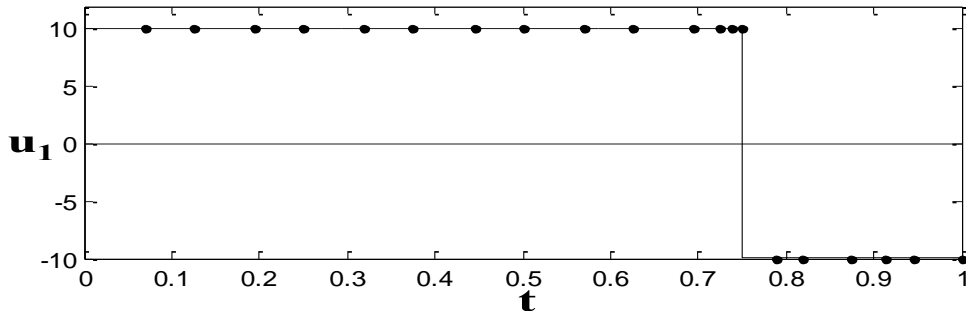
Uma desvantagem do SS é que não funciona bem quando a decomposição wavelets resulta em coeficientes de detalhes com valores próximos a zero, gerando vetores com elevado nível de esparsividade. Geralmente, para conjunto de dados bem comportados, ou que apresenta pouca ocorrência de descontinuidades, níveis com baixo grau de esparsividade podem caracterizar o conjunto de dados original (DONOHO e JOHNSTONE, 1995). Em outras palavras, quando a parcela mais importante está concentrada em níveis wavelets mais baixos, o SS tende a apresentar um desempenho indesejável, pois nesses níveis há coeficientes de detalhes insuficientes para estimação do *threshold*.

Nota-se que os problemas com baixo f_d apresentam de fato trajetórias de controle com número baixo de arcos de controle. O Tópico 7.4 também mostra que problemas multi-variáveis, como o problema 13, que apresenta três variáveis de controle, possui uma trajetória contínua, o que pode prejudicar o desempenho do SS nesses casos. Logo, problemas multi-variáveis não garantem necessariamente melhor desempenho do SS. Essa característica é mais evidente na Figura 7.12, que disponibiliza as trajetórias de controle dos três piores desempenhos do SS.

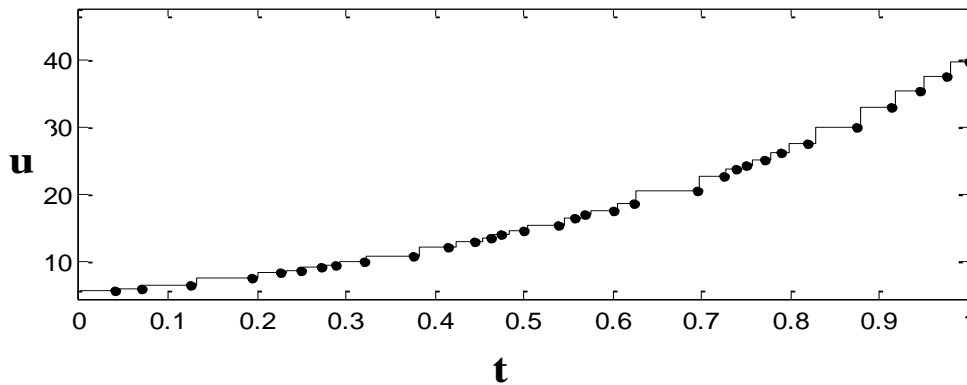
Em outra análise da Tabela 7.5 nota-se que o VS enfrenta dificuldades em problemas com trajetórias de controle altamente descontínuas. Porém, tal fato só foi observado nos problemas com valores de $f_d > 0,5$. Observa-se, por exemplo, que em alguns problemas com dois arcos (problemas 3, 4,5 e 6) o VS apresenta bons resultados. Já na zona de elevado f_d , os problemas possuem três ou mais arcos de controle, o que impõe dificuldades. Pelo fato de o VS estimar um valor de *threshold* fixo, tal estratégia tende a suavizar excessivamente o conjunto de dados, preservando detalhes que poderiam ser naturalmente eliminados



(a) (Problema3) Van der Pol Oscillator (VASSILIADIS *et al.*, 1994a)



(a) (Problema4) Linear Bang Bang Control (LUUS, 2002)



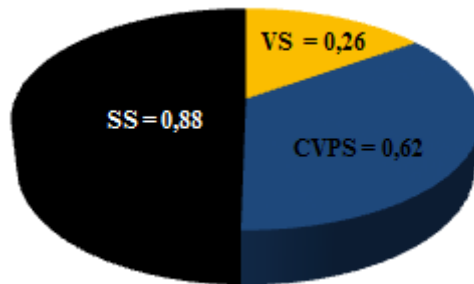
(c) (Problema2) Temperature Control (RUTQUIST e EDVALL, 2010)

Figura 7.12. Trajetória de controle dos três piores desempenhos do SS.

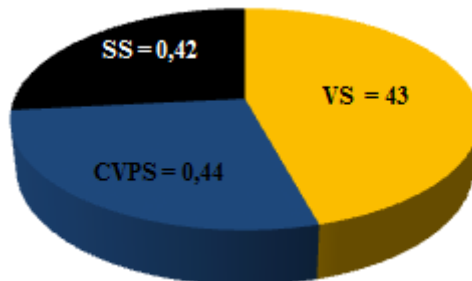
O CVPS é a estratégia sugerida neste trabalho para balancear os dois extremos VS e SS. Conforme introduzido no Capítulo 6, a função objetivo do CVPS tende a equilibrar o efeito da compressão, assim como o efeito da parametrização. Logo, espera-se que o algoritmo WtA seja menos vulnerável que o SS para regiões de baixo f_d . Observa-se, na Figura 7.13, de fato, que para essas situações, o CVPS apresenta a segunda menor CPU médio, com valor de 0,62, perdendo apenas para o VS, que apresenta CPU média de 0,26. O mesmo ocorre nas regiões de elevado f_d onde o VS apresenta também a segunda menor CPU médio, com valor de 0,44, enquanto o SS apresenta um valor de 0,42.

O maior destaque da aplicação do CVPS é observado nos problemas com múltiplas variáveis de controle. Justamente por ser um critério que pondera a compressão e parametrização, aparenta ser mais estável quando há múltiplas trajetórias com diferentes características, envolvidas no processo de *thresholding*.

Na Figura 7.13 ilustra-se a CPU média das três estratégias, para a região de baixo $f_d < 0,5$ e elevado $f_d \geq 0,5$.



(a) CPU média para $f_d < 0,5$



(a) CPU média para $f_d \geq 0,5$

Figura 7.13. CPU média das três estratégias.

7.3.2 Observações Complementares

Na Tópico 7.4 também é possível mensurar a razão de compressão média do algoritmo. Essa métrica é calculada dividindo-se o número de estágios da estratégia WtA mais eficiente pelo número de estágios da discretização equidistante. A Figura 7.14 ilustra o valor obtido para cada problema.

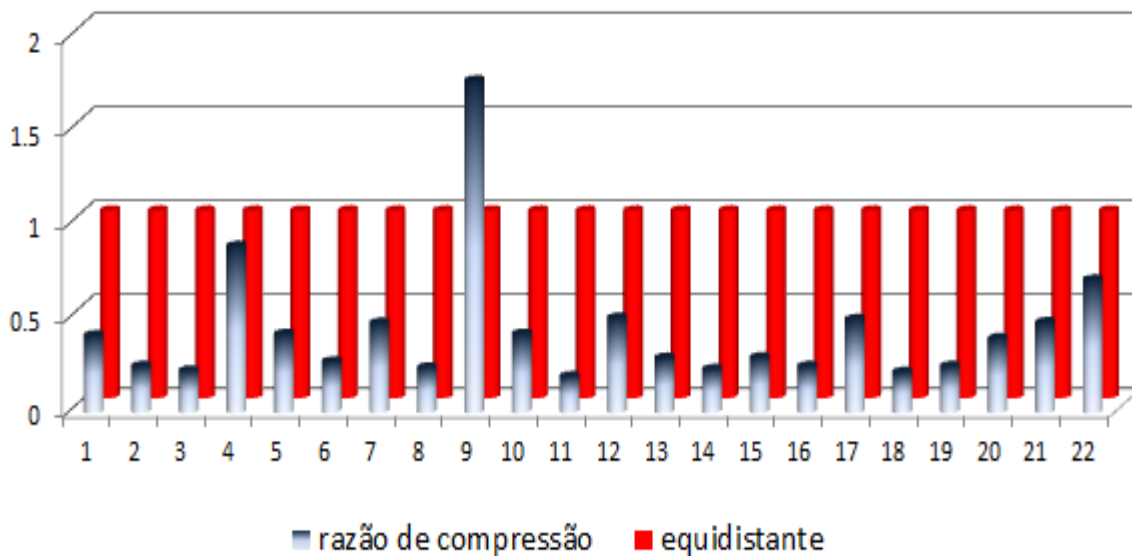


Figura 7.14. Razão de compressão.

Observa-se que para a maioria dos problemas foi possível obter uma elevada razão de compressão, sem comprometer a precisão da solução, de acordo com o critério de convergência utilizado. Dentre esses 22 problemas verificados a média de razão de compressão foi de 0,44.

Percebe-se no Tópico 7.4, porém, que nem sempre o menor número de estágios significa menor CPU. Em alguns exemplos foi detectada a situação de existir uma estratégia *thresholding* mais eficiente, porém com maior número de estágios. Tal comportamento se explica pelo fato de não só o número de estágios, mas também o tamanho de cada estágio ser impactante, o que pode beneficiar (ou atrapalhar) a rotina de otimização implementada, mas influenciar negativamente (ou positivamente) a etapa de integração.

A Figura 7.15, por exemplo, ilustra o espectro final dos detalhes wavelets após a solução do problema 9 que apresenta 4 variáveis de controle. Mesmo apresentando significativa redução de CPU, a razão de compressão ultrapassou a unidade conforme mostra a Figura 7.14. Em outras palavras, foi possível reduzir o custo computacional,

porém, um número maior de estágios (324) foi necessário para obtenção de tal resultado.

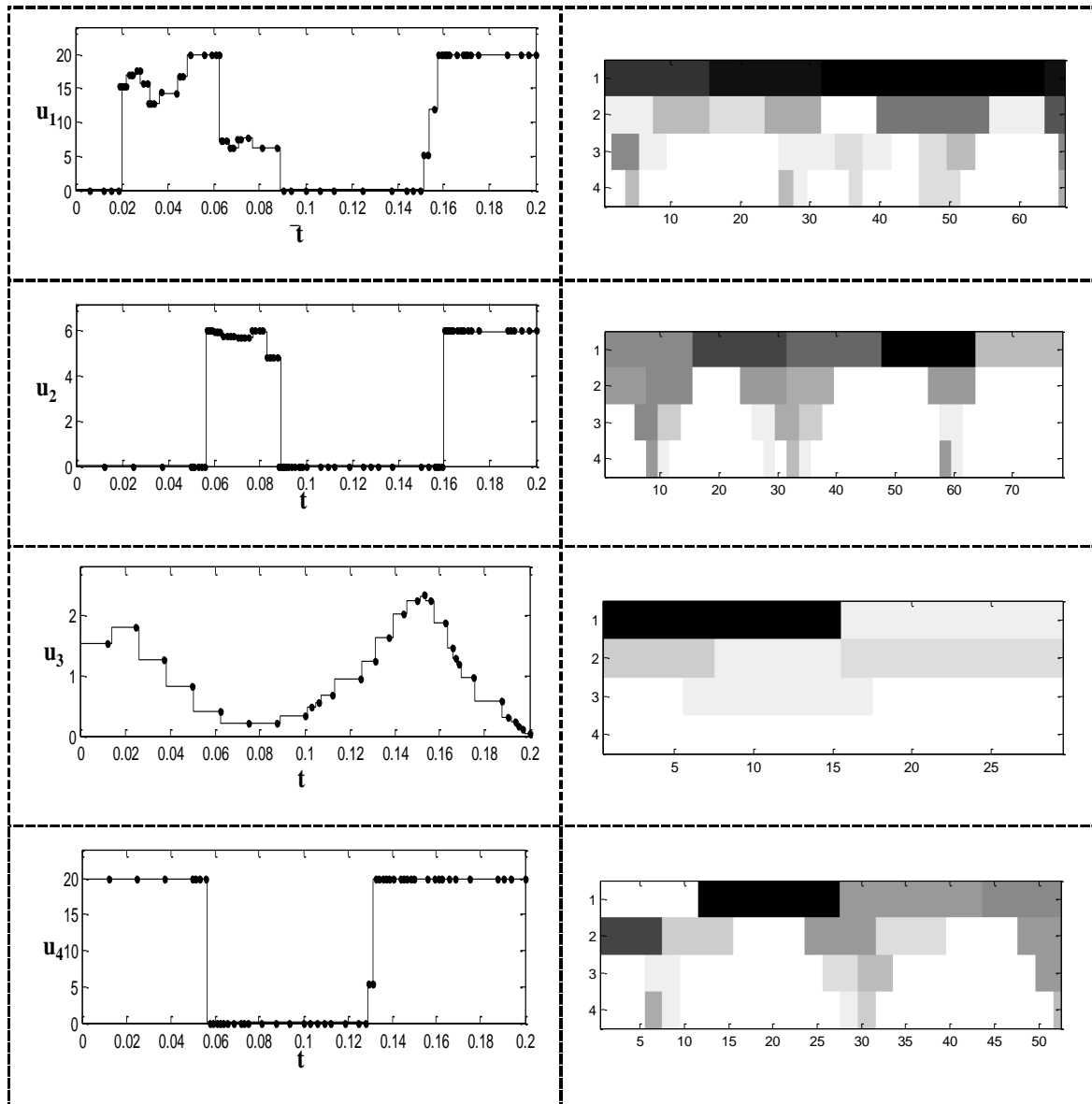


Figura 7.15. Espectro de detalhes do problema 9.

Na Figura 7.15 observa-se que as variáveis de controle 1 e 2 apresentam trajetórias bastante descontínuas, contribuindo para uma discretização significativamente refinada nas regiões de mudança de arcos. Porém, o que caracteriza a elevada razão de compressão são fatores que possivelmente dependem de outros fatores também, como tolerâncias dos algoritmos utilizados para integração numérica e otimização, assim como o critério de convergência utilizado. Logo, todos esses ajustes podem intervir no número de estágios finais.

