

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA DE  
AUTOMAÇÃO E SISTEMAS**

**JOÃO MARCELO ROMANO**

**INTEGRAÇÃO NUMÉRICA EFICIENTE EM CONTROLE AVANÇADO  
DE PROCESSOS: APLICAÇÃO EM UM SISTEMA DE COMPRESSÃO  
DE GÁS**

Florianópolis

2017



JOÃO MARCELO ROMANO

**INTEGRAÇÃO NUMÉRICA EFICIENTE EM CONTROLE AVANÇADO  
DE PROCESSOS: APLICAÇÃO EM UM SISTEMA DE COMPRESSÃO  
DE GÁS**

Tese submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas da Universidade Federal de Santa Catarina para a obtenção do Grau de Mestre em Engenharia de Controle e Automação.

Orientador: Prof. Julio Elias Normey-Rico, Dr.  
Coorientador: Prof. Agostinho Plucenio, Dr.

Florianópolis

2017

Ficha de identificação da obra elaborada pelo autor,  
através do Programa de Geração Automática da Biblioteca Universitária da UFSC.

Romano, João Marcelo  
INTEGRAÇÃO NUMÉRICA EFICIENTE EM CONTROLE  
AVANÇADO DE PROCESSOS: APLICAÇÃO EM UM SISTEMA DE  
COMPRESSÃO DE GÁS / João Marcelo Romano ;  
orientador, Julio Elias Normey-Rico, coorientador,  
Agustinho Plucenio, 2017.  
143 p.

Dissertação (mestrado) - Universidade Federal de  
Santa Catarina, Centro Tecnológico, Programa de Pós  
Graduação em Engenharia de Automação e Sistemas,  
Florianópolis, 2017.

Inclui referências.

1. Engenharia de Automação e Sistemas. 2.  
Problemas de Valor Inicial. 3. Equações Diferenciais  
e Algébricas. 4. Métodos Numéricos. 5. Sistemas de  
Compressão. I. Normey-Rico, Julio Elias . II.  
Plucenio, Agustinho . III. Universidade Federal de  
Santa Catarina. Programa de Pós-Graduação em  
Engenharia de Automação e Sistemas. IV. Título.

JOÃO MARCELO ROMANO

**INTEGRAÇÃO NUMÉRICA EFICIENTE EM CONTROLE AVANÇADO  
DE PROCESSOS: APLICAÇÃO EM UM SISTEMA DE COMPRESSÃO  
DE GÁS**

Esta Tese foi julgada adequada para a obtenção do Título de “Mestre em Engenharia de Controle e Automação”, e aprovada em sua forma final pelo Programa de Pós-Graduação em Engenharia de Automação e Sistemas.

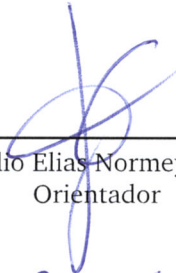
Florianópolis, 28 de Julho de 2017.



---

Prof. Daniel Ferreira Coutinho, Dr.  
Coordenador do Curso

**Banca Examinadora:**



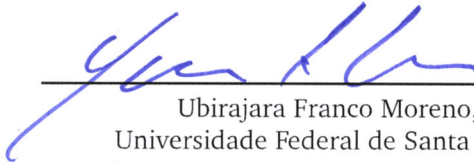
---

Prof. Julio Elias Normey-Rico, Dr.  
Orientador



---

Hector Bessa Silveira, Dr.  
Universidade Federal de Santa Catarina



---

Ubirajara Franco Moreno, Dr.  
Universidade Federal de Santa Catarina



---

Daniel Martins Lima, Dr.  
Universidade Federal de Santa Catarina



## AGRADECIMENTOS

Agradeço a minha família que sempre me incentiva e me apoia em todos os momentos. Ao professor Julio pelas orientações na condução da pesquisa e na revisão do texto. Ao meu coorientador e professor Agustinho (*in memoriam*) pelas inúmeras trocas de ideias, por me motivar a pesquisar e desenvolver soluções próprias para os problemas. Aos meus colegas de laboratório pelos bate-papos acadêmicos e momentos de descontração. Ao meu colega e amigo Eduardo pela ajuda com a edição do documento, e pelas inúmeras quedas n'água entre “expedientes”. Por fim, e não menos importante, à UFSC e ao PGEAS por me aceitarem no programa de mestrado e à PETROBRAS pelo apoio financeiro a este trabalho de pesquisa.





A beleza não está na partida nem na chegada,  
mas na travessia.

(Guimarães Rosa)



## RESUMO

Sistemas dinâmicos são geralmente descritos por um conjunto de equações diferenciais e algébricas ([Differential Algebraic Equation \(DAE\)](#)). Modelar sistemas dessa forma é fisicamente mais intuitivo, especialmente quando conhecimento suficiente sobre o processo está disponível, podendo-se empregar, por exemplo, parâmetros e dados reais de operação. Quanto à resolução ou obtenção da resposta destes modelos, esta pode se tornar computacionalmente custosa devido à quantidade de equações ou a propriedades intrínsecas da estrutura matemática destes tipos de equações. Esta dissertação investiga e propõe soluções mais eficientes para a tarefa de resolução numérica de sistemas [DAEs](#) quando utilizados por um controlador [PNMPC \(Practical Nonlinear Model Predictive Control\)](#) como modelo de predição. Também será estudado o desempenho computacional quando a própria planta precisa ser simulada por um sistema [DAE](#), isto é quando em fase de desenvolvimento dos algoritmos de controle. Como caso de estudo, a eficiência computacional da simulação de um sistema de controle avançado de compressão de gás será abordada. O trabalho apresenta e analisa as principais ferramentas existentes para resolução de sistemas [DAEs](#), e propõe três metodologias baseadas na manipulação e controle de parâmetros típicos da formulação de métodos numéricos. Os resultados obtidos com a aplicação destas metodologias são comparados com os obtidos por *solvers* comerciais tradicionais, através de indicadores da eficiência numérica e computacional da simulação para apontar as principais vantagens e desvantagens em utilizar as metodologias propostas.

**Palavras-chave:** Problemas de Valor Inicial. Equações Diferenciais e Algébricas. Métodos Numéricos. Sistemas Rígidos. Sistemas de Compressão.



## ABSTRACT

Dynamic systems are commonly described as a set of **DAE**. Modeling dynamic behaviors as **DAEs** systems is physically more intuitive especially when enough information is available. When it comes to the development of control algorithms, the simulation may be run hundreds of times in test phase, whether to investigate modifications on its parameters, or to evaluate the result of alternative control strategies. Thus, having a fast robust solver becomes very important. Concerning the process of solving a **DAE** system, this task may be computational burden due to a high number of equations or regarding to the mathematical structure of the system, that is a algebraic equations constraining the differential set. This dissertation proposes more efficient solving solutions to the numerical integration of a complex **DAE** system used as predictive model into a **PNMPC (Practical Nonlinear Model Predictive Control)** control algorithm. In addition, it is studied the efficiency of the solver when the plant is modeled by a **DAE** system, focusing on the development of control algorithms. As a case study, the computational efficiency of an advanced gas compression control system is addressed. The research presents and analyses the main existing tools to the numerical integration addressed to the **DAE** systems, and proposes three methodologies based on the manipulation and control of typical variables on the numerical methods formulas. The results obtained with the application of these methodologies into the simulation are compared with some traditional commercial solvers, and efficiency indicators of the computational performance are used to point out the main advantages and disadvantages of using the proposed methodologies.

**Keywords:** Initial Value Problems. Differential Algebraic Equations. Numerical Methods. Stiff Systems. Compression Systems.