De acordo com os resultados apresentados, afirma-se que o fator de dificuldade sugerido é uma métrica interessante para prever qual estratégia WtA é adequada para um determinado problema de controle ótimo. No entanto, o f_d só é obtido após o problema ser solucionado, pois é necessário computar o número de arcos, assim como o fator de redução f_R . Uma estratégia para contornar essas duas limitações é estimar esses dois valores previamente a partir da solução parcial do problema (com poucos estágios). A experiência numérica obtida até então evidencia que a maior parcela da redução da função objetivo ocorre nas primeiras iterações do WtA. Para a maioria dos problemas a variação relativa da função objetivo da primeira à segunda iteração do WtA corresponde a aproximadamente 70~80% do f_R . Já o número de arcos pode ser razoavelmente estimado a partir da segunda ou terceira iteração. Por consequência, seria possível estimar qual a melhor política de *thresholding* a ser utilizada.

7.4 Resultados: Modelos e Trajetórias de Controle

1. Batch Reactor Problem (CIZNIAR et al., 2005)

$$J = \text{máx} [x_2(t_f)]$$

$$\dot{x}_a = -k_a \cdot (x_a)^2$$

$$\dot{x}_b = k_a \cdot (x_a)^2 - k_b \cdot (x_b)$$

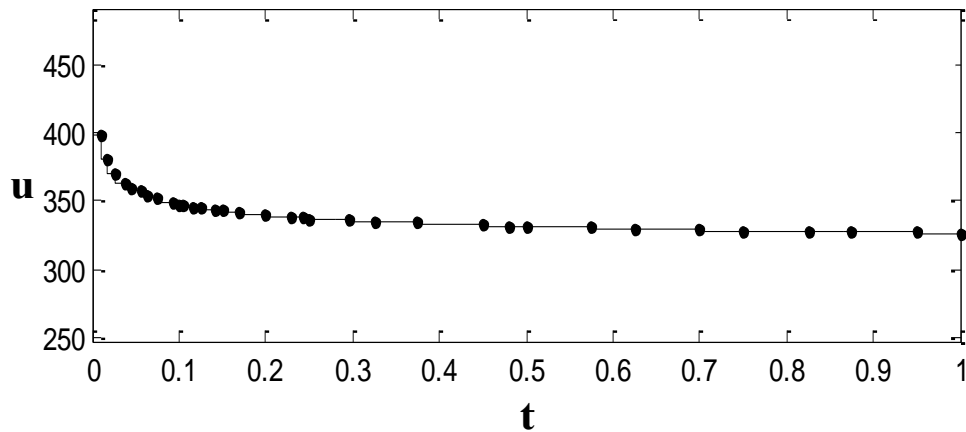
$$k_a = 4000 \cdot e^{-2000/u}$$

$$k_b = 620000 \cdot e^{-5000/u}$$

$$\mathbf{x}_0 = [1 \ 0]^T$$

$$298 \leq u \leq 398$$

$$t \in [0 \ 1]$$



2. Temperature Control (RUTQUIST e EDVALL, 2010)

$$J = \min [x_2(t_f)]$$

$$\dot{x}_1 = -2x_1 + u$$

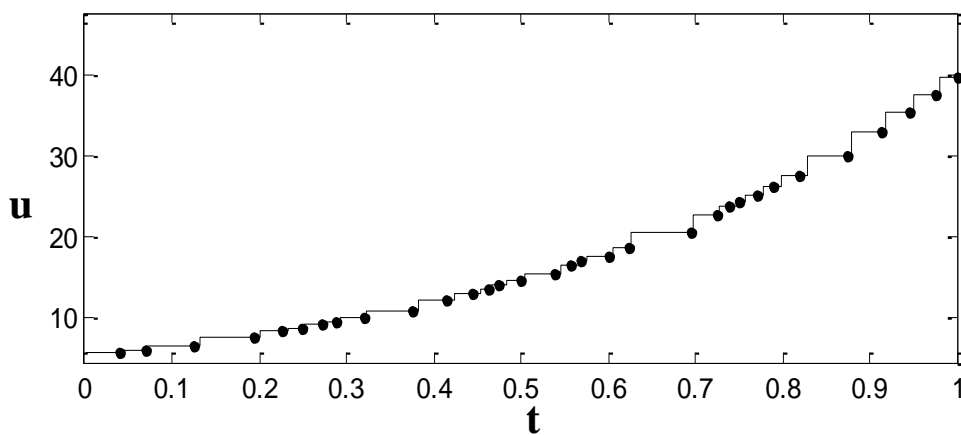
$$\dot{x}_2 = 0,5u^2$$

$$\mathbf{x}_0 = [0 \ 0]^T$$

$$0 \leq u \leq 50$$

$$x_1(t_f) \leq 10$$

$$t \in [0 \ 1]$$



3. Van der Pol Oscillator (VASSILIADIS *et al.*, 1994a)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_a = (1 - x_b^2)x_a - x_b + u$$

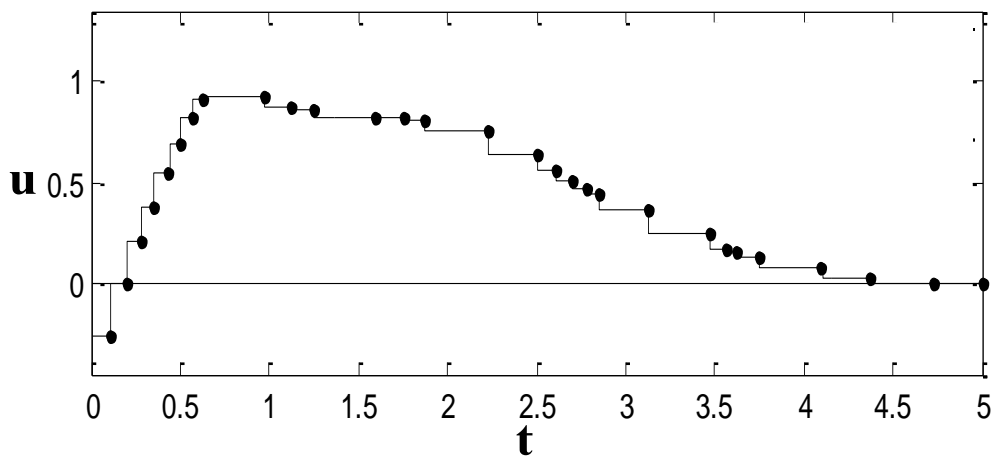
$$\dot{x}_b = x_a$$

$$\dot{x}_c = x_a^2 + x_b^2 + u^2$$

$$\mathbf{x}_0 = [0 \ 1 \ 0]^T$$

$$-0.3 \leq u \leq 1$$

$$t \in [0 \ 5]$$



4. Linear Bang Bang Control (LUUS, 2002)

$$J = \min \int (-6x_1 - 12x_2 + 3u_1 + u_2) dt$$

$$\dot{x}_1 = u_2$$

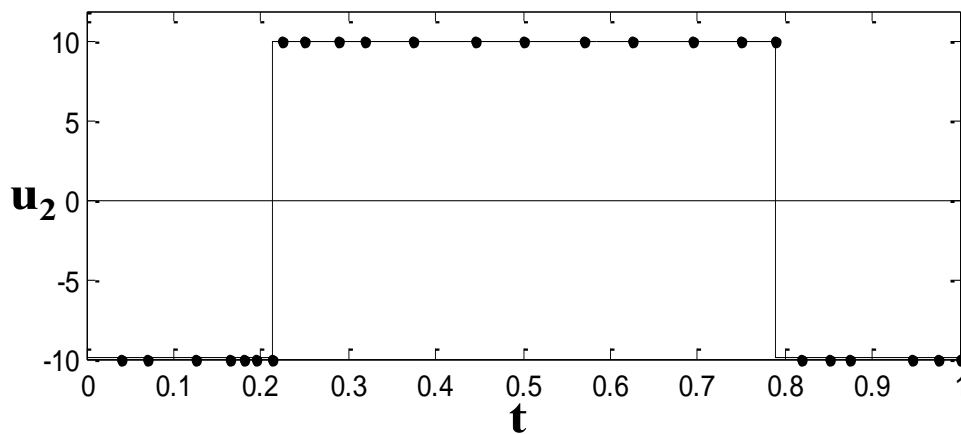
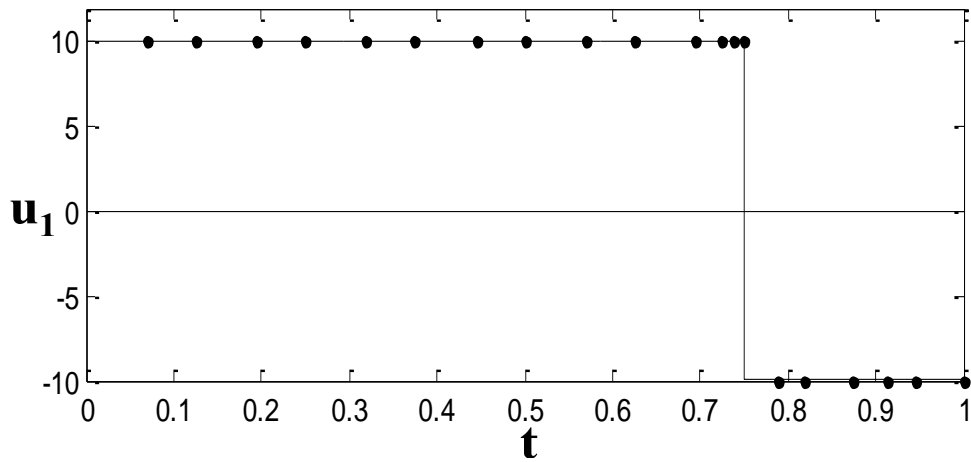
$$\dot{x}_2 = -x_1 + u_1$$

$$\mathbf{x}_o = [1 \ 0]^T$$

$$-10 \leq u_1 \leq 10$$

$$-10 \leq u_2 \leq 10$$

$$t \in [0 \ 1]$$



5. Pressure-constrained Batch Reactor (HUANG, 2002)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_a = -k_a(x_a) + k_b(x_b)^2 + (u/V) - k_c x_a x_b$$

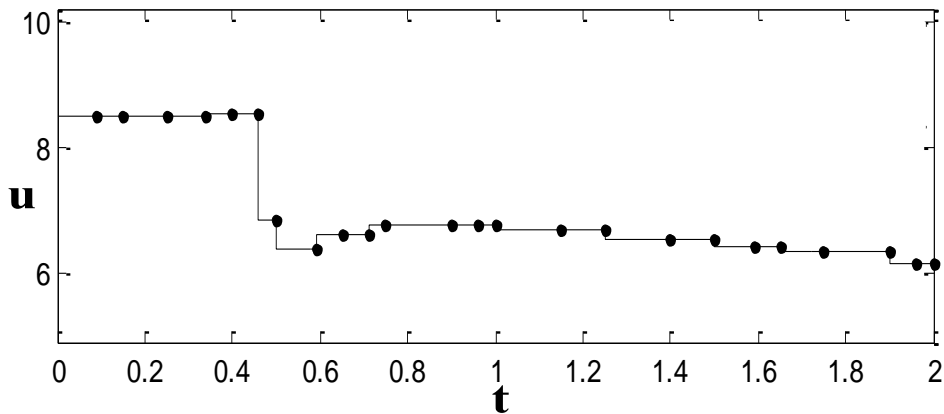
$$\dot{x}_b = k_a(x_a) - k_b(x_b)^2 - k_c x_a x_b$$

$$\dot{x}_c = k_c x_a x_b$$

$$\mathbf{x}_0 = [1 \ 0]^T$$

$$\begin{cases} RT(x_1(t) + x_2(t) + x_3(t)) < 340 \\ 0 \leq u \leq 8,5 \end{cases}$$

$$t \in [0 \ 2]$$



6. Singular Control 5 (LUUS, 2002)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_a = -k_a(x_a) + k_b(x_b)^2 + (u/V) - k_c x_a x_b$$

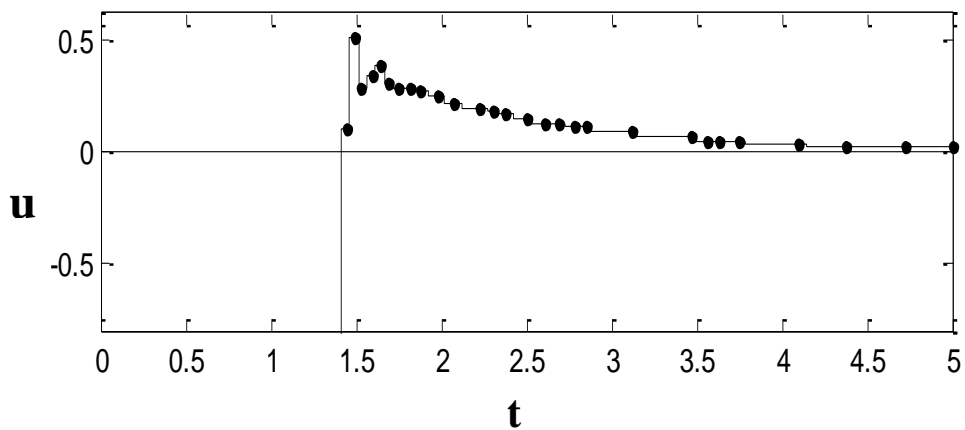
$$\dot{x}_b = k_a(x_a) - k_b(x_b)^2 - k_c x_a x_b$$

$$\dot{x}_c = k_c x_a x_b$$

$$\mathbf{x}_0 = [1 \ 0]^T$$

$$\begin{cases} RT(x_1(t) + x_2(t) + x_3(t)) < 340 \\ 0 \leq u \leq 8,5 \end{cases}$$

$$t \in [0 \ 2]$$



7. Lee & Ramirez Bioreactor $Q = 2.5$ (BALSA-CANTO et al, 2001)

$$J = \text{máx} \left[x_4(t_f) \cdot x_1(t_f) - Q \cdot \int_{t_0}^{t_f} u_2 dt \right]$$

$$\dot{x}_a = u_1 + u_2$$

$$\dot{x}_b = g_1 x_b - (u_1 + u_2) \frac{x_b}{x_a}$$

$$\dot{x}_c = \frac{u_1}{x_a} c_1 - (u_1 + u_2) \frac{x_c}{x_d} - g_1 \frac{x_b}{c_2}$$

$$\dot{x}_d = g_2 x_2 - (u_1 + u_2) \frac{x_d}{x_a}$$

$$\dot{x}_e = \frac{u_2 c_3}{x_a} - (u_1 + u_2) \frac{x_e}{x_a}$$

$$\dot{x}_f = -g_3 x_f$$

$$\dot{x}_g = g_3 (1 - x_g)$$

$$y_a = 14,35 + x_c + \left(\frac{x_c^2}{111,5} \right)$$

$$t_b = 0,22 + x_e$$

$$y_c = x_f + \frac{0,22}{y_b} \cdot x_g$$

$$g_1 = \frac{x_c}{y_a} \left(x_f + x_g \cdot \frac{0,22}{y_b} \right),$$

$$g_2 = 0,223 \frac{x_c}{y_a} \left(\frac{0,0005 + x_5}{0,022 + x_5} \right)$$

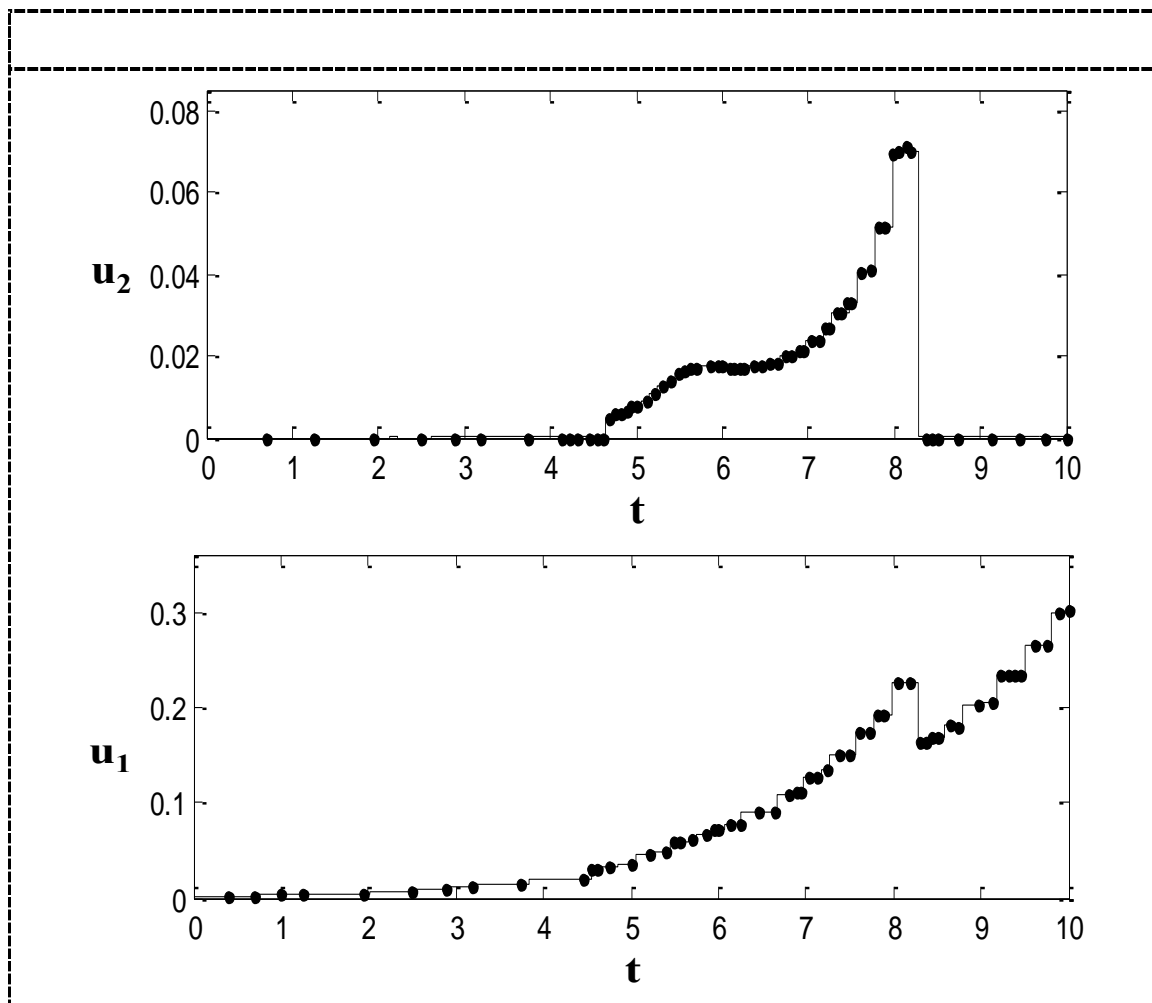
$$g_3 = 0,223 \frac{x_c}{y_a} \left(\frac{0,0005 + x_5}{0,022 + x_5} \right)$$

$$\mathbf{x}_o = [0 \ 0 \ 1 \ 5 \ 1]^T$$

$$0 \leq u_1 \leq 1$$

$$0 \leq u_2 \leq 1$$

$$t \in [0 \ 10]$$



8. Time Free Bang Bang Control (LIANG *et al.* 2003)

$$J = \text{mín} [t_f]$$

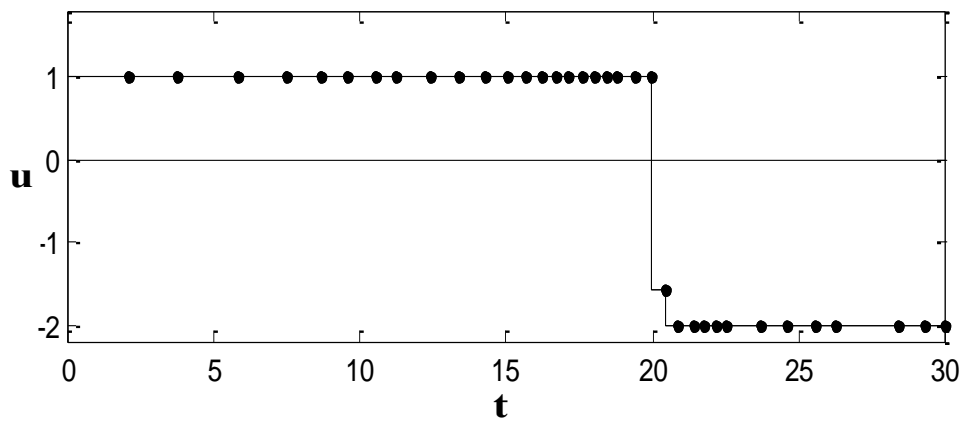
$$\dot{x}_a = x_b$$

$$\dot{x}_b = u$$

$$\mathbf{x}_0 = [0 \ 0]^T$$

$$\begin{cases} x_a(t_f) = 300 \\ x_b(t_f) = 0 \\ -2 \leq u \leq 1 \end{cases}$$

$$t \in [0 \ t_f]$$



9. Non-linear CSTR with Four Control Variables (BALSA-CANTO *et al.*, 2001)

$$J = \text{máx} \left[x_8(t_f) \right]$$

$$\dot{x}_1 = u_4 - (u_1 + u_2 + u_4)x_1 - 17,6x_1x_2 - 23x_1x_6u_3$$

$$\dot{x}_2 = u_1 - (u_1 + u_2 + u_4)x_2 - 17,6x_1x_2 - 146x_2x_3$$

$$\dot{x}_3 = u_2 - (u_1 + u_2 + u_4)x_3 - 73x_2x_3$$

$$\dot{x}_4 = -(u_1 + u_2 + u_4)x_4 + 35,2x_1x_2 - 51,3x_4x_5$$

$$\dot{x}_5 = -(u_1 + u_2 + u_4)x_5 + 219x_2x_3 - 51,3x_4x_5 \quad \dot{x}_6 = -(u_1 + u_2 + u_4)x_6 + 102,6x_4x_5 - 23x_1x_6u_3$$

$$\dot{x}_7 = -(u_1 + u_2 + u_4)x_7 + 46x_1x_6u_3$$

$$\dot{x}_8 = 5,8[(u_1 + u_2 + u_4)x_1 - u_4] - 3,7u_1 - 4,1u_2 + (u_1 + u_2 + u_4)(23x_4 + 11x_5 + 28x_6 + 35x_7) - 5u_3 - 0,09$$

$$\mathbf{x}_o = [0,1883 \ 0,2507 \ 0,0467 \ 0,0899 \ 0,7804 \ 0,1394 \ 0,1394 \ 0,000]^T$$

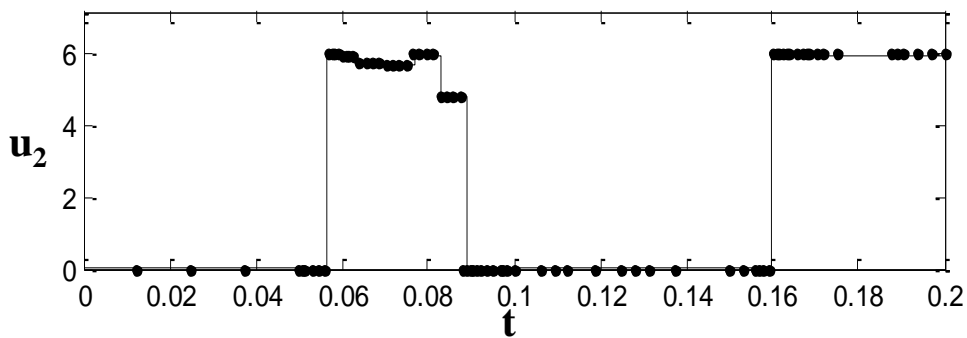
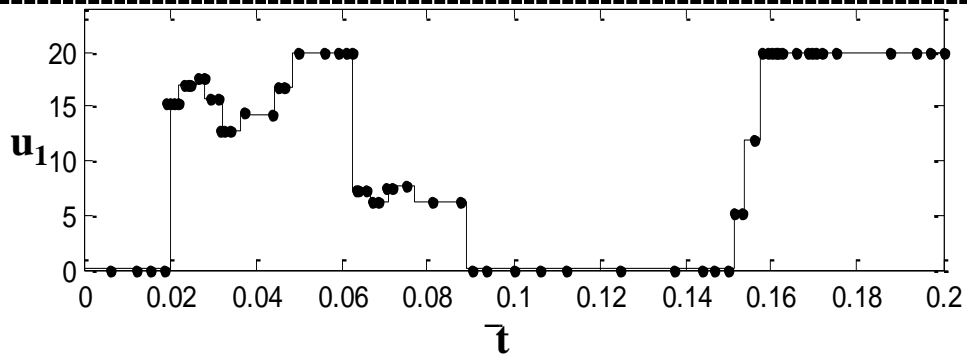
$$0 \leq u_1 \leq 20$$

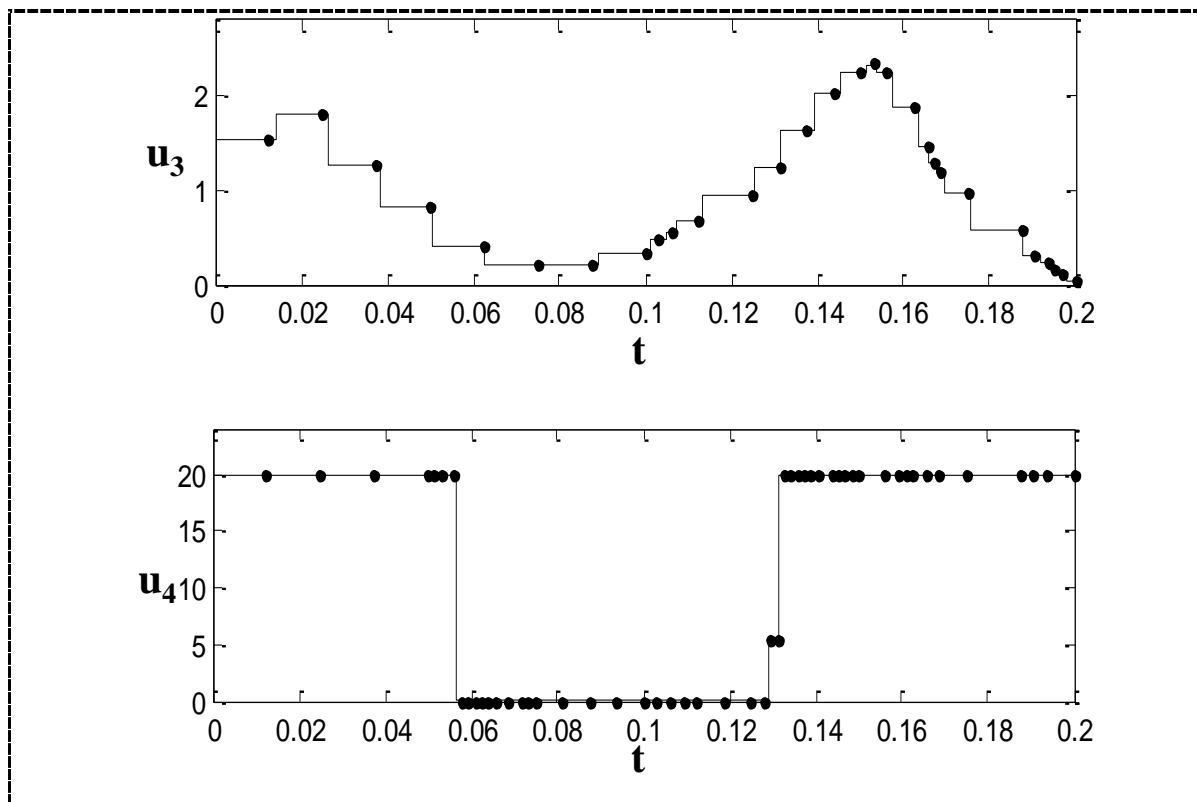
$$0 \leq u_2 \leq 6$$

$$0 \leq u_3 \leq 4$$

$$0 \leq u_4 \leq 20$$

$$t \in [0, 0,2]$$





10. Drug Displacement Problem (LUUS, 2002)

$$J = \text{mín} (t_f)$$

$$y_1 = y_2^2 + (y_3 y_4 - 2152,96 x_1 x_2)^{-1}$$

$$y_2 = 1 + 0,2(x_1 + x_2)$$

$$y_3 = y_2^2 + 232 + 46,4x_2$$

$$y_4 = y_2^2 + 232 + 46,4x_1$$

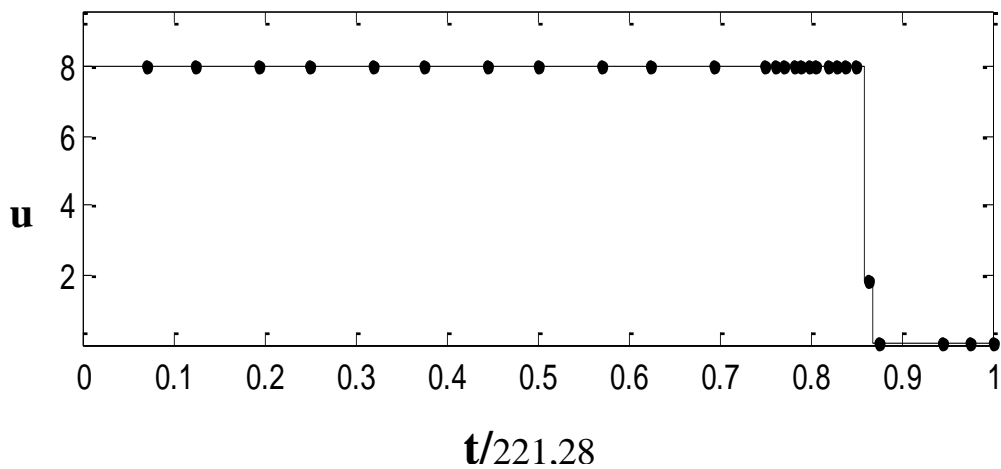
$$\dot{x}_1 = y_1 y_4 [(0,02 - x_1) + 46,4x_1 (u - 2x_2)]$$

$$\dot{x}_2 = y_1 [y_3 (u - 2x_2) + 46,4x_1 (0,02 - x_1)]$$

$$\mathbf{x}_0 = [0,02 \ 0]^T$$

$$0 \leq u \leq 8$$

$$t \in [0 \ t_f]$$



11. Continuous State Constraint Problem (JACOBSON e LELE, 1969)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_2 + u$$