## LISTA DE FIGURAS

1.1	Processamento primário de fluidos. (Fonte: [1]). . . . .	23
1.2	Curva característica de um estágio de compressão. (Fonte: [2]). . . . .	24
1.3	Compressor com uma válvula de reciclo. . . . .	25
1.4	Diagrama de blocos do algoritmo MPC. (Fonte: [3]). . . . .	26
2.1	Tabela de Butcher. . . . .	37
2.2	O Método de colocação. (Fonte: [4]). . . . .	39
2.3	O fenômeno da rigidez. (Fonte: [5]). . . . .	50
3.1	Estação de compressão de uma unidade de processamento de gás. . . . .	53
3.2	Estágio de compressão e seus componentes. . . . .	54
3.3	Mapa de compressão. A linha que conecta as curvas corresponde ao limite de operação do estágio, denominada linha de <i>surge</i> . (Fonte: [2]). . . . .	58
3.4	Estágio de compressão, identificando-se as vazões que o circulam. (Fonte: [6]). . . . .	60
3.5	Mapa de compressão indicando a linha de surge e a linha de segurança. . . . .	67
4.1	Exemplificação do método ponderado. . . . .	77
4.2	Malha de controle do erro local. . . . .	83
4.3	Malha de controle PI final. . . . .	86
4.4	Malha de controle do erro local por cancelamento de não linearidade. . . . .	89
4.5	Curva ( $h_{n+1} = \Phi^{-1}(\bar{e}_{n+1})$ ) relacionando o erro local ao tamanho do passo. . . . .	90
4.6	Resposta ao degrau (sinal em azul) da planta <b>P</b> à entrada $\bar{e}_{n+1}$ (em preto) e o correspondente passo de integração $h_{n+1}$ (em vermelho) . . . . .	91
4.7	Sobreposição da resposta temporal de <b>P</b> (em azul) e o correspondente modelo identificado (em vermelho). . . . .	92
4.8	Malha de controle IMC de erro local. . . . .	93
5.1	Cenário 2: método mPV: dinâmicas do erro local $e_{n+1}$ , tamanho de passo $h_{n+1}$ e ponderação $\theta_{n+1}$ . . . . .	108
5.2	Cenário 2: método mPI: dinâmicas do erro local $e_{n+1}$ e do tamanho de passo $h_{n+1}$ . . . . .	109
5.3	Método mNL: dinâmicas do erro local $e_{n+1}$ e do tamanho de passo $h_{n+1}$ . . . . .	110





## LISTA DE TABELAS

5.1	Métodos numéricos utilizados para comparação. . . . .	101
5.2	Parametrização método ponderação variável. . . . .	103
5.3	Parametrização método controle PI do erro local. . . . .	103
5.4	Parametrização método controle do erro local por cancelamento de não linearidade. . . . .	103
5.5	Cenário 1: resultados de simulação para uma tolerância de $\mathcal{E}_1 = 10^{-3}$ . . . . .	104
5.6	Cenário 1: resultados de simulação para uma tolerância de $\mathcal{E}_2 = 10^{-6}$ . . . . .	104
5.7	Cenário 2: Resultados de simulação para uma tolerância de $\mathcal{E}_1 = 10^{-3}$ . . . . .	106
5.8	Cenário 2: resultados de simulação para uma tolerância de $\mathcal{E}_2 = 10^{-6}$ . . . . .	106



## LISTA DE SIGLAS E ABREVIATURAS

<b>BDF</b> Backward Differences Formulas . . . . .	42, 44, 46, 51
<b>BVP</b> Boundary Value Problem . . . . .	117
<b>DAE</b> Differential Algebraic Equation . . . . .	ix, xi, 27–35, 42, 49, 51, 69, 70, 73, 75, 80–83, 85, 88–90, 93, 95, 97, 99–101, 104–106, 111, 113–117
<b>GOR</b> Gas Oil Ratio . . . . .	25
<b>IMC</b> Internal Model Control . . . . .	xiii, 92–95, 114, 116
<b>LGR</b> Legendre-Gauss-Radau . . . . .	39, 101
<b>mNL</b> método controle por cancelamento de Nao Linearidade	xiii, 104, 106, 110, 111
<b>MPC</b> Model Predictive Control . . . . .	xiii, 26–28, 123
<b>mPI</b> método controle PI do erro local	xiii, 104–107, 109–111, 115
<b>mPV</b> método Ponderação Variável com controle do tamanho de passo . . . . .	xiii, 104, 106–108, 111
<b>NMPC</b> Nonlinear Model Predictive Control . . . . .	27
<b>ODE</b> Ordinary Differential Equation	31–33, 35, 38, 40, 41, 45, 46, 50, 69, 73, 86
<b>PI</b> Proporcional Integral	xiii, xv, 64–66, 68, 85, 86, 94, 103, 114, 115
<b>PID</b> Proporcional Integral Derivativo . . . . .	25, 55, 63
<b>PNMPC</b> Practical Nonlinear Model Predictive Control . . .	ix, xi, 27–30, 89, 99, 100, 106, 111, 113, 115, 123, 124, 126, 128
<b>PVI</b> Problema de Valor Inicial .	32, 34, 42, 44–46, 50, 51, 69, 71, 73, 83, 88, 106, 113
<b>RK2</b> Runge-Kutta de ordem 2 . . . . .	43
<b>RK3</b> Runge-Kutta de ordem 3 . . . . .	43



# SUMÁRIO

<b>1</b>	<b>Introdução</b>	<b>23</b>
1.1	Compressão de gás e controle avançado em unidades de produção de petróleo . . . . .	23
1.2	Motivação . . . . .	27
1.3	Objetivos e contribuições . . . . .	29
1.3.1	Estrutura da dissertação . . . . .	30
<b>2</b>	<b>Métodos numéricos para a resolução de modelos</b>	<b>31</b>
2.1	O problema de valor inicial . . . . .	31
2.2	Métodos numéricos para a resolução de sistemas DAEs de índice unitário . . . . .	34
2.2.1	Métodos ponderados e métodos de Runge-Kutta	35
2.2.2	Métodos de colocação . . . . .	38
2.2.3	Métodos de múltiplos passos e de diferenças finitas . . . . .	41
2.3	Propriedades numéricas . . . . .	43
2.3.1	Erro local de aproximação . . . . .	43
2.3.2	Estabilidade . . . . .	44
2.3.2.1	Estabilidade dos métodos de passo único . . . . .	45
2.3.2.2	Estabilidade dos métodos de múltiplos passos . . . . .	47
2.3.3	Controle do tamanho do passo de integração . . . . .	48
2.3.4	Equações rígidas . . . . .	49
2.4	Comentários finais . . . . .	51
<b>3</b>	<b>Modelagem de uma estação de compressão controlada de gás</b>	<b>53</b>
3.1	Descrição do sistema . . . . .	53
3.2	Modelagem matemática fenomenológica . . . . .	56
3.2.1	Modelagem termodinâmica . . . . .	56
3.2.2	Modelagem do controle regulatório . . . . .	63
3.2.2.1	Controle da pressão de sucção na entrada das linhas de compressão . . . . .	63
3.2.2.2	Controle da temperatura de sucção em cada estágio . . . . .	64
3.2.2.3	Controle da vazão das linhas de injeção . . . . .	65
3.2.2.4	Controle <i>anti-surge</i> . . . . .	66
3.3	Complexidade numérica do modelo . . . . .	69
3.4	Comentários finais . . . . .	70

<b>4</b>	<b>Contribuições quanto à eficiência computacional de métodos numéricos</b>	<b>73</b>
4.1	Escolhendo fórmulas para calcular aproximações e estimar o erro local . . . . .	73
4.1.1	Estabilidade do método ponderado . . . . .	75
4.1.2	Estimativa do erro local . . . . .	76
4.2	Método ponderação variável com controle do tamanho do passo . . . . .	76
4.3	Método controle PI do erro local . . . . .	80
4.3.1	Análise do problema de controle . . . . .	82
4.3.2	Projeto do algoritmo de controle PI filtrado com <i>anti wind-up</i> . . . . .	84
4.3.3	Considerações a cerca da estabilidade do método controle PI do erro local . . . . .	86
4.4	Método de controle do erro Local por cancelamento de não linearidade . . . . .	88
4.4.1	Identificação do modelo discreto . . . . .	89
4.4.2	Projeto e sintonia do controlador . . . . .	92
4.4.3	Análise da estabilidade do controlador IMC de erro local . . . . .	95
4.5	Comentários finais . . . . .	97
<b>5</b>	<b>Resultados e discussões</b>	<b>99</b>
5.1	Cenários de simulação . . . . .	99
5.1.1	Métodos utilizados para comparação e análise . . . . .	101
5.1.2	Parametrização dos métodos numéricos desenvolvidos . . . . .	102
5.1.3	Indicadores de desempenho . . . . .	102
5.2	Resultados de simulação . . . . .	104
5.2.1	Primeiro Cenário . . . . .	104
5.2.2	Segundo Cenário . . . . .	106
5.3	Comentários finais . . . . .	111
<b>6</b>	<b>Conclusões</b>	<b>113</b>
6.1	Trabalhos futuros . . . . .	116
	<b>Referências</b>	<b>119</b>
	<b>APÊNDICE A - Controle preditivo PNMPC</b>	<b>123</b>
	Obtenção das matrizes G e F . . . . .	124
	Obtenção da ação de controle . . . . .	126

<b>APÊNDICE B - Parâmetros de simulação</b>	<b>129</b>
<b>APÊNDICE C - Equações diferenciais e algébricas</b>	<b>137</b>







A fim de aumentar a eficiência deste processo, cada compressor é composto, em geral, por mais de um estágio de compressão. O comportamento e o desempenho de operação de cada estágio são governados pelos mapas de compressão, curvas estáticas como as da Figura 1.2, que são fornecidas pelos fabricantes e apresentam, por exemplo, a relação entre o ganho de pressão fornecido pelo compressor a uma determinada vazão de gás que o percorre. Cada curva corresponde a uma velocidade de rotação constante da máquina.

Naturalmente, tratando-se de uma forma gráfica, estas curvas são limitadas em termos de representação de um equipamento com tantos parâmetros e variáveis. Entretanto, em condições normais de operação, e dentro de uma faixa limitada de velocidade de rotação, pode-se obter com boa qualidade de informação o ganho de pressão, ou a pressão de descarga, a qual o compressor entrega determinada vazão de gás aspirada em sua sucção.

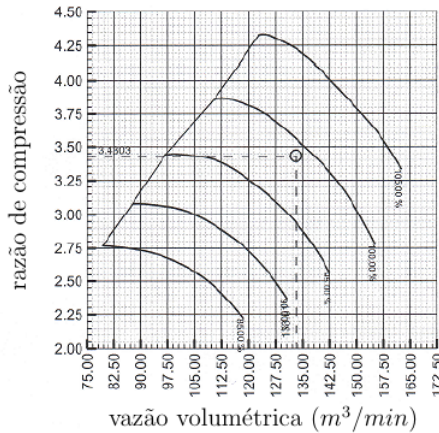


Figura 1.2: Curva característica de um estágio de compressão. (Fonte: [2]).

Uma característica importante dessas máquinas é a ocorrência de um fenômeno denominado *surge*, que ocorre quando a vazão de gás no compressor é muito baixa ou decai para um ponto de inclinação positiva de sua curva de operação. Esse fenômeno provoca uma operação bastante instável do compressor, com a ocorrência de flutuações de pressão na máquina e a reversão do sentido da vazão da descarga para a sucção.

A fim de evitar esse fenômeno, os fabricantes disponibilizam nos compressores uma válvula denominada *anti-surge*, ou de reciclo, como mostra a Figura 1.3, e cuja abertura redireciona parte do gás comprimido para a sucção da máquina através de um caminho *bypass*, aumentando assim sua vazão quando necessário.

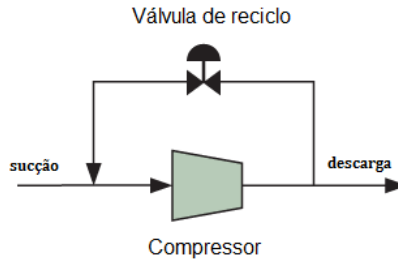


Figura 1.3: Compressor com uma válvula de reciclo.

Embora os compressores tenham dispositivos de segurança como esse implementados por seus fabricantes, é possível que uma falha (ou queda) de um compressor leve ao completo desligamento de uma unidade de produção [2].

A produção de gás em uma plataforma não diminui automaticamente com a diminuição da disponibilidade de gás comprimido. Muitos poços produzem com uma alta razão gás-óleo – no inglês *Gas Oil Ratio (GOR)*. Estes irão produzir gás em uma taxa maior do que aquela que pode ser comprimida na estação. Isso significa que parte do gás produzido terá de ser enviada para um queimador (denominado *flare*) até que uma solução seja tomada para resolver este problema. Menos gás comprimido significa menos gás para injeção e para exportação, e a possibilidade de serem emitidas multas pela agência reguladora devido a uma queima de gás fora dos limites impostos.

Assim, com o objetivo de manter uma operação livre de perigos, as estações de compressão apresentam uma série de malhas de controle regulatório implementadas com controladores *Proporcional Integral Derivativo (PID)*, que atuam, por exemplo, nas válvulas de reciclo para evitar o *surge*, no controle da temperatura do gás para evitar o sobreaquecimento das máquinas, ou para manter a pressão na sucção ou e na descarga de gás em níveis seguros.

Além de garantir a segurança dos processos e instalações de produção, é de extrema importância a minimização de perdas com gastos energéticos indesejados ou desnecessários, bem como a maxi-

mização do desempenho e a eficiência de operação dos equipamentos. Das tecnologias que buscam suprir esta necessidade, encontram-se as técnicas de controle avançado, tal como o controle **MPC (Model Predictive Control)** que se destaca por ser uma das técnicas de controle moderno mais potentes e que vêm obtendo muito êxito em aplicações na indústria de petróleo e gás, inclusive no processamento primário de fluidos tal como nos trabalhos [2, 3, 7, 8].

Conceitualmente, os controladores **Model Predictive Control (MPC)** utilizam-se do conhecimento do processo a ser controlado, representado por um modelo dinâmico, para prever em um horizonte de tempo finito, qual seria a resposta do sistema a uma determinada ação de controle, antes mesmo de aplicá-la. De posse de um critério de performance e das restrições do processo, estes controladores são capazes de determinar um plano de ação a partir da otimização deste critério de desempenho, cujo resultado apresenta a melhor sequência de ações de controle a serem tomadas, com o menor esforço possível.

Em geral, os algoritmos de controle **MPC** possuem alguns elementos em comum: um modelo do processo para calcular a predição, uma função custo com determinado objetivo de desempenho, e um procedimento para calcular a ação de controle [9]. As diferenças entre os diversos algoritmos se devem, basicamente, à forma de representar e escolher tais elementos. A Figura 1.4 apresenta a estrutura geral de um algoritmo **MPC**, onde pode ser observado seu funcionamento.

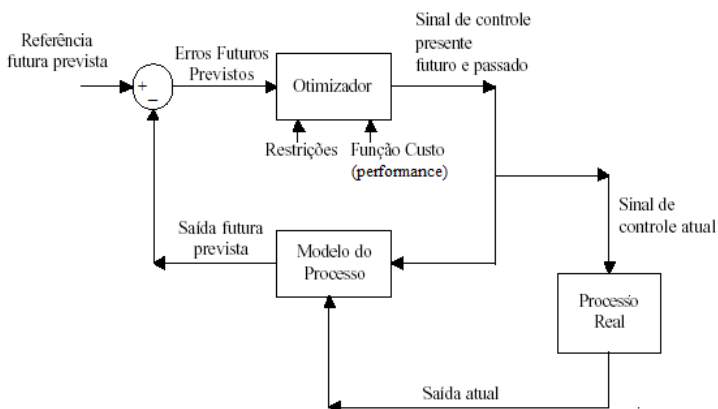


Figura 1.4: Diagrama de blocos do algoritmo **MPC**. (Fonte: [3]).