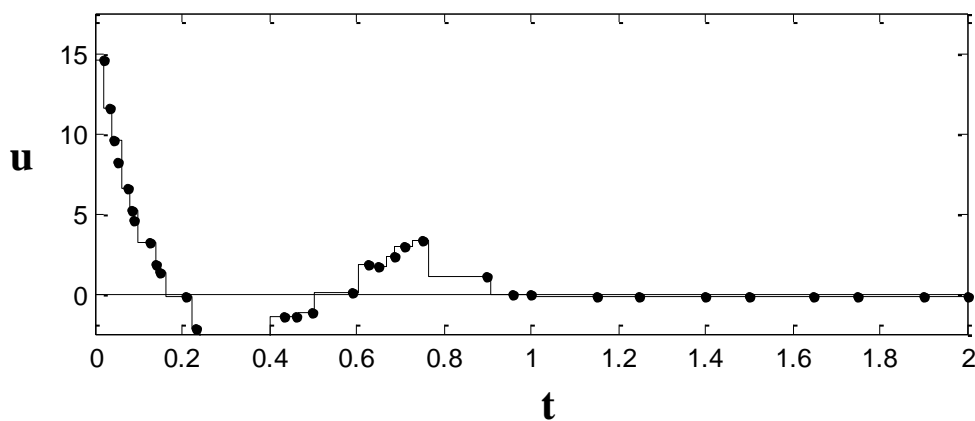
$$\dot{x}_3 = x_1^2 + x_2^2 + 0,005u^2$$

$$\mathbf{x}_0 = [0 \ -1 \ 0]^T$$

$$x_2(t) \leq 8(t-0,5)^2 + 0,5$$

$$-3 \leq u \leq 20$$

$$t \in [0 \ 2]$$



12. Lee & Ramirez Bioreactor $Q = 0$ (BALSA-CANTO *et al.*, 2001)

$$J = \text{máx} \left[x_4(t_f) \cdot x_1(t_f) - Q \cdot \int_{t_0}^{t_f} u_2 dt \right]$$

$$\dot{x}_a = u_1 + u_2$$

$$\dot{x}_b = g_1 x_b - (u_1 + u_2) \frac{x_b}{x_a}$$

$$\dot{x}_c = \frac{u_1}{x_a} c_1 - (u_1 + u_2) \frac{x_c}{x_d} - g_1 \frac{x_b}{c_2}$$

$$\dot{x}_d = g_2 x_2 - (u_1 + u_2) \frac{x_d}{x_a}$$

$$\dot{x}_e = \frac{u_2 c_3}{x_a} - (u_1 + u_2) \frac{x_e}{x_a}$$

$$\dot{x}_f = -g_3 x_f$$

$$\dot{x}_g = g_3 (1 - x_g)$$

$$y_a = 14,35 + x_c + \left(\frac{x_c^2}{111,5} \right)$$

$$t_b = 0,22 + x_e$$

$$y_c = x_f + \frac{0,22}{y_b} \cdot x_g$$

$$g_1 = \frac{x_c}{y_a} \left(x_f + x_g \cdot \frac{0,22}{y_b} \right),$$

$$g_2 = 0,223 \frac{x_c}{y_a} \left(\frac{0,0005 + x_5}{0,022 + x_5} \right)$$

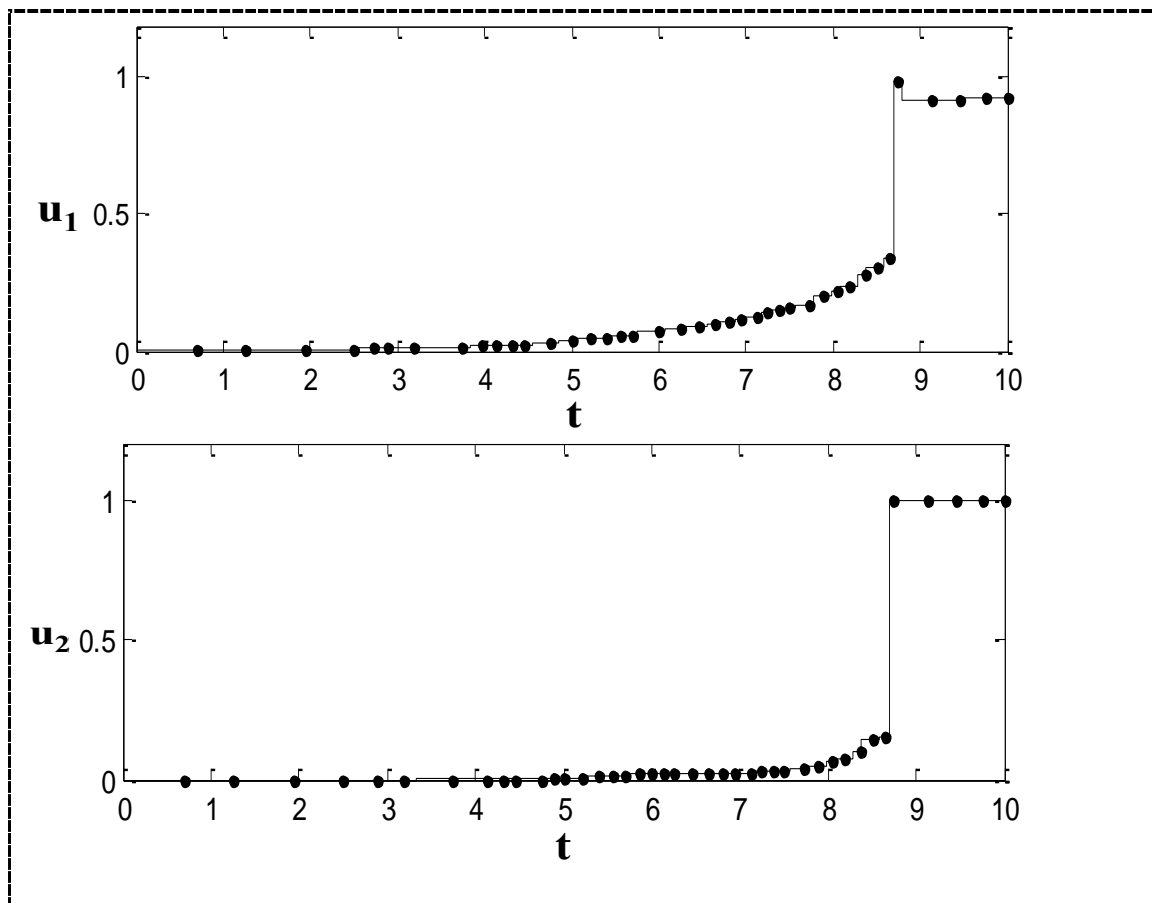
$$g_3 = 0,223 \frac{x_c}{y_a} \left(\frac{0,0005 + x_5}{0,022 + x_5} \right)$$

$$\mathbf{x}_o = [0 \ 0 \ 1 \ 5 \ 1]^T$$

$$0 \leq u_1 \leq 1$$

$$0 \leq u_2 \leq 1$$

$$t \in [0 \ 10]$$



13. Non-linear CSTR with Three Control Variables (BALSA-CANTO *et al.*, 2001)

$$J = \text{máx} \left[x_8(t_f) \right]$$

$$\dot{x}_1 = u_4 - (u_1 + u_2 + u_4)x_1 - 17,6x_1x_2 - 23x_1x_6u_3$$

$$\dot{x}_2 = u_1 - (u_1 + u_2 + u_4)x_2 - 17,6x_1x_2 - 146x_2x_3$$

$$\dot{x}_3 = u_2 - (u_1 + u_2 + u_4)x_3 - 73x_2x_3$$

$$\dot{x}_4 = -(u_1 + u_2 + u_4)x_4 + 35,2x_1x_2 - 51,3x_4x_5$$

$$\dot{x}_5 = -(u_1 + u_2 + u_4)x_5 + 219x_2x_3 - 51,3x_4x_5 \quad \dot{x}_6 = -(u_1 + u_2 + u_4)x_6 + 102,6x_4x_5 - 23x_1x_6u_3$$

$$\dot{x}_7 = -(u_1 + u_2 + u_4)x_7 + 46x_1x_6u_3$$

$$\dot{x}_8 = 5,8[(u_1 + u_2 + u_4)x_1 - u_4] - 3,7u_1 - 4,1u_2 + (u_1 + u_2 + u_4)(23x_4 + 11x_5 + 28x_6 + 35x_7) - 5u_3 - 0,09$$

$$\mathbf{x}_o = [0,1883 \ 0,2507 \ 0,0467 \ 0,0899 \ 0,7804 \ 0,1394 \ 0,1394 \ 0,000]^T$$