O modelo de predição do processo é o elemento mais importante dentro do controlador, e deve ser capaz de representar adequadamente a dinâmica do processo, pois a lei de controle é totalmente baseada nele. Modelos lineares (MPC linear) são muito comuns na prática, pois as predições podem ser calculadas utilizando-se o princípio de superposição ou soma de respostas: a resposta forçada, que compreende a saída do sistema como resultado de mudanças nas ações de controle futuras, e a resposta livre, saída do sistema conseqüente às entradas passadas que ocorreram até o último ciclo de amostragem. Como vantagem, modelos lineares podem ser utilizados em plantas multivariáveis sem acrescentar complexidade, e os problemas de otimização associados ao algoritmo de controle podem ser resolvidos por técnicas de otimização lineares, numericamente mais simples de resolver.

Todavia, quando os processos possuem uma dinâmica muito não linear ou quando a faixa de operação é variável, necessita-se tomar-se em conta um modelo não linear de predição no projeto do controle, de forma a permitir que se mantenha o desempenho desejado para o sistema em malha fechada [3]. Isto tem motivado o estudo teórico e aplicado do chamando *Nonlinear Model Predictive Control* (NMPC), ao qual a técnica *PNMPC* (*Practical Nonlinear Model Predictive Control*) proposta por Plucenio [10] é enquadrada, e tem apresentado bons resultados em se tratando de processos tipicamente não lineares encontrados na indústria de petróleo e gás, tais como os estudados em [3, 11, 12], e cuja eficiência computacional e numérica de seu algoritmo será abordada neste trabalho.

## 1.2 MOTIVAÇÃO

A fim de garantir a fidelidade do modelo não linear ao processo real e assegurar um bom desempenho dos sistemas de controle NMPC, é de comum interesse a utilização de modelos de predição fenomenológicos, em especial quando conhecimento suficiente sobre o processo a controlar é disponibilizado, podendo-se empregar, por exemplo, parâmetros e dados reais de operação. Este tipo de modelo descreve o sistema a partir de um conjunto de equações diferenciais e algébricas – em inglês DAEs que, embora representem matematicamente o sistema com alto rigor, para modelos não lineares complexos agrega-se alto custo computacional devido (a) à sua resolução, isto é, a integração numérica de um sistema DAE para obter-se a resposta do modelo de predição e (b)

à solução de um problema de otimização não linear para obter-se a lei de controle a cada ciclo de amostragem, dificultando assim a obtenção de um algoritmo implementável em tempo real.

O algoritmo de controle **Practical Nonlinear Model Predictive Control (PNMPC)** resolve o problema de eficiência computacional associado à (b) propondo uma solução alternativa de calcular as previsões através de uma representação linear aproximada das saídas em relação aos incrementos de controle futuros, obtendo-se assim uma resposta forçada que depende linearmente destes controles futuros. Desta maneira, consegue-se representar as previsões como uma soma de respostas livre e forçada, tal como no caso linear, mas tomando o próprio modelo não linear para computá-las. Assim é possível fazer uso de ferramentas de otimização linear para encontrar a lei de controle.<sup>1</sup>

Com relação à resolução numérica do modelo **DAE**, encontram-se metodologias bem definidas na literatura para resolver este problema, e uma grande variedade de pacotes e *solvers* comerciais disponíveis. Todavia, existem propriedades numéricas intrínsecas à estrutura matemática de alguns sistemas que fazem determinados métodos de integração ou serem muito ineficientes em relação a outros, ou não serem capazes de entregar uma resposta do sistema com o nível de precisão desejada.

Desta forma, a escolha e a aplicação do método de resolução do modelo podem se tornar tarefas mais trabalhosas do que a própria sintonia do controlador. Além disso, na simulação de algoritmos **MPC** em geral, além de serem usados para calcular previsões em todo instante do horizonte de predição definido, modelos dinâmicos do processo podem ser aplicados para realizar a estimação de variáveis não medidas, ou ainda para simular a planta. A fim de obter-se todas essas respostas eficientemente, um algoritmo de integração numérica ágil e robusto necessita ser utilizado.

A literatura é vasta quanto a metodologias numéricas para resolução de sistemas **DAEs**, como em [13, 14, 15, 16, 17], mas há escassez com relação às que utilizam a teoria de controle, como em [18], para melhorar o desempenho e a eficiência computacional da obtenção da resposta de sistemas complexos e rígidos.

Este trabalho investiga e propõe soluções mais eficientes para a tarefa de integração numérica de um modelo fenomenológico complexo em um algoritmo de controle **PNMPC**, a partir da aná-

---

<sup>1</sup>Detalhes do funcionamento e do cálculo da lei de controle do algoritmo **PNMPC** serão dados no **Apêndice A**.

lise de propriedades numéricas de sistemas **DAEs**, das ferramentas utilizadas pelos métodos de resolução existentes e da aplicação da teoria de controle. Como caso de estudo é abordada a eficiência computacional de um algoritmo de controle **PNMPC** de compressão de gás.

### 1.3 OBJETIVOS E CONTRIBUIÇÕES

O principal objetivo desta dissertação é desenvolver algoritmos de integração numérica robustos e eficientes, aplicando-se teoria de controle, para diminuir o custo computacional tomado com a obtenção da resposta de modelos dinâmicos fenomenológicos, isto é, com chamadas de *solvers* de integração numérica em algoritmos de controle preditivo. O caso de estudo abordado contempla um sistema de controle avançado **PNMPC** aplicado a uma estação controlada de compressão de gás, composta por dois compressores centrífugos de três estágios cada, presente em uma unidade de produção de petróleo. O modelo utilizado para a estação é representado por um complexo sistema de equações diferenciais rígidas e algébricas.

O trabalho apresenta e analisa as principais ferramentas existentes na teoria de integração numérica de sistemas **DAEs** complexos, e propõe três métodos diferentes baseados na manipulação e controle de parâmetros da fórmula de aproximação utilizada. Os resultados obtidos com a aplicação dos algoritmos destes métodos são comparados com os obtidos por *solvers* comerciais tradicionais, através de indicadores da eficiência numérica e computacional, para apontar as principais vantagens e desvantagens em utilizar os métodos propostos.

As principais contribuições desta dissertação podem ser sintetizadas como:

- O desenvolvimento de algoritmos robustos e eficientes para realizar a tarefa de integração numérica de um modelo **DAE** complexo utilizado por um sistema de controle avançado de compressão de gás;
- O uso da teoria de controle para melhorar o desempenho numérico e computacional de algoritmos de integração;
- Uma análise comparativa da eficiência computacional das principais ferramentas comerciais disponíveis para a integração numérica de sistemas **DAEs**.

Uma consideração importante a ser tomada neste trabalho de pesquisa está no fato de que todas as análises são realizadas utilizando-se o mesmo software de simulação, portanto os algoritmos desenvolvidos são executados sob as mesmas condições computacionais. Indiscutivelmente *solvers* de integração comerciais dispõem-se de recursos de processamento e compilação mais eficientes. Entretanto, este tipo de informação não é, em geral, disponibilizada, impossibilitando uma análise da eficiência computacional dos *solvers* também quanto ao uso desses recursos. Mesmo assim, muitas vezes estas técnicas não são eficientes em resolver alguns tipos de problemas **DAEs**.

Neste trabalho, a comparação de ferramentas computacionais é tratada somente com relação à eficiência numérica, isto é, sem considerar recursos de processamento e compilação adicionais. Desta forma, procurou-se desenvolver algoritmos de integração estruturados numericamente de forma simples, possibilitando fácil extensão a outros softwares e linguagens de programação, bem como a possíveis recursos computacionais que agreguem melhora a sua eficiência.

### 1.3.1 Estrutura da dissertação

Esta dissertação é dividida em seis capítulos e três apêndices. O **Capítulo 2** descreve o problema de integração numérica de sistemas **DAEs**, revisa as principais metodologias existentes na literatura para resolvê-lo, e levanta propriedades numéricas importantes na implementação de métodos numéricos, como a rigidez e a estabilidade. O **Capítulo 3** traz o modelo fenomenológico da estação controlada de compressão de gás, e faz uma análise quantitativa do sistema quanto a sua estrutura numérica e complexidade.

O **Capítulo 4** descreve as metodologias propostas levantando as principais características de cada uma delas que contribuem para o aumento de eficiência computacional da simulação. O **Capítulo 5** apresenta os resultados de simulação e analisa a eficiência computacional das metodologias aplicadas em comparação a *solvers* tradicionais de resolução de sistemas **DAEs**. Conclusões desta dissertação são apresentadas no **Capítulo 6** com comentários sobre os resultados alcançados e perspectivas de trabalhos futuros.

Finalmente, o **Apêndice A**, o **Apêndice B**, e o **Apêndice C** apresentam respectivamente, os detalhes da técnica de controle **PNMPC**, os parâmetros usados nas simulações, e as equações utilizadas para modelar a estação de compressão de gás.



## 2 MÉTODOS NUMÉRICOS PARA A RESOLUÇÃO DE MODELOS

Este capítulo aborda matematicamente o problema da integração numérica de modelos contínuos no tempo, descritos por sistemas de equações diferenciais e algébricas. Em seguida, realiza uma sucinta revisão das metodologias tradicionais de resolução, e por fim apresenta algumas propriedades que oferecem alguns obstáculos à determinação numérica da solução do problema.

### 2.1 O PROBLEMA DE VALOR INICIAL

Sistemas dinâmicos são normalmente descritos por um conjunto de equações diferenciais ordinárias – em inglês **Ordinary Differential Equations (ODEs)**, e um conjunto de equações algébricas, formando assim um sistema de equações diferenciais e algébricas, ou sistema **DAE (Differential Algebraic Equation)**, que também pode ser visto como um sistema de equações diferenciais com restrições algébricas [4]. É possível tornar um sistema **DAE** em um sistema **ODE** puro através da diferenciação das equações algébricas. O número de vezes que estas precisam ser diferenciadas para que se obtenha equações diferenciais ordinárias para todas as variáveis algébricas do sistema é denominado índice de um sistema **DAE** [17].

Considere, por exemplo, um caso simples de sistema **DAE**, definido pelo sistema (2.1) abaixo:

$$\frac{dw}{dt} = \dot{w} = -w - 1 \quad (2.1a)$$

$$0 = (w + 1)z + 2 \quad (2.1b)$$

Diferenciando-se a equação (2.1b) com relação ao tempo, obtém-se a derivada da variável  $z$  explicitamente:

$$\frac{d}{dt}[0] = \frac{d}{dt}[(w + 1)z + 2] \quad (2.2a)$$

$$0 = \dot{w}z + (w + 1)\dot{z} \quad (2.2b)$$

$$\dot{z} = -\frac{\dot{w}z}{(w + 1)} \quad (2.2c)$$

Substituindo (2.1a) na equação (2.2c) acima, obtém-se o seguinte sistema **ODE**:

$$\dot{w} = -w - 1 \quad (2.3a)$$

$$\dot{z} = z \quad (2.3b)$$

Como a variável algébrica  $z$  necessitou ser diferenciada no tempo apenas uma vez para obter-se uma equação diferencial, o sistema DAE é dito possuir índice 1. O processo de diferenciação de uma equação algébrica é denominado redução de índice, cuja denominação é utilizada para classificar o sistema com relação a constituição de seu conjunto de equações algébricas.

A modelagem fenomenológica descrita por sistemas DAEs é, em geral, fisicamente mais intuitiva em relação à descrição por um sistema ODE. Nota-se no sistema anterior, que poderia-se simplesmente substituir a equação (2.1b) em (2.1a), obtendo-se uma única ODE. Entretanto, a diferenciação das equações algébricas nem sempre é uma operação tão trivial, e esta operação pode causar perda do significado físico das variáveis e das relações envolvidas. [19].

Considere um sistema dinâmico contínuo em malha fechada representado por um sistema DAE. Quando a resolução deste sistema está sujeita a um determinado conjunto de condições iniciais, isto é, a evolução da dinâmica do sistema depende dos valores iniciais dos estados, o problema de encontrar a resposta<sup>1</sup> deste sistema em um determinado intervalo  $t \in T = [t_0, t_f]$  é denominado **Problema de Valor Inicial (PVI)**. Matematicamente o problema de interesse é descrito pelo sistema (2.4), onde o sistema DAE é exposto em uma forma semi-explícita.

$$\mathbf{F}(\mathbf{x}(t), T) = \begin{cases} \dot{\mathbf{x}} = \frac{d\mathbf{x}}{dt} = \mathbf{f}(t, \mathbf{x}(t), \mathbf{p}(t)) & (2.4a) \\ 0 = \mathbf{g}(\mathbf{x}(t), \mathbf{p}(t)) & (2.4b) \\ \mathbf{x}(t_0) = \mathbf{x}_0 & (2.4c) \end{cases}$$

onde  $\mathbf{x} \in \mathbb{R}^{n_x}$  é o vetor-solução de interesse com os  $n_x$  estados do sistema,  $\mathbf{p} \in \mathbb{R}^{n_p}$  é o vetor com as  $n_p$  variáveis algébricas e  $\mathbf{x}(t_0) = \mathbf{x}_0 \in \mathbb{R}^{n_x}$  é o vetor com as  $n_x$  condições iniciais. As funções-vetores  $\mathbf{f}$  e  $\mathbf{g}$  descrevem respectivamente as equações dinâmicas, e algébricas não lineares do sistema.

Em geral, se o gradiente da função  $\mathbf{g}$  em relação à  $\mathbf{p}$  ( $\nabla_{\mathbf{p}}\mathbf{g}$ ) é não singular, é possível encontrar uma relação dinâmica que descreva  $\dot{\mathbf{p}}$  em função de  $t$  e  $\mathbf{x}$ , através da redução de índice [20].

Em se tratando de sistemas DAEs de índice unitário, existe uma importante propriedade que consiste na possibilidade de utilizar métodos numéricos para resolver primeiramente as ODEs do sis-

---

<sup>1</sup>Por resposta do sistema entende-se a saída do sistema, ou um subconjunto de estados do sistema.

tema, para dada condição inicial, e utilizar a solução obtida para solucionar as equações algébricas [16], ou seja, o sistema de equações algébricas é visto como uma restrição à resolução do sistema de equações diferenciais [17, 20].

Tomando-se o exemplo do sistema (2.1), para uma condição inicial  $w_0 = -1$ , primeiramente resolve-se a ODE dada por (2.1a). Integrando-se analiticamente, por exemplo, para o instante  $t = 2$  segundos, chega-se a:

$$w(t) = w_0 \cdot e^{-t} - 1 \quad (2.5a)$$

$$w(2) = -e^{-2} - 1 \quad (2.5b)$$

$$w(2) = -1, 1353 \quad (2.5c)$$

substituindo o resultado na equação algébrica (2.1b):

$$0 = (w(2) + 1)z(2) + 2 \quad (2.6a)$$

$$0 = (-1, 1353 + 1)z(2) + 2 \quad (2.6b)$$

$$z(2) = 14, 7820 \quad (2.6c)$$

Dessa forma a solução do sistema DAE (2.1) no instante  $t = 2$  é o par  $(w(2), z(2)) = (-1, 1353; 14, 7820)$ .

Algumas considerações são aqui assumidas como verdade para este trabalho:

- (i) O problema (2.4) possui solução  $\mathbf{x}(t)$  única, contínua e diferenciável;
- (ii) As restrições algébricas são consistentes, isto é, a condição  $0 = \mathbf{g}(\mathbf{x}(t), \mathbf{p}(t))$  é assegurada, e  $\mathbf{p}(t)$  possui solução única para todo instante  $t$ .