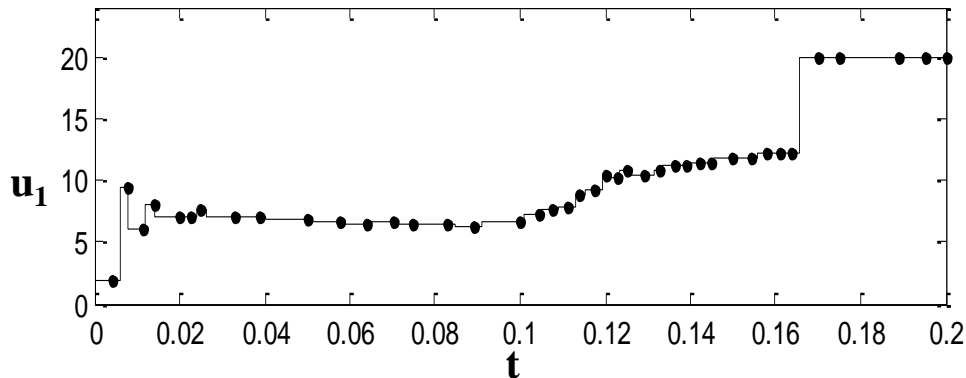
$$0 \leq u_1 \leq 20$$

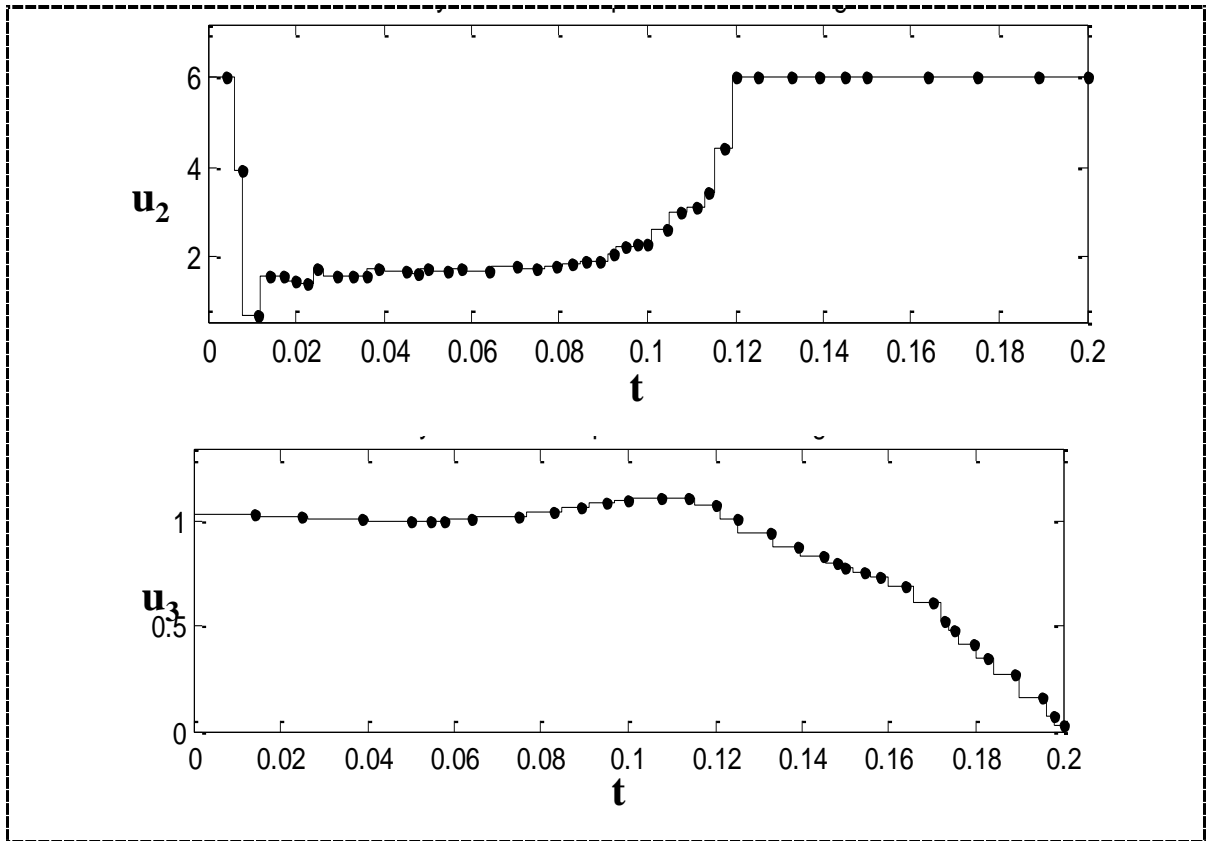
$$0 \leq u_2 \leq 6$$

$$0 \leq u_3 \leq 4$$

$$u_4 = 6$$

$$t \in [0, 0,2]$$





14. CSTR Reactor (SRINAVASAN *et al.*, 2003a)

$$J = \text{máx} \left[x_c(t_f) V(t_f) \right]$$

$$\dot{x}_a = -k_1 x_a x_b - \frac{u}{V} x_a$$

$$\dot{x}_b = -k_1 x_a x_b - 2k_2 x_b^2 + \frac{u}{V} (x_{b,in} - x_b)$$

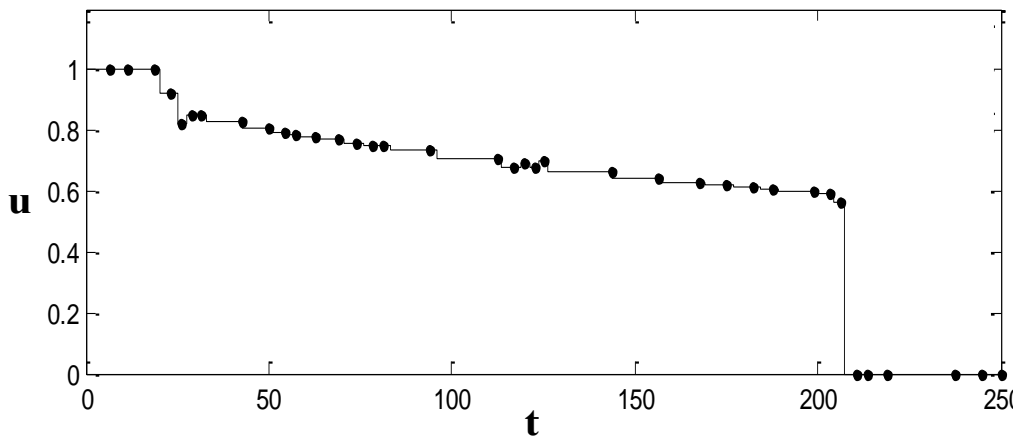
$$\dot{V} = u$$

$$x_c = \frac{1}{V} (x_{a,o} V_o - x_a V)$$

$$x_d = \frac{1}{2V} \left[(x_a + x_{b,in} - x_b) V - (x_{a,o} + x_{b,in} - x_{b,o}) V_o \right] \mathbf{x}_o = [0,72 \ 0,05]^T$$

$$\begin{cases} 0 \leq u \leq 0,001 \\ x_b(t_f) \leq 0,025 \\ x_d(t_f) \leq 0,15 \end{cases}$$

$$t \in [0 \ 250]$$



15. Batch Fermentator Penicilin (BANGA *et al.*, 2005)

$$J = \text{máx} [x_2(t_f) x_4(t_f)]$$

$$y_1 = 0,11x_3 / (0,006x_1 + x_3)$$

$$y_2 = (0,0055 \cdot x_3) / [0,0001 + x_3 \cdot (1 + 10 \cdot x_3)]$$

$$\dot{x}_1 = y_1 \cdot x_1 - (u \cdot x_1 / 500 \cdot x_4)$$

$$\dot{x}_2 = y_2 x_1 - 0,01x_2 - (ux_2) / 500x_4$$

$$\dot{x}_3 = (-y_1 x_1) / 0,47 - (y_2 x_1) / 1,2 - (x_1 x_3 0,029) / (0,0001 + x_3) + u(1 - x_3 / 500) / x_4$$

$$\dot{x}_4 = u / 500$$

$$x_0 = [1,5 \ 0 \ 0 \ 7]^T$$

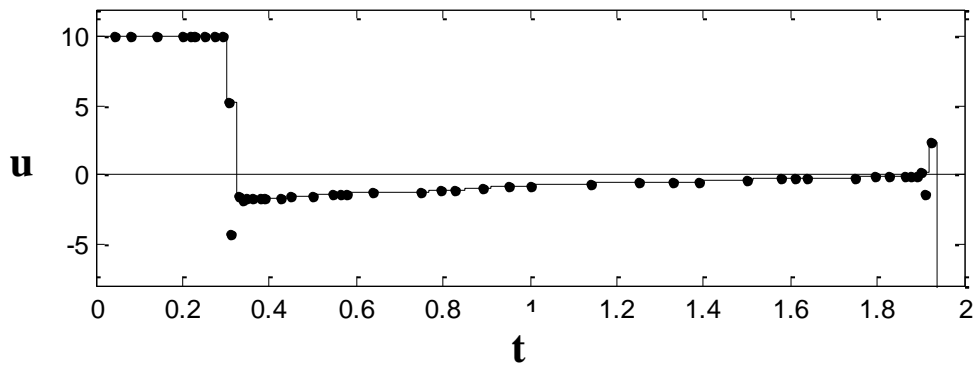
$$0 \leq x_1(t) \leq 40$$

$$0 \leq x_3(t) \leq 25$$

$$0 \leq x_4(t) \leq 10$$

$$0 \leq u \leq 50$$

$$t \in [0 \ 132]$$



16. The Mixed Catalyst Problem (BELL e SARGENT, 2000)

$$J = \text{máx} [x_c(t_f)]$$

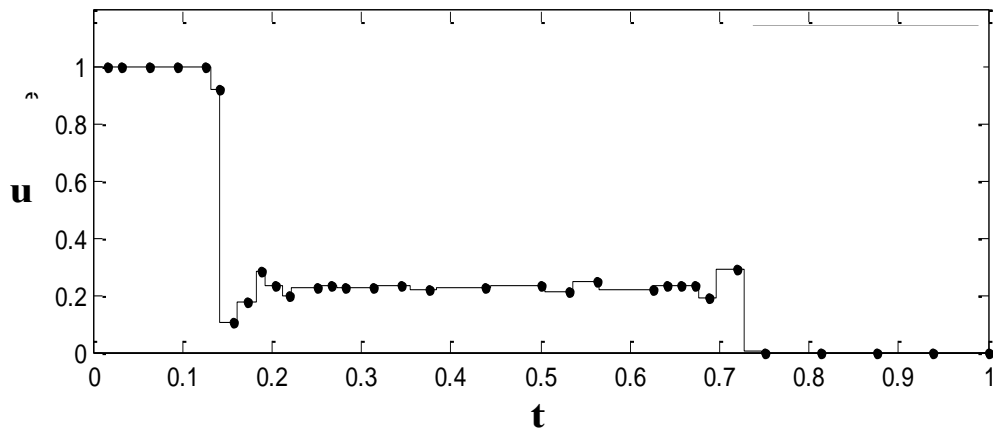
$$\dot{x}_a = u(10x_b - x_a)$$

$$\dot{x}_b = u(x_a - 10x_b) - (1-u)x_b$$

$$\mathbf{x}_o = [1 \ 0]^T$$

$$0 \leq u \leq 1$$

$$t \in [0 \ 1]$$



17. Benoit Chachuat Example (CHACHUAT, 2006)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_1 = x_2 + u$$

$$\dot{x}_2 = -u$$

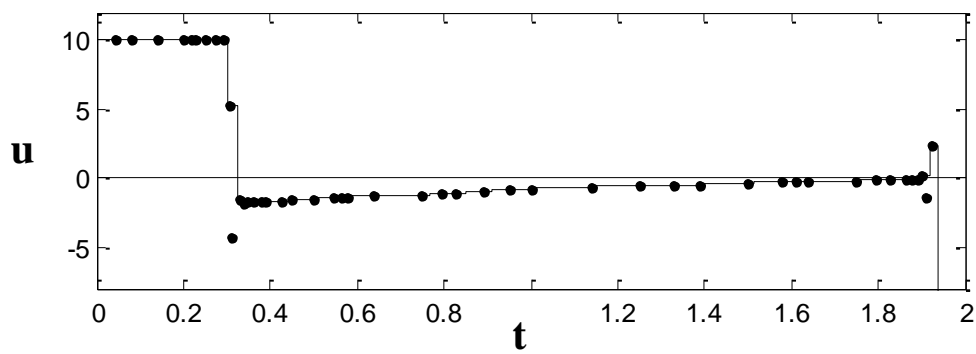
$$\dot{x}_3 = x_1^2/2$$

$$\mathbf{x}_0 = [1 \ 1 \ 0]^T$$

$$\mathbf{x}(t_f) = [0 \ 0]^T$$

$$-10 \leq u \leq 10$$

$$t \in [0 \ 2]$$



18. Singular Control 4 (LUUS, 2002)

$$J = \min [x_4(t_f)]$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = -x_3 u + 16x_5 - 8$$

$$\dot{x}_3 = u$$

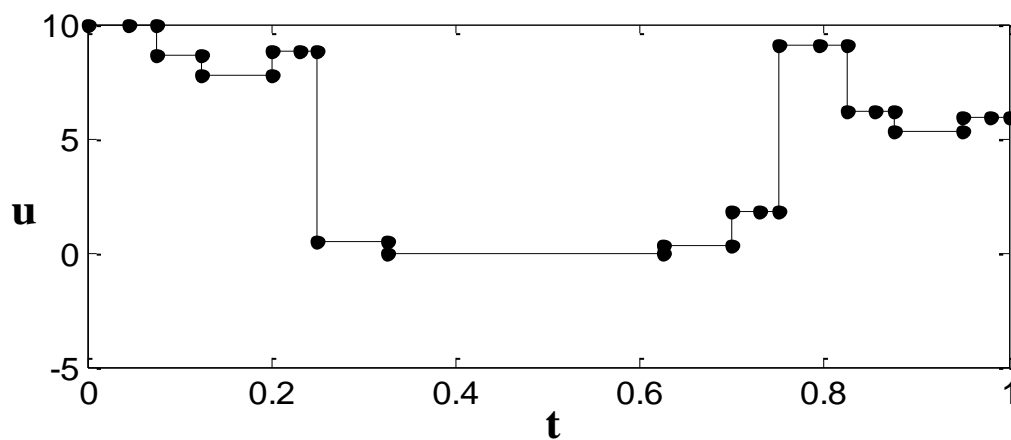
$$\dot{x}_4 = x_1^2 + x_2^2 + 0,005 \left(x_2 + 16x_5 - 8 - 0,1x_3 u^2 \right)^2$$

$$\dot{x}_5 = 1$$

$$\mathbf{x}_0 = [0 \ -1 \ -\sqrt{5} \ 0 \ 0]^T$$

$$-4 \leq u \leq 10$$

$$t \in [0 \ 1]$$



19. Quadruple Integral (RUTQUIST e EDVALL, 2010)

$$J = \text{mín} [t_f]$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = x_3$$