Dado que o interesse deste trabalho corresponde a um sistema de compressão, cujo modelo é descrito por um sistema DAE de índice 1, este capítulo abordará o estudo de técnicas e métodos para resolver equações deste tipo de sistema, isto é métodos de resolução de ODEs, utilizando-se as soluções obtidas para calcular um sistema de equações não lineares a cada passo dado pelas equações algébricas do modelo [16, 21, 22].

## 2.2 MÉTODOS NUMÉRICOS PARA A RESOLUÇÃO DE SISTEMAS DAES DE ÍNDICE UNITÁRIO

Esta seção traz de maneira sucinta algumas das metodologias tradicionais de resolução de sistemas DAEs, dando enfoque às que são disponibilizadas em *solvers* comerciais como os do pacote *ode* do software MATLAB 2015a<sup>®</sup>. Primeiramente, são tomadas algumas definições.

Seja  $T = [t_0, t_f]$  o intervalo de integração dividido em  $N$  subintervalos de tamanho  $h_i$ , tal que  $t_i = t_{i-1} + h_i, \forall i \in [1, \dots, N]$ . À grandeza  $h$  dá-se o nome de tamanho do passo de integração, ou apenas “tamanho do passo” (de *stepsize*). A solução de interesse do PVI corresponde a  $\mathbf{x}(t_i)$  e deve satisfazer determinada condição de precisão.

Em se tratando de computação numérica, é importante lembrar que um erro local  $e(t_i)$  é sempre cometido em toda aproximação computada  $\mathbf{x}(t_i)$ . Como a solução analítica é desconhecida, necessita-se tomar uma estimativa para quantificar este erro, que deve ser limitado. Existem algumas maneiras de estimar-se este erro, algumas das quais serão apresentadas na Seção 2.3.1.

A fim de que o algoritmo de integração seja realizável, define-se critérios de parada para o cálculo de  $\mathbf{x}(t)$  em cada instante do intervalo de integração  $t_i \in T$ . Deve-se fornecer uma aproximação  $\mathbf{x}(t_i)$  sempre que: ou (a) o erro local satisfaça uma tolerância de precisão  $\mathcal{E}$  pré-especificada, isto é,  $e(t_i) < \mathcal{E}$ ; ou (b) caso o número  $j, j \in [1, N_{max}]$ , de passos tomados pelo algoritmo do método para calcular  $\mathbf{x}(t_i)$  exceda um número máximo de iterações “ $N_{max}$ ” também pré-definido.

Em resumo, um método numérico deve apresentar em seu algoritmo pelo menos os seguintes elementos em cada passo  $j$ , para computar uma aproximação no instante  $t_i$ :

- (i) Uma fórmula para computar cada aproximação, isto é,  $\mathbf{x}^{(j)}(t_i)$ ;
- (ii) Uma fórmula para estimar o erro local  $e^{(j)}(t_i)$  cometido pela aproximação computada;
- (iii) Uma regra ou critério de parada para decidir aceitar a solução aproximada proposta, dada por  $e^{(j)}(t_i) < \mathcal{E}$  ou  $j > N_{max}$ ;
- (iv) Caso a aproximação não seja aceita, mas o número máximo de passos não for violado,  $j \leq N_{max}$ , tomar então alguma atitude afim de reduzir o erro local, para que na próxima iteração ( $j + 1$ ), satisfaça-se a condição  $e^{(j+1)}(t_i) < \mathcal{E}$ .

Os métodos apresentados a seguir diferenciam-se justamente pelas formas de obtenção da aproximação à solução, isto é, quanto ao elemento (i).

### 2.2.1 Métodos ponderados e métodos de Runge-Kutta

Os métodos aqui expostos são chamados métodos de passo único (*one step methods*), pois a solução local depende apenas da computada no passo imediatamente anterior [20].

A fim de melhorar a eficiência, o tamanho do passo de integração pode ser variável. Para evitar equações com muitos índices, adota-se neste trabalho a notação  $n = t_i$ , tal que para dado passo de tamanho  $h_i$ , a aproximação a calcular seria expressa por  $\mathbf{x}_n = \mathbf{x}(t_i) = \mathbf{x}(t_{i-1} + h_i)$ , com  $n \in [0, N]$ . Similarmente:

$$\mathbf{f}_n = \mathbf{f}(t_i, \mathbf{x}(t_i), \mathbf{p}(t_i)) \quad (2.7a)$$

e

$$\mathbf{g}_n = \mathbf{g}(\mathbf{x}(t_i), \mathbf{p}(t_i)) \quad (2.7b)$$

Reforça-se aqui que, por considerar-se a resolução de sistemas DAE de índice unitário, para todas as metodologias apresentadas nesta seção, a solução obtida  $\mathbf{x}_{n+1}$  do sistema ODE (2.4a) satisfaz:

$$0 = \mathbf{g}_{n+1} = \mathbf{g}(\mathbf{x}_{n+1}, \mathbf{p}_{n+1}) \quad (2.8)$$

ou seja, as variáveis algébricas são calculadas pela substituição das aproximações obtidas através dos métodos a cada passo no sistema de equações algébricas.

O método mais simples conhecido é o método de Euler explícito, onde é utilizada a aproximação *forward* para a derivada:

$$\dot{\mathbf{x}} = \frac{\mathbf{x}_{n+1} - \mathbf{x}_n}{h} \quad (2.9)$$

Neste caso, as aproximações são dadas por uma predição linear, isto é:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}_n \quad (2.10)$$

Os métodos ponderados consistem em uma família de métodos dados por uma aproximação linear ponderada, isto é, a direção da solução a computar é uma combinação convexa da inclinação

(derivada) da solução passada e a inclinação atual:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h(\theta \mathbf{f}_n + (1 - \theta) \mathbf{f}_{n+1}) \quad (2.11)$$

Note que para  $\theta = 1$ , obtém-se o método de Euler explícito. Com  $\theta = 1/2$  e  $\theta = 0$ , temos os conhecidos métodos do trapézio e o método de Euler implícito (*backward*), respectivamente. Para os casos em que  $\theta \neq 1$ , necessita-se a cada passo calcular  $\mathbf{f}_{n+1}$ , e portanto o ponto  $\mathbf{x}_{n+1}$  precisa ser estimado ao menos na primeira chamada do método, utilizando algum método numérico explícito. Para casos em que  $h$  é suficientemente pequeno, é possível usar um método simples de baixa ordem, como o método de Euler, para fazer esta estimativa rapidamente [17].

As vantagens desses métodos consistem justamente no seu baixo custo computacional, uma vez que, a menos no primeiro passo, é necessário computar as equações de  $\mathbf{f}$  uma única vez, admitindo-se  $\mathbf{f}_0$  conhecida. Entretanto, é de se esperar que uma aproximação linear seja pouco precisa, ou tome muitas iterações para computar uma solução  $\mathbf{x}_{n+1}$  dentro da tolerância.

Para evitar computar-se aproximações com derivadas de mais alta ordem, os métodos de Runge-Kutta computam a função da dinâmica  $\mathbf{f}$  em uma série de pontos intermediários. A ideia é utilizar combinações lineares de  $\mathbf{f}$  em diferentes instantes para aproximar a solução de (2.4a) em uma determinada ordem, a qual quanto maior, mais combinações serão empregadas, e maior a precisão da aproximação. Por outro lado, quanto maior a ordem, mais tempo é consumido para computar cada solução.

A fórmula geral de um Runge-Kutta de ordem  $s$  é dada pelo conjunto de equações (2.12) a seguir:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h \sum_{j=1}^s b_j \mathbf{k}_j \quad (2.12a)$$

onde

$$\mathbf{k}_1 = \mathbf{f}(n, \mathbf{x}_n, \mathbf{p}_n), \quad (2.12b)$$

⋮

$$\mathbf{k}_s = \mathbf{f}\left(n + hc_j, \mathbf{x}_n + h \sum_{j=1}^{i-1} a_{sj} \mathbf{k}_j, \mathbf{p}_n\right) \quad (2.12c)$$

$$c_i = \sum_{j=1}^{i-1} a_{sj}, \quad i = 2, \dots, s. \quad (2.12d)$$

$$\sum_{j=1}^s b_j = 1. \quad (2.12e)$$

Os coeficientes  $\{b_j, c_i, a_{sj}\}$  definem o método e devem satisfazer as restrições impostas por (2.12d) e (2.12e), que são representadas por uma matriz<sup>2</sup> denominada Tabela de *Butcher*, como mostra a Figura 2.1, e são provenientes da comparação da equação (2.12a) com sua expansão em série de Taylor de ordem  $s$ . Maiores detalhes podem ser obtidos em [13].

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} = \begin{array}{c|c} \mathbf{c} & \mathbf{A} \\ \hline & \mathbf{b}^T \end{array}$$

Figura 2.1: Tabela de Butcher.

Como exemplo, considere um Runge-Kutta de segunda ordem ( $s = 2$ ):

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h(b_1 \mathbf{k}_1 + b_2 \mathbf{k}_2) \quad (2.13a)$$

$$\mathbf{k}_1 = \mathbf{f}(n, \mathbf{x}_n, \mathbf{p}_n) \quad (2.13b)$$

$$\mathbf{k}_2 = \mathbf{f}(n + hc_2, \mathbf{x}_n + ha_{21} \mathbf{k}_1, \mathbf{p}_n) \quad (2.13c)$$

<sup>2</sup>Desta matriz vem a origem dos índices dos  $i, j$  e  $s_j$  que correspondentes às posições dos coeficientes na tabela.

De (2.12d) e (2.12e) temos que  $b_1 + b_2 = 1$  e  $c_2 = a_{21}$ . Note que podem ser obtidas infinitas possibilidades de escolha. Uma opção seria  $b_1 = b_2 = 1/2$  e  $c_2 = a_{21} = 1$ , da qual obtém-se um método de Runge-Kutta muito conhecido, o método do Trapézio:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + \frac{h}{2}(\mathbf{f}_n + \mathbf{f}_{n+1}) \quad (2.14)$$

Métodos de Runge-Kutta estão entre os mais populares, em especial por sua alta precisão e estabilidade numérica [21]. Para ordens acima da quarta são, em geral, muito precisos mas a custo computacional muito elevado.

### 2.2.2 Métodos de colocação<sup>3</sup>

Os métodos de colocação são também métodos de passo único e compartilham algumas similaridades com o método de Runge-Kutta. Esses métodos aproximam a solução da ODE (2.4a) por um determinado polinômio  $\tilde{\mathbf{x}}$  de ordem  $s$  em uma série de  $s + 1$  pontos de seu domínio, denominados pontos de colocação, tal que sua derivada nesses pontos seja a mesma, ou tenha a mesma inclinação, que a da equação diferencial a resolver, ou seja,  $\dot{\tilde{\mathbf{x}}} = \mathbf{f}$ .

Para tal interpolação podem ser usados diferentes tipos de curvas, como séries de potências, splines<sup>4</sup> e mesmo simples interpolações lineares. Entretanto, os polinômios de Lagrange são os mais populares devido à sua excelente precisão e facilidade de calcular [22].

Define-se a variável  $\tau$  como a variável normalizada do tempo em um determinado subintervalo, tal que no  $i$ -ésimo subintervalo, tem-se:

$$t_i = t_{i-1} + h\tau \quad \tau \in [0, 1] \quad (2.15)$$

com  $t_i \in T = [t_0, t_f]$ . A base para o polinômio de Lagrange é dada por:

$$\ell_j(\tau) = \prod_{k=0, \neq j}^s \frac{\tau - \tau_k}{\tau_j - \tau_k} \quad (2.16)$$

onde  $\tau_j$  e  $\tau_k$  são os pontos de colocação, com a seguinte propriedade:  $\tau_0 = 0$  e  $\tau_j < \tau_{j+1}$  para  $j = 0, \dots, s - 1$ . Verifica-se ainda

<sup>3</sup>Esta seção foi baseada em [4]

<sup>4</sup>Curvas polinomiais por partes [23].



que quando  $\tau = \tau_j$ , então  $\ell_j(\tau_j) = 1$ , mas  $\ell_k(\tau_j) = 0$  para  $k = 0, \dots, s - 1$ , se  $k \neq j$ .

A Figura 2.2 ilustra estes conceitos que, em suma, explicam que se o polinômio  $\ell_j$  for multiplicado por um valor qualquer  $\tilde{x}_{ij}$ , o resultado assumirá o valor de  $\tilde{x}_{ij}$  quando  $\tau = \tau_j$ , ou seja, nos pontos de colocação, e em todos os outros  $k$  pontos, os valores  $\ell_k$  do polinômio serão nulos. Esta ideia pode ser usada para descrever

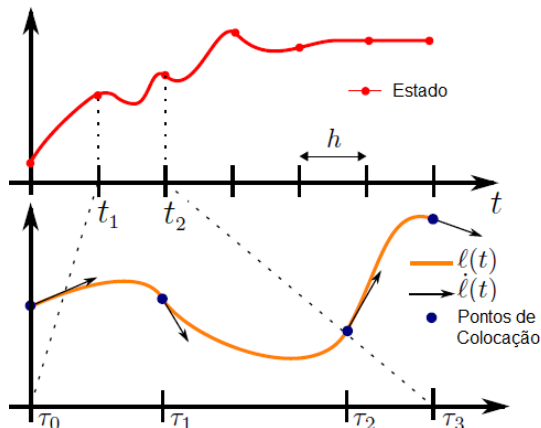


Figura 2.2: O Método de colocação. (Fonte: [4]).

a curva que passa pelos pontos  $(\tau_j, \tilde{x}_{ij})$  para  $j = 1, \dots, s$  através da soma de todos os termos  $\ell_j(\tau)\tilde{x}_{ij}$ . Generalizando para a forma vetorial:

$$\tilde{\mathbf{x}}(t) = \sum_{j=0}^s \ell_j(\tau)\tilde{\mathbf{x}}_{ij} \quad (2.17)$$

onde  $\tilde{\mathbf{x}}(t)$  representa a evolução da curva-solução do estado no intervalo  $T$ , e  $\tilde{\mathbf{x}}_{ij}$  é o valor correspondente dos estados no ponto de colocação  $j$ , no  $i$ -ésimo subintervalo  $t_i$ .

Os pontos de colocação  $\tau_j$  podem ser obtidos utilizando-se quadratura Gaussiana. Existem uma série de metodologias para a obtenção destes pontos a partir desta técnica. A mais comum é a denominada raízes Legendre-Gauss-Radau (LGR). Basicamente define-se uma ordem  $s$  e um intervalo tal como  $T$ , e o procedimento retorna um número  $s + 1$  de pontos pertencentes ao domínio de um polinômio-solução, que resolve uma equação diferencial de-

nominalizada Equação de Legendre. Mais detalhes sobre esta e outras técnicas encontram-se em [24].

Além de passar pelos pontos  $\tau_k$  de colocação desejados, deve-se também garantir que o polinômio aproximado dado pela equação (2.17), e o sistema ODE a resolver (2.4a) tenham a mesma inclinação nestes pontos, ou seja:

$$\frac{d\tilde{\mathbf{x}}}{dt}(\tau_k) = \mathbf{f}(\tau_k, \tilde{\mathbf{x}}(\tau_k), \mathbf{p}(\tau_k)), \quad k = 1, \dots, s. \quad (2.18)$$

Diferenciando-se as equações (2.15) e (2.17) em relação a  $\tau$  e substituindo-as na igualdade acima, chega-se a:

$$\sum_{j=0}^s \tilde{\mathbf{x}}_{ij} \frac{d\ell_j}{d\tau}(\tau_k) = h\mathbf{f}(\tau_k, \tilde{\mathbf{x}}(\tau_k), \mathbf{p}(\tau_k)), \quad k = 1, \dots, s. \quad (2.19a)$$

com

$$\mathbf{g}(\tilde{\mathbf{x}}(\tau_k), \mathbf{p}) = 0 \quad (2.19b)$$

onde, para o primeiro subintervalo  $i = 1$ , a condição inicial é dada por  $\tilde{\mathbf{x}}_{10} = \mathbf{x}_0$ .