$$\dot{x}_3 = x_4$$

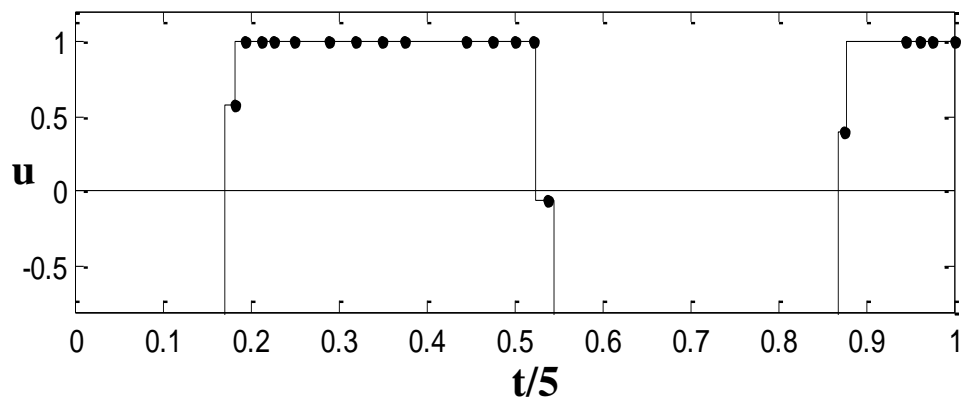
$$\dot{x}_4 = u$$

$$\mathbf{x}_0 = [0, 1, 0, 2, 0, 3, 0]^T$$

$$\mathbf{x}(t_f) = [0, 0, 0, 0]^T$$

$$-1 \leq u \leq 1$$

$$t \in [0, t_f]$$



20. Singular Control 2 (LUUS, 2002)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_1 = x_2$$

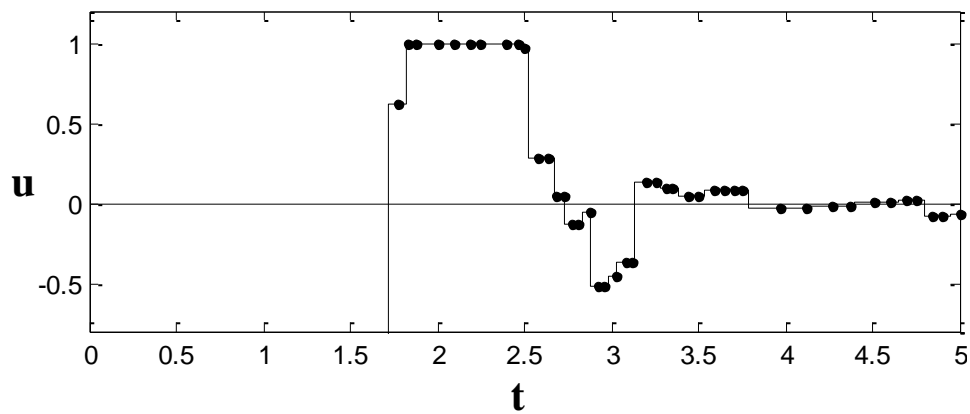
$$\dot{x}_2 = u$$

$$\dot{x}_3 = x_1^2$$

$$\mathbf{x}_0 = [0 \ 1 \ 0]^T$$

$$-3 \leq u \leq 20$$

$$t \in [0 \ 5]$$



21. Singular Control 3 (LUUS, 2002)

$$J = \min [x_3(t_f)]$$

$$\dot{x}_1 = x_2$$

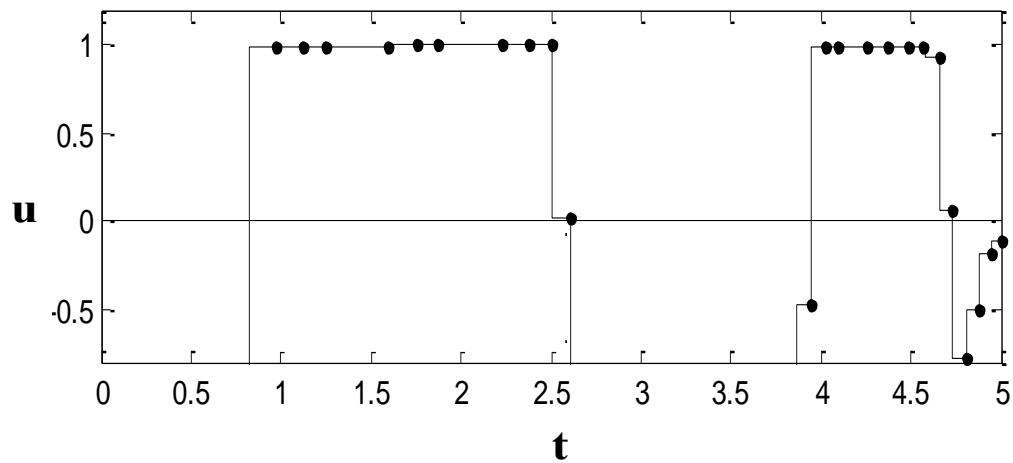
$$\dot{x}_2 = u$$

$$\dot{x}_3 = x_1^2$$

$$\mathbf{x}_0 = [0 \ 1 \ 0]^T$$

$$-3 \leq u \leq 20$$

$$t \in [0 \ 5]$$



22. Park and Ramirez Bioreactor (BALSA-CANTO *et al.*, 2001)

$$J = \text{máx} [x_a(t_f) \cdot V(t_f)]$$

$$\dot{x}_a = \Theta(x_a - x_b) - \frac{u}{V} x_a$$

$$\dot{x}_b = f_P x_b - \frac{u}{V} x_b$$

$$\dot{x}_c = \mu_x x_c - \frac{u}{V} x_b$$

$$\dot{x}_d = -7,3\mu_x(x_d) x_c - \frac{u}{V} (m - x_d)$$

$$\dot{V} = u$$

$$\Theta = \frac{4,75\mu_x}{0,2 + \mu_x}$$

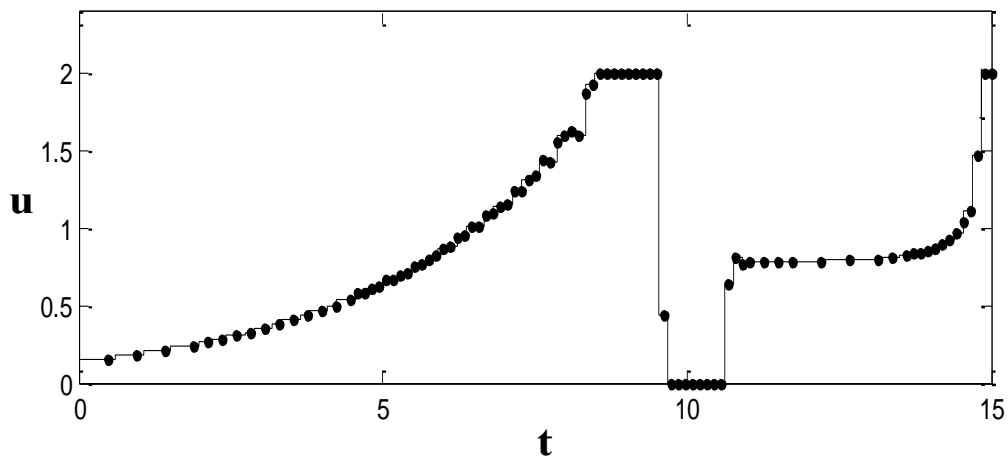
$$f_P = x_c \frac{e^{-5,0x_d}}{0,1 + x_d}$$

$$\mu_x = \frac{21,87x_d}{(x_d + 0,4)(x_d + 62,5)}$$

$$0 \leq u \leq 2$$

$$\mathbf{x}_0 = [0 \ 0 \ 1 \ 5 \ 1]^T$$

$$t \in [0 \ 15]$$



8. Conclusões

Este estudo evidenciou a potencialidade das wavelets como ferramenta para representação multi-escala de dados. Comparando-se com sua aplicação na área de processamento digital, as wavelets possuem ainda pouca aplicação em engenharia química, especialmente para solução de problemas que envolvem técnicas de discretização de equações diferenciais, como é o caso dos métodos diretos para solução de problemas de otimização dinâmica.

Foi desenvolvido um algoritmo, *Wavelet-thresholding* Adaptativo (WtA), que acopla três estratégias *thresholding*: *Visushrink*, *Sureshrink* e CVPS. A cada ciclo do algoritmo, um problema NLP é solucionado, de acordo com um determinado nível de discretização. Com a aplicação da transformada wavelets e subsequente compressão de acordo com a estratégia *thresholding* utilizada, é possível indicar quais regiões são candidatas a serem refinadas para a próxima etapa de otimização.

De acordo com os resultados de alguns exemplos de otimização dinâmica foi possível observar que a estratégia wavelets *thresholding* afeta bastante o desempenho computacional do algoritmo. A análise de 22 problemas permitiu construir uma métrica para a avaliação do desempenho dos algoritmos, definida como fator de dificuldade. Na maioria dos casos, o critério *Visushrink* obteve os melhores resultados, permitindo minimizar mais de 50% de custo computacional em alguns problemas, quando comparado ao critério de discretização equidistante da malha, assumido como solução de referência. No entanto, para problemas com fatores de dificuldade elevados, a estratégia *Sureshrink* obteve resultados mais satisfatórios. Foi observado que em tais problemas, a trajetória de controle apresenta número de arcos maior ou igual a três. Tal característica prova que a natureza adaptativa do *Sureshrink* é bastante adequada para detectar discontinuidades e adaptar a discretização, fato também observado na literatura de processamento de imagens. Por outro lado, na presença de trajetórias de controle suaves, ou com pouca ocorrência de discontinuidades, a estratégia *Sureshrink* tende a apresentar desempenho inferior ao *Visushrink*. A metodologia desenvolvida neste trabalho, CVPS, lida melhor com as deficiências das estratégias *Sureshrink* e *Visushrink* em situações extremas. Isso se mostra evidente em problemas com múltiplas variáveis

de controle, onde ocorre naturalmente o cômputo de trajetórias de controle com diversas características. Por exemplo, a presença de uma trajetória com grande ocorrência de descontinuidades Logo, o CVPS apresenta maior regularidade quando comparado aos outros métodos, principalmente em casos extremos, com poucos arcos de controle ou muitos arcos de controle.

Algumas sugestões baseadas nas dificuldades encontradas e identificação de melhorias podem ser ressaltadas:

(i) Constata-se que o critério de convergência é fundamental para avaliar o desempenho de cada algoritmo. O critério utilizado no WtA avalia a variação relativa da função objetivo. Porém, observa-se que o critério é local, e não impede o algoritmo de parar em caso de baixa variação da função objetivo em uma etapa intermediária sem ter atingido o valor ótimo de fato. Nos problemas analisados não foram detectadas essas dificuldades, porém, critérios de convergência global, como por exemplo, incorporação de avaliação da qualidade da discretização, poderiam aprimorar o critério de convergência;

(ii) Há outras estratégias *wavelet thresholding* na literatura de estatística, como o *Minimaxi* e *Bayeshrink*, que não foram discutidas neste trabalho. Todas podem ser facilmente incorporadas à rotina MATLAB desenvolvida. No entanto, estudos futuros podem ser realizados para averiguar a viabilidade de aplicação desses critérios;

(ii) Nesta tese foi utilizada a função Daubechie-I ou Haar. Essa é uma base altamente indicada para detecção de descontinuidades, pois é naturalmente uma função descontínua. Porém, novas bases wavelets também podem ser incorporadas ao algoritmo. Um estudo avaliando o desempenho de diferentes bases wavelets ajudaria a prever, da mesma maneira que diferentes estratégias *thresholding*, quais bases seriam indicadas para diferentes problemas.

(iii) O algoritmo WtA apresentado aqui foi aplicado ao método *single-shooting*. Ademais, aplicações incluindo o algoritmo *multiple-shooting* também são possíveis de serem incorporadas ao WtA. Ressalta-se também a possibilidade de implementação ao método simultâneo.

Finalmente, a maior contribuição do trabalho consiste em avaliar diferentes técnicas de compressão de dados, ainda não difundidas na área de otimização dinâmica. Além de se mostrar uma ferramenta bastante útil para aprimoramento dos métodos numéricos, a discussão sobre o tema encoraja futuras implementações, não só para otimização dinâmica, mas também para solução de equações diferenciais e procedimentos de compressão de grande massa de dados.

Referências

- ABRAMOVICH, F., BENJAMINI, Y., 1996, “Adaptive Thresholding of Wavelet Coefficients”, *Computational Statistics and Data Analysis*, v. 22, n. 4, pp. 351-361.
- ADJIMAN, C.S., ANDROULAKIS, I.P., MARANAS, C.D., FLOUDAS, C.A., 1996, “A Global Optimization Method, α BB, for Process Design”, *Computers & Chemical Engineering*, v. 20, n. 1, pp. S419-S424.
- ALMEIDA E., SECCHI, A.R., 2012, “Solving dynamic optimization infeasibility problems”, *Computers & Chemical Engineering*, v. 36, n. 10, pp. 27-246.
- ALMEIDA Nt, E., SECCHI, A.R., 2011, “Dynamic Optimization of a FCC Converter Unit: Numerical Analysis”, *Brazilian Journal of Chemical Engineering*, v. 28, n. 1, pp. 117-136.
- ALMEIDA, E., 2011, *Otimização Dinâmica em Tempo Real: Arquitetura de Software, Diagnóstico e Análise de Inviabilidades*, Tese de Dsc., Departamento de Engenharia Química, Universidade Federal do Rio Grande do Sul, RS, Brasil.
- ASSASSA, F., MARQUARDT, W., 2014, “Dynamic optimization using adaptive direct multiple shooting”, *Computers & Chemical Engineering*, v. 60, n. 10, pp. 242-259.
- ATHANS, M., FALB, P.L., 1966, *Optimal Control. An Introduction to the Theory and Its Applications*, McGraw-Hill Book Company, USA.
- AZZALINE, A., FARGE, M., SCHNEIDER, K., 2005, “Nonlinear Wavelet Thresholding: A Recursive Method to Determine the Optimal Denoising Threshold”, *Applied and Computational Harmonic Analysis*, v. 18, n. 2, pp. 177-185.
- BALSA-CANTO, E., BANGA, J.R., ALONSO, A.A., VASSILIADIS, V.S., 2001, “Dynamic Optimization of Chemical and Biochemical Processes using Restricted Second-Order Information”. *Computers and Chemical Engineering*, v. 25, n. 4, pp. 539-546.

- BANGA, J.R., BALSACANTO, E., MOLES, C.G., ALONSO, A.A., 2005, “Dynamic Optimization of Bioprocesses: Efficient and Robust Numerical Strategies”, *Journal of Biotechnology*, v. 117, pp. 407-419.
- BANGA, J.R., IRIZARRY-RIVERA, R., SEIDER, W.D., 1998, “Stochastic optimization for optimal and model-predictive control”, *Computers & Chemical Engineering*, v. 22, n. 4, pp. 603-612.
- BARTUSEK, K., PRINOSIL, J., SMEKAL, Z., 2011, “Optimization of Wavelet-Based De-noising in MRI”, *Radioengineering*, v. 20, n.1, pp. 85 - 93.
- BAYÓN, L., GRAU, J.M, RUIZ, M.M., SUÁREZ, P.M., 2010, “Initial guess of the solution of dynamic optimization of chemical processes”, *Journal of Mathematical Chemistry*, v. 48, n. 1, pp. 28-37.
- BELL, M.L., SARGENT, R.W.H., 2000, “Optimal control of inequality constrained DAE systems”, *Computers and Chemical Engineering*, v. 24, n. 1, pp. 2385-2404.
- BELLMAN, R., 1957, *Dynamic Programming*, Princeton University Press, Princeton, New Jersey, USA.
- BERTOLUZZA, S. e NALDI, G., 1994, “A Wavelet Collocation Method for the Numerical Solution of Partial Differential Equations”, *Applied and Computational Harmonic Analysis*, v.3, n.1, pp. 1-9.
- BETTS, J.T., 2010, *Practical Methods for Optimal Control and Estimation Using Nonlinear Programming*, SIAM Society for Industrial and Applied Mathematics, Philadelphia, USA.
- BETTS, J.T., HUFFMAN, W.P., 1992, “Application of Sparse Nonlinear Programming to Trajectory Optimization”, *Journal of Guidance, Control and Dynamics*, v. 15, n. 1, pp. 198-206.
- BETTS, J.T., HUFFMAN, W.P., 1993, “Path constrained Trajectory Optimization Using Sparse Sequential Quadratic Programming”, *Journal of Guidance, Control and Dynamics*, v. 16, n. 1, pp. 59-68.
- BETTS, J.T., HUFFMAN, W.P., 1999, “Mesh Refinement in Direct Transcription Methods for Optimal Control”, *Optimal Control Applications & Methods*, v. 8, n. 1, pp.1-21.

- BEYLKIN, G., 1993, “Wavelets and Fast Numerical Algorithms”, *Proceeding of Symposia in Applied Mathematics*, v.47, pp. 89-117.
- BIEGLER, L.T., 1984, “Solution of Dynamic Optimization Problems by Successive Quadratic Programming and Orthogonal Collocation”, *Computers and Chemical Engineering*, v. 8, n. (3-4), pp.243–248.
- BIEGLER, L.T., 2007, “An Overview of Simultaneous Strategies for Dynamic Optimization”, *Chemical Engineering and Processing*, v. 46, n. 11, pp. 1043-1053.
- BIEGLER, L.T., 2010, *Nonlinear Programming Concepts, Algorithms, and Applications to Chemical Processes*, Society for Industrial and Applied Mathematics SIAM Series on Optimization, Pennsylvania, USA.
- BIEGLER, L.T., CERVANTES, A.M., WATCHER, A., 2002, “Advances in Simultaneous Strategies for Dynamic Process Optimization”, *Chemical Engineering Science*, v. 57, n. 4, pp. 575-593.
- BINDAL, A., KHINAST, J.G., IERAPETRITOU, M.G., 2003, “Adaptive Multiscale Solution of Dynamical Systems in Chemical Processes Using Wavelets”, *Computers and Chemical Engineering*, v. 27, n. 1, pp. 131-142.
- BINDER, T., 2002, *Adaptive Multiscale Methods for the Solution of Dynamic Optimization*, Problems. VDI-Fortschrittsbericht, Reihe 8, Nr. 969. VDI-Verlag. Düsseldorf, PhD Thesis., Rheinisch-Westfaelische Technische Hochschule, Aachen, Deutschland.
- BINDER, T., BLANK, L., DAHMEN, W., MARQUARDT, W., 2000a, “Grid Refinement in Multiscale Dynamic Optimization”, *European Symposium on Computer Aided Process Engineering*, v. 10, pp. 31-36.
- BINDER, T., CRUSE, C.A., VILLAR, C., MARQUARDT, W., 2000b, “Dynamic Optimization Using a Wavelet Based Adaptive Control Vector Parameterization Strategy”, *Computers & Chemical Engineering*, v. 24, n. 2-7, pp. 1201-1207.
- BLANCO, R.M., 2002, *Um Método Adaptativo de Diferenças Finitas Utilizando Wavelets e sua Aplicação na Resolução Numérica de um Modelo Transiente de Micropropulsores a Hidrazina*, Dissertação de Mestrado., Universidade Estadual

- de Campinas. Instituto de Matemática, Estatística e Computação Científica, São Paulo, Brasil.
- BOJKOV, B.; LUUS, R., 1996, “Optimal Control of Nonlinear Systems with Unspecified Final Times”, *Chemical Engineering Science*, **1996**, v. 56, n. 6, pp. 905-916.
- BORGES, F.B., 2005, *Codificadores de Voz a Baixas Taxas Operando em Ambientes Ruidosos e Redes IP*, Dissertação de MSc., Departamento de Energia Elétrica-PUC-RJ, Rio de Janeiro, Brasil.
- BRENAN, K.E., CAMPBELL S.L., PETZOLD, L.R., 1996, *Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations*, SIAMS Classic in Applied Mathematics, Society for Industrial and Applied Mathematics, NY, USA.
- BRIESEN, H., MARQUARDT, W., 2003, “An Adaptive Multigrid Method for Steady-State Simulation of Petroleum Mixture Separation Processes”, *Ind. Eng. Chem. Res* v. 42, pp. 2334-2348.
- BROOKE, D., KEDRICK, MEERAUS, A., RAMAN, .R., 1998, GAMS Home Page: <http://www.gams.com>, 1998.
- BRYSON, A. E., HO, Y. C., 1975, *Applied Optimal Control: Optimization, Estimation and Control*, CRC Press, Washington DC, USA.
- BURRUS, C.S., GOPINATH, R.A., GUO, H., 1998, *A Primer Introduction to Wavelets and Wavelets Transforms: a primer*, New Jersey, USA, Prentice-Hall.
- CAUMO, L., 2006, *Aplicação e Cálculo da Derivada de Sinais de Processos Industriais*, Dissertação de Mestrado., Escola de Engenharia-DEQ, Universidade Federal do Rio Grande do Sul, Brasil.
- CERVANTES, A.M., BIEGLER, L.T., 2000, “A Stable Elemental Decomposition for Dynamic Process Optimization”, *Journal of Computational and Applied Mathematics*, v. 120, n. (1.2), pp. 41-57.
- CERVANTES, A.M., TONELLI, S., BRANDOLIN, A., BANDONI, J.A., BIEGLER, L.T., “Large-scale dynamic optimization for grade transitions in a low density polyethylene plant”, *Computers & Chemical Engineering*, v. 26, n. 2, pp. 227-237.

- CERVANTES, A.M., WATCHER, A., TUTUNCU, R.H., BIEGLER, L.T., 2000, “A Reduced Space Interior Point Strategy for Optimization of Differential Algebraic Systems”, *Computers & Chemical Engineering*, v. 24, n.1, pp. 39-51.
- CHACHUAT, B.C., 2006, *Nonlinear and Dynamic Optimization from Theory to Practice*, Laboratoire d'Automatique, École Polytechnique Fédérale de Lausanne. <http://infoscience.epfl.ch/record/111939>.
- CHEN, C.F., HSIAO, C.H., 1997, “Haar Wavelet Method for Solving Lumped and Distributed-Parameter Systems”, *Control Theory and Applications, IEE Proceedings*, v. 44, n. 1, pp. 87-94.
- CHEN, C.F., HSIAO, C.H., 1999, “Wavelet Approach to Optimising Dynamic Systems”, *Control Theory and Applications, IEE Proceedings*, v. 146, n. 2, pp. 213-219.
- CIZNIAR, M., SALHI, D., FIKAR, M., LATIFI, M.A., 2005, “A Matlab Package For Orthogonal Collocations on Finite Elements in Dynamic Optimisation”, *15th Int. Conference Process Control*, Strbské Pleso, Slovakia.
- COSTA, E.F., SECCHI, A.R., BISCAIA Jr., E.C., 2002, “Automatic Integration of High-Index Dynamic Systems”. *Computer Aided Chemical Engineering*, v. 10, pp. 865-870.
- CRUZ, P., MENDES, A., MAGALHÃES, F.D., 2001, “Using Wavelets for Solving PDEs: An Adaptive Collocation Method”, *Chemical Engineering Science*, v. 56, n.10, pp. 3305-3309.
- CUTHRELL, J.E., BIEGLER, L.T., 1989, “Simultaneous optimization and solution methods for batch reactor control profiles”, *Computers & Chemical Engineering*, v.13, n.1, pp., 49-62.
- DAI, R. e COCHRAN JR., J.E., 2009, “Wavelet Collocation Method for Optimal Control Problems”, *Journal of Optimization Theory and Applications*, v. 143, n. 2, pp. 265-278.
- DANGETI, S., 2003, *Denoising Techniques – A Comparison*, PhD Thesis, Graduate Faculty of the Louisiana State University and Agricultural and Mechanical College, USA.

- DAUBECHIES, I., 1992, *Ten Lectures on Wavelets*, Philadelphia, USA, Society for Industrial and Applied Mathematics.
- DIEHL, M., 2011, *Numerical Optimal Control*, Optimization in Engineering Center (OPTEC), Leuven, Belgium.
- DIEHL, M., BOCK, H.G., DIEDAM, H., WIEBER, P-B., 2006, “Fast Direct Multiple Shooting Algorithms for Optimal Robot Control”, *Fast Motions in Biomechanics and Robotics*, v. 340, pp. 65-93.
- DONOHO, D.L. e JOHNSTONE, I.M., 1994, “Ideal Spatial Adaptation by Wavelet Shrinkage”, *Biometrika*, v. 81, pp. 425-455.
- DONOHO, D.L. e JOHNSTONE, I.M., 1995, “Adapting to Unknown Smoothness via Wavelet Shrinkage”, *Journal of the American Statistical Association*, v. 90, n. 432, pp. 1200-1224.
- DONOHO, D.L., 1992, “-De-noising by Soft Thresholding”, *IEEE Transactions in Information Theory*, v. 41, n. 3, pp. 613-627.
- DONOHO, D.L., 1993, “Unconditional Bases are Optimal Bases for Data Compression and for Stastical Estimation”, *Applied and Computational Harmonic Analysis*, n. 1, pp. 100-115.
- DONOHO, D.L., 1995, "De-noising by Soft-Thresholding", *IEEE Transactions on Information Theory*, v. 41, n. 3, pp. 613-627.
- DRAG, P., STYCZEN, K., 2012, Parallel Simultaneous Approach for Optimal Control of DAE Systems, *Proceedings of the Federated Conference on Computer Science and Information Systems*, pp. 517-523.
- ESPOSITO W.R., FLOUDAS, C.A., 2000, "Global Optimization of Nonconvex Problems with Differential-Algebraic Constraints", *Computer Aided Chemical Engineering*, v. 8, pp. 73-78.
- FAHOO, F., ROSS, I.M., “Convergence of the Costates Does Not Imply Convergence of the Control”, 2008, *Journal of Guidance, Control and Dynamics*, v.31, v.5, pp. 1492-1496.
- FALUGI, P., KERRIGAN, E., 2010, *Imperial College London Optimal Control Software User Guide (ICLOCS)*, Department of Electrical and Electronic Engineering, Imperial College London, English, UK.

- FEEHERRY, .W.F., BARTON, P.I., 1996, “Dynamic Simulation and Optimization with Inequality Path Constraints”, *Computers & Chemical Engineering*, v. 20, pp. S707-S712.
- FESKO, O., 2012, “A Parallel Approach to Improvement and Estimation of the Approximate Optimal Control”, *Journal of Computational Science*, v. 3, n. 6, pp. 486-491.
- FLORES, A. X., 2011, *Cálculo Variacional: Aspectos Teóricos e Aplicações*, Dissertação de MSc., Instituto de Geociências e Ciências Exatas – Campus de Rio Claro, Universidade Estadual Paulista Júlio de Mesquita Filho, São Paulo, Brasil.
- FRAZIER, M.W., 1999, *An Introduction to Wavelets through Linear Algebra*, (Undergraduate Texts in Mathematics), Springer-Verlag, New York, USA.
- FREIKAMP, F., 1999, *Design and Implementation of Routines for the Evaluation of Gradients in Wavelet – Discretized Optimal Control Problems*, New Jersey, Rheinisch-Westfaelische Technische Hochschule, Aachen, Deutschland..
- GARCÍA, C. E., PRETT D.M., MORARI, M., 1989, “Model Predictive Control: Theory and Practice-a Survey”, *Automatica*, v. 25, n. 3, pp. 335-348.
- GEERING, H.P., 2007, *Optimal Control with Engineering Applications*, Springer, Zurich, Switzerland.
- GHANBARI, Y., KARAMI-MOLLAEI, M.R., 2006, “A New Approach for Speech Enhancement Based on the Adaptive Thresholding of the Wavelet Packets”, *Speech Communication*, v. 48, n. 8, pp. 927-940.
- GRASS, D., SEIDL, A. 2011, *OCMat Manual*, Institute of Mathematical Methods in Economics, Technische Universität Wien, Wien, Österreich.
- HADYANTO, H., ESVELD, D.C., BOOM, R.M., VAN STRATEN, G., VAN BOXTEL, A.J.B., 2008, “Control Vector Parameterization with Sensitivity Based Refinement Applied to Baking Optimization”, *Food and Bioproducts Processing*, v. 86, n. 2, pp. 130-141.
- HALL, P. PATIL, P., 1996, “Effects Of Threshold Rules on Performance of Wavelet-Based Curve Estimators”, *Statistica Sinica*, v. 6, pp. 331-345.

- HANNEMANN-TAMÁS, R., MARQUARDT, 2012, “How to Verify optimal controls computed by direct shooting methods? –A tutorial”, *Journal of Process Control*, v. 22, n. 2, pp. 494-507.
- HARTWICH, A., ASSASSA, F., MARQUARDT, W., 2010, “Hybrid Shooting – a New Optimization Method for Unstable Dynamic Systems”, *Proceedings of the 9th International Symposium on Dynamics and Control of Process Systems (DYCOPS 2010)*.
- HARTWICH, A., MARQUARDT, W., 2010, “Dynamic optimization of the load change of a large-scale chemical plant by adaptive single shooting”, *Computers and Chemical Engineering*, v. 34, n. 11, pp. 1873-1889.
- HARTWICH, A., STOCKMANN, K., TERBOVEN, C., FEUERRIEGEL, S., MARQUARDT, W., 2011, “Parallel sensitivity analysis for efficient large-scale dynamic optimization”, *Optimization and Engineering*, v. 12, n. 4, pp. 489-508.
- HAYA, A.A.A, 2011, *Solving Optimal Control Problem Via Chebyshev Wavelet*, MSc Dissertation, Islamic University of Gaza.
- HIRMAJER, T, BALSACANTO, E., BANGA, J.R., 2009, “DOTcvpSB: a Matlab Toolbox for Dynamic Optimization in Systems Biology”, *BMC Bioinformatics*.
- HOFFMAN, C., KIRCHES, C., POTSCHEKA, A., SAGER, S., WIRSCHING, L., 2011. *MUSCOD User Manual*, Simulation & Optimization Group, University of Heidelberg, Germany.
- HOOG, F.R., MATTHEIJ, R.M.M., 1987, “An Algorithm for Solving Multi-point Boundary Value Problems”, *Computing*, v. 38, n. 3, pp. 219-234.
- HSIAO, C.H., 2004, “Haar Wavelet Direct Method for Solving Variational Problems”, *Mathematics and Computers in Simulation*, v. 64, n. 5, pp. 569-585.
- HSIAO, C.H., WANG, W.J., 1999, “Optimal Control of Linear Time-Varying Systems via Haar Wavelets”, *Journal of Optimization Theory and Applications*, v. 103, n. 3, pp. 641-655.
- HUANG, R., 2010, Nonlinear Model Predictive Control and Dynamic Real Time Optimization for Large-scale Processes, PhD Thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

- HUANG, Y.J., REKLAITIS, G.V., VENKATASUBRAMANIAN, V., 2002, “Model Decomposition based Method for Solving General Dynamic Optimization Problems”, *Computers & Chemical Engineering*, v. 26, pp. 863-873.
- JACOBSON, D.H., LELE, M.M., 1969, “A Transformation Technique for Optimal Control Problems with a State Variable Inequality Constraint”, *IEEE Transactions on Automatic Control*, v. 14, n. 5, pp.457-464.
- JAFFARD, S., 1992, “Wavelet Methods for Fast Resolution of Elliptic Problems”, *SIAM Journal on Numerical Analysis*, v. 29, n.4, pp. 965-986.
- JAIN, S. & TSOTRAS, P., 2007, “Multiresolution-Based Direct Trajectory Optimization”, *Proceedings of the 46th IEEE Conference on Decision and Control*, n. 4434076, pp. 5991-5996.
- JANSEN, M., 2000, *Wavelet Thresholding and Noise Reduction*,, PhD Thesis, Katholieke Universiteit Leuven, Belgie.
- JENNINGS, L.S., TEO, K.L., 1990, “A Computational Algorithm for Functional Inequality Constrained Optimization Problems”, *Automatica*, v. 26, n. 2, pp. 371-375.
- KAMIEN, M.I., SCHWARTZ, N.L., 1991, *Dynamic Optimization The Calculus of Variations and Optimal Control in Economics and Management*, Advanced Textbooks in Economics, Editors C.J. BLISS and INTRILIGATOR, Second Edition, Illinois, USA.
- KARAMI, A., KARIMI, H.R., MOSHIRI, B., MARALANI, P.J., 2008, “Investigation of Interpolating Wavelets Theory for the Resolution of PDE”, *International Journal of Contemporary Mathematical*, v. 3, n. 21, pp. 1017-1029.
- KEISER, J.M., 1995, *Wavelet Based Approach of Nonlinear Partial Differential Equations and II. Non Linear Waves in Fully Discrete Dynamical Systems*, PhD Thesis, Faculty of the Graduate School of the University of Colorado, USA.
- KIRCHES, C., LEYFFER, S., 1011, *TACO - A Toolkit for AMPL Control Optimization*, Mathematical Programming Computation Manuscript.
- KOTZ, S., NADARAJAH, S., 2000, *Extreme Value Distributions Theory and Applications*, Imperial College Press, London, England.

- KREYSZIG, E., 1989, *Introductory Functional Analysis with Applications*, John Wiley & Sons.
- KUMAR, V. e MEHRA, M., 2007, “Wavelet Optimized Finite Difference Method Using Interpolating Wavelets for Solving Singularity Perturbed Problems”, *Journal of Wavelet Theory and Applications*, v. 1, n. 1, pp. 83-96.
- LANG, Y.D., BIEGLER, L.T., 2007, “A Software Environment for Simultaneous Dynamic Optimization”, *Computers & Chemical Engineering*, v. 31, n. 8, pp. 931-942.
- LEADBETTER, M.R., LINDGREN, G., R., HOLGER, 1983, *Extremes and Related Properties of Random Sequences and Processes*, New York, Springer-Verlag.
- LEE, M.H., HAN, C., CHANG, K.S., 1999, “Dynamic Optimization of a Continuous Polymer Reactor Using a Modified Differential Evolution Algorithm”, *Industrial & Engineering Chemistry Research*, v. 38, n. 12, pp. 4825-4831.
- LEINEWEBER, D.B., SCHAFER, A.M.H., BOCK, H.G., SCHLODER, J.P., 2003, “An Efficient Multiple Shooting Based Reduced SQP Strategy for Large-scale Dynamic Process Optimization Part II: Software Aspects and Applications”, *Computers and Chemical Engineering*, v. 27, pp. 167-174.
- LEMONS, E.M., 2007, *Implementação dos Métodos de Resíduos Ponderados por Quadraturas Gaussianas*, Tese de DSc., Programa de Engenharia Química, COPPE-UFRJ, Rio de Janeiro, Brasil.
- LEPIK, U., 2005, “Numerical Solution of Differential Equations Using Haar Wavelets”, *Mathematics and Computers in Simulation*, v. 68, n. 2, pp. 127–143.
- LI, P., FANG, Y., 2010, “A Wavelet Interpolation Galerkin Method for the Simulation of MEMS Devices under the Effect of Squeeze Film Damping”, *Mathematical Problems in Engineering*, v. 2010, pp. 1-25.
- LIANG, J., MENG, M., CHEN, Y., FULLMER, R. 2003, “Solving tough optimal control problems by network enabled optimization server (NEOS)”, *Technical report*, School of Engineering, Utah State University USA.
- LIU, Y., CAMERON, L.T., BHATIA, S.K., 2001, “A Wavelet-based Adaptive Technique for Adsorption Problems Involving Steep Gradients”, *Computers and Chemical Engineering*, v. 25, pp. 1611-1619.

- LOGIST, F., VALLERIO, M., HOUSKA, B., DIEHL, M., VAN INPE, J., 2012, “Multi-objective optimal control of chemical processes using ACADO toolkit”, *Computers and Chemical Engineering*, v. 37, n. 10, pp. 191-199.
- LOGSDON, J.S., BIEGLER, L.T., 1993, “A Relaxed Reduced Space SQP Strategy for Dynamic Optimization Problems”, *Computers & Chemical Engineering*, v. 17, n. 4, pp. 367-372.
- LONGUSKI, J.M., GUZMÁN, J.J., PRUSSING, J.E., 2014, *Optimal Control with Aerospace Applications*, Microcosm Press and Springer, NY, USA.
- LUISIER, F. & BLU, T., 2007, “A New SURE Approach to Image Denoising: Interscale Orthonormal Wavelet Thresholding”, *IEEE Transactions on Image Processing*, v. 16, n. 3, pp. 593-606.
- LUSS, R. 2002, *Iterative dynamic programming*, Chapman and Hall/CRC, USA.
- LUUS, R., ROSEN, O., 1991, “Application of Dynamic Programming to Final State Constrained Optimal Control Problems”, *Industrial & Engineering Chemical Research*, v. 30, n. 7, pp. 1525-1530.
- MALLAT S., 1989, “A theory for multiresolution signal decomposition: the wavelet representation”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, v. 2, n. 7, pp. 674-693.
- MATLAB User’s Guide. 2009, *The MathWorks*.
- MISITI, M., MISITI, Y., OPPENHEIN, G., POGGI, J.M., 2007, *Wavelets and their Applications*, London UK, ISTE Ltd.
- MORETTIN, P.A., 1999, *Ondas e Ondoletas Da Análise de Fourier à Análise de Ondoletas*, São Paulo, Editora da Universidade de São Paulo.
- NALUM, K., 2013, *Modeling and Dynamic Optimization in Oil Production*, MSc. Dissertation, Norwegian University of Science and Technology – NTNU, Norway.
- NASON, G. P., 1996, “Wavelet Shrinkage using Cross Validation”, *Journal of the Royal Statistical Society, Series B*, v. 58, pp.463–479.
- NOCEDAL, J., WRIGHT, S.J., 1999, *Numerical Optimization*, Springer Series in Operations Research, Sprig-Verlag, NY, USA.

- OROPEZA, M.V.C., 2007, *Avaliação do Método Wavelet-Galerkin Multi-malha para Caracterização das Propriedades de Petróleo e Subprodutos*, Dissertação de Mestrado, Escola Politécnica, Unversidade de São Paulo.
- PARK, S., RAMIREZ, W. F., 1988. "Optimal production of secreted protein in fed-batch reactors", *AIChE Journal*, v. 34, n. 8, pp. 1550–1558.
- PETZOLD, L. R., 1989, *Computing and Mathematics Research Division*, Lawrence Livermore National Laboratory, L316, PO Box 808, Livermore, CA 94559.
- PFEIFER, A.A., 2007, *Controle Ótimo de Sistemas Algébrico-Diferenciais com Flutuação do Índice Diferencial*, Dissertação de MSc., Universidade Federal de Uberlândia, Uberlândia, MG, Brasil.
- POLLARD, G. P., SARGENT, R. W. H., 1970, "Off line Computation of Optimum Controls for a Plate Distillation Column", *Automatica*, v. 6, pp. 59-76.
- PONTRYAGIN, L.S., BOLTYANSKII, V.G., GAMKRELIDZE, R.V., MISHCHENKO, E.F., 1962, *The Mathematical Theory of Optimal Processes*, John Wiley & Sons, USA.
- PRATA, A., OLDENBURG, J., KROLL, A., MARQUARDT, W., 2008, "Integrated scheduling and dynamic optimization of grade transitions for a continuous polymerization reactor", *Computers & Chemical Engineering*, v. 32, n. 3, pp. 463-476.
- PSHENICHNY, B.N., DANILIN, Y.M., 1978, *Numerical Methods in Extremal Problems*, Mir Publishers, Moscow, Russia.
- QUINTO, T., 2010, *Abordagem Algébrico-Diferencial da Otimização Dinâmica de Processos com Índice Flutuante*, Dissertação de Mestrado, Programa de Engenharia Química-COPPE-UFRJ.
- RAZZAGHI, M. E YOUSEFI, S., 2001, "Legendre Wavelets Method for the Solution of Nonlinear Problems in the Calculus of Variations", *Mathematical and Computer Modelling*, v. 34, pp. 45-54.
- RICE, R.G. DO, D.D, 1994, *Applied Mathematics and Modeling for Chemical Engineers*, John Wiley & Sons, Inc., USA.
- ROSAS-OREA, M.C.E., HERNADEZ-DIAZ, M., ALARCON-AQUINO, V., GUERRERO-OJEDA, L.G., 2005, "A Comparative Simulation Study of Wavelet

- Based Denoising Algorithms”, *Proceedings of the 15th International Conference on Electronics, Communications and Computers (CONIELECOMP 2005)*, pp. 125-130.
- ROSS, I.M., 1998, *A Beginner’s Guide to DIDO – A MATLAB Application Package for Solving Optimal Control Problems*, Elissar, LCC, USA.
- ROSS, I.M., 2006, “A Historical Introduction to the Covector Mapping Principle,” *Advances in the Astronautical Sciences*, v. 123, pp. 1257–1278.
- RUTQUIST, P.E., EDVALL, M.M., 2010, TOMLAB OPTIMIZATION, PROPT - Matlab Optimal Control Software.
- SANTOS, L.S., SECCHI, A.R., BISCAIA Jr., E.C., 2012, “Wavelet-Threshold Influence in Optimal Control Problems”, *Computer Aided Chemical Engineering*, v. 30, pp. 1222-1226.
- SANTOS, L.S., SECCHI, A.R., BISCAIA Jr., E.C., 2014, “Optimal Wavelet-Threshold Selection to Solve Dynamic Optimization Problems”, *Computer Aided Chemical Engineering*, v. 33, pp. 247-252.
- SCHEWEIGER, C.A., FLOUDAS, C.A., 1998, *MINOPT: A Modeling Language and Algorithm Framework for Linear, Mixed Integer, Nonlinear, Dynamic and Mixed-Integer Nonlinear Optimization*, Princeton University, USA.
- SCHLEGEL, M., 2004, *Adaptive discretization methods for the efficient solution of dynamic optimization problems*, PhD Thesis., Rheinisch-Westfaelische Technische Hochschule, Aachen, Deutschland.
- SCHLEGEL, M., STOCKMANN, K., BINDER, T., MARQUARDT, W., 2005, “Dynamic Optimization Using Adaptive Control Vector Parameterization”, v. 29, n. 8, pp. 1731-1751.
- SCHWARTZ, A., POLAK, E., CHEN, Y., 1997, *A Matlab Toolbox for Solving Optimal Control Problems*, Stanford University, CA, USA.
- SOARES, R.P., SECCHI, A.R., 2003, “EMSO: A new environment for modelling, simulation and optimization”, *Computer Aided Chemical Engineering*, v. 14, pp. 947-952.

- SOUZA, D.F.S., 2007, *Abordagem Algébrico-diferencial da Otimização Dinâmica de Processos*, Tese de DSc., Programa de Engenharia Química-COPPE-UFRJ, RJ, Brasil.
- SRINIVASAN, B., BONVIN, D., PALANKI, S., 2003b, "Dynamic optimization of batch processes: II. Role of Measurements in Handling Uncertainty". *Computers and Chemical Engineering*, v. 27, n. 1, pp. 27–44.
- SRINIVASAN, B., PALANKI, S., BONVIN, D., 2003a, "Dynamic Optimization of Batch Processes: I. Characterization of the Nominal Solution". *Computers and Chemical Engineering*, v. 27, n. 1, pp. 1–26.
- STEIN, C M., 1981, "Estimation of the Mean of a Multivariate Normal Distribution", in: *The Annals of Statistics*, v.9, n.6, pp. 1135–1151.
- VON STRYC, O., 1999, "User's Guide for DIRCOL A Direct Collocation Method for the Numerical Solution of Optimal Control Problems", Technische Universit at Darmstadt.
- TOSUKHOWONG, T., LEE, .H., 2009, "Approximate dynamic programming based optimal control applied to an integrated plant with a reactor and a distillation column with recycle", *AIChE Journal*, v. 55, n. 4, pp. 919-930.
- UPRETI, S.R., 2004, "A New Robust Technique for Optimal Control of Chemical Engineering Processes", *Computers & Chemical Engineering*, v. 28, n. 8, pp. 1325-1336.
- VASSILIADIS, V.S., R.W.H. SARGENT, C.C. PANTELIDES, 1994a, "Solution of a Class of Multistage Dynamic Optimization Problems. 1. Problems without Path Constraints", *Industrial & Engineering Chemistry Research*., v. 33, n. 9, pp. 2111-2122.
- VASSILIADIS, V.S., R.W.H. SARGENT, C.C. PANTELIDES, 1994b, "Solution of a Class of Multistage Dynamic Optimization Problems. 2. Problems with Path Constraints", *Industrial & Engineering Chemistry Research*., v. 33, n. 9, pp. 2123–2133.
- VASSILIADIS, V.S., PANTELIDES, C.C., SARGENT, R.W.H., 1994c, "Optimization of Discrete Charge Batch Reactors", *Computers and Chemical Engineering*, v. 18, pp. S415-S419.

- VERTELLI, M. & KOVAČEVIĆ, J., 2007, *Wavelets and Subband Coding*, New Jersey, USA, Prentice Hall PTR.
- VIEIRA, R. C, 2001, *Técnicas de Inicialização de Sistemas Algébrico-Diferenciais*, Tese de Doutorado, Programa de Engenharia Química-COPPE-UFRJ.
- VILLADSEN, J.V., STEWART, W.E., 1967, "Solution of Boundary Value Problems by Orthogonal Collocation", *Chemical Engineering Science*, v. 22, pp. 1483-1501.
- VON WATZDORF, R., URBAN, K., DAHMEN, W., MARQUARDT, W, 1996, "A wavelet-Galerkin method applied to separations processes", *Scientific computing in chemical engineering*, Berlin, Springer, pp. 246-252.
- WÄCHTER, T. & BIEGLER, L.T., 2006, "On the Implementation of a Primal-Dual Interior Point Filter Line Search Algorithm for Large-Scale Nonlinear Programming", *Mathematical Programming*, v. 106, n.1, pp. 25-57.
- WEISER, M., 2001, *Function Space Complementarity Methods for Optimal Control Problems*, Dsc. Thesis., Rheinisch-Westfaelische Technische Hochschule, Aachen, Deutschland.
- WIELAND, 2009, *Speech Signal Noise Reduction with Wavelets*, PhD Thesis., Universität Ulm.
- WOZNY, G., LI, P., 2000, "Planning and Optimization of Dynamic Plant Operation", *Applied Thermal Engineering*, v. 20, n. 15-16, pp. 1393-1407.
- WÜRTH, L., HANNEMANN, R., MARQUARDT W., 2009, "Neighboring-extremal updates for nonlinear model-predictive control and dynamic real-time optimization", *Journal of Process Control*, v. 19, pp. 1277-1288.
- XU, J., ZHANG, K., XU, M., ZHOU, Z., 2009, "An adaptive threshold method for image denoising based on wavelet domain", *Proc of SPIE*, v. 7495,
- YANG, S., YAO, X., 2013, *Evolutionary Computation for Dynamic Optimization Problems*, Springer, United Kingdom.