A resolução deste sistema de equações não lineares fornece a aproximação para os estados nos pontos de colocação. As variáveis algébricas  $\mathbf{p}(\tau_k)$  são obtidas pela substituição da solução  $\tilde{\mathbf{x}}_{ij}$  de (2.19a) na restrição algébrica (2.19b). Observa-se a necessidade de computar as derivadas do polinômio de Lagrange ( $d\ell_j/d\tau$ ) que, para elevadas ordens, pode tornar-se uma operação muito custosa, prejudicando sua eficiência.

Ao final do último intervalo  $i$  de integração ( $t_i = t_f$ ), o método entrega a solução desejada, correspondente ao último ponto de colocação deste intervalo, isto é,  $\tilde{\mathbf{x}}_{is}$ , sendo  $s$  a ordem do método, que usa um número  $(s + 1)$  de pontos de colocação para computar a solução.

Este trabalho tem o objetivo de passar uma visão geral do funcionamento dos métodos, já que a escolha de um deles não é sempre trivial e depende do problema a resolver. Em termos práticos, para um *solver* comercial é necessário apenas fornecer uma ordem para o polinômio ( $s$ ), a condição inicial ( $\mathbf{x}_0$ ) e o intervalo de integração ( $T$ ). Aumentando-se a ordem do método, obtém-se uma aproximação mais precisa, ao custo de aumentar-se o tempo computacional despendido para a tarefa. Desenvolver cada método analiticamente, mesmo para um exemplo simples, é trabalhoso

e não é escopo desta tese. Se o leitor deseja aprofundar-se no desenvolvimento analítico de cada técnica, recomenda-se a leitura de [13, 16, 20].

### 2.2.3 Métodos de múltiplos passos e de diferenças finitas

As metodologias numéricas vistas até então utilizavam apenas informação da solução calculada no passo anterior, que era usada como ponto de partida para calcular a solução corrente, e por isso, tornaram-se conhecidos como uma classe de métodos, denominada métodos de passo único. Existe, porém, outra classe de métodos que não apenas se utilizam da solução anterior, mas de soluções computadas em uma série de passos anteriores, utilizando-se assim de um histórico de múltiplas soluções previamente computadas. Por esta razão, são denominados métodos de múltiplos passos.

Nestas técnicas, para uma dada ordem  $s > 0$ , deve-se conhecer um total de  $(s - 1)$  pontos que serão usados para construir uma aproximação para a solução do problema (2.4a). A formulação geral desses métodos é dada por:

$$\sum_{j=0}^s \alpha_j \mathbf{x}_{n-j+1} = h \sum_{j=0}^s \beta_j \mathbf{f}_{n-j+1} \quad (2.20)$$

onde  $\alpha_j$  e  $\beta_j$  são obtidos de uma interpolação polinomial de ordem  $s$  para a integral da ODE (2.4a) em um intervalo equidistante de pontos. Como existem inúmeras possibilidades para a escolha do polinômio, muitos pesquisadores deram nome às expressões resultantes das aproximações polinomiais, tal como o método de Milnes (ordem  $s = 2$ ) abaixo:

$$\mathbf{x}_{n+1} = \mathbf{x}_{n-1} + \frac{h}{3} (\mathbf{f}_{n-1} + 4\mathbf{f}_n + \mathbf{f}_{n+1}) \quad (2.21)$$

onde  $\alpha_0 = \alpha_2 = 1$ ,  $\beta_0 = \beta_2 = 1/3$ ,  $\alpha_1 = 0$  e  $\beta_1 = 4/3$ .

Observa-se que nesta técnica de integração se faz necessário utilizar um método de partida para calcular as  $(s - 1)$  soluções anteriores. Por exemplo, no caso do método de Milnes (2.21), no instante  $n = 1$  e fornecido valor inicial  $\mathbf{x}_0$ , necessita-se calcular  $\mathbf{x}_1$ . Para tal tarefa, deve-se utilizar um método de passo único, tal como um método de Euler ou Runge-Kutta.

Esta característica faz com que os métodos de múltiplos passos sejam frequentemente usados na estrutura *preditor-corretor*, onde é fornecida uma estimativa inicial da solução  $\mathbf{x}_{n+1}$ , chamada de

predição, por um método de passo único, que é corrigida por um método de múltiplos passos (de igual ordem) até que a diferença entre preditor e corretor satisfaça uma determinada condição de precisão/tolerância para a aproximação.

Por exemplo, utiliza-se o método do trapézio como preditor  $(\mathbf{x}_{n+1})_p$ , e o método de Milnes como corretor  $(\mathbf{x}_{n+1})_c$ , repetindo-se o cálculo até que a diferença satisfaça a condição  $\mathcal{E}$  de parada, retornando-se como solução a aproximação corrigida  $(\mathbf{x}_{n+1})_c$ :

$$(\mathbf{x}_{n+1})_p = \mathbf{x}_n + \frac{1}{2}h(\mathbf{f}_n + \mathbf{f}_{n+1}) \quad (2.22a)$$

$$(\mathbf{x}_{n+1})_c = \mathbf{x}_{n-1} + \frac{h}{3}(\mathbf{f}_{n-1} + 4\mathbf{f}_n + \mathbf{f}_{n+1}) \quad (2.22b)$$

$$\|(\mathbf{x}_{n+1})_p - (\mathbf{x}_{n+1})_c\|_\infty < \mathcal{E} \quad (2.22c)$$

onde a norma infinita acima toma o máximo erro absoluto computado entre as duas aproximações calculadas.

Uma vez que os valores dos cálculos das funções-vetores  $\mathbf{f}_n$  podem ser armazenados a cada cômputo, estes métodos requerem muito menos processamento com relação aos métodos de Runge-Kutta de mesma ordem. Entretanto, em caso de mudança no tamanho passo de integração  $h$ , os valores de  $\mathbf{f}$  precisam ser recomputados para este novo valor de passo.

Um subgrupo desta classe de métodos que calculam a função  $\mathbf{f}$  somente no ponto a ser aproximado são os chamados métodos de diferenças finitas ou **Backward Differences Formulas (BDF)-methods** [20]. Um número variável de soluções passadas pode ser usado, mas obrigatoriamente  $\beta_0 \neq 0$  e  $\beta_{1\dots s} = 0$ . Ou seja:

$$\sum_{j=0}^s \alpha_j \mathbf{x}_{n-j+1} = h\beta_0 \mathbf{f}_{n+1} \quad (2.23)$$

O método **BDF** é, portanto, puramente implícito, fato que agrega excelentes propriedades de estabilidade numérica, como será apresentado na Subseção 2.3.2.

A aplicação de Métodos **BDFs** a problemas **DAEs** foi muito estudada no final da década de 80, e até hoje são um dos métodos mais poderosos para resolução de **PVIs** envolvendo **DAEs** de índice até 2.

## 2.3 PROPRIEDADES NUMÉRICAS

### 2.3.1 Erro local de aproximação

Como discutido na seção anterior, além de um método numérico para computar as aproximações, é necessário que se tome uma estimativa do erro local da aproximação que possa ser usada como medida da precisão da solução. Apresentam-se aqui duas formas de se calcular o erro local nas quais pelo menos duas estimativas da solução devem ser fornecidas.

A primeira forma envolve o tamanho do passo  $h$ . Utiliza-se o mesmo método numérico para calcular o valor de uma aproximação utilizando-se tamanhos de passo diferentes. Por exemplo:

$$e_{n+1}^{(h)} = \left\| \mathbf{x}_{n+1}^{(h_1)} - \mathbf{x}_{n+1}^{(h_2)} \right\|_{\infty} \quad (2.24)$$

que é conhecida como fórmula da extrapolação de Richardson e nada mais indica do que a distância entre as soluções computadas com cada tamanho de passo, indicados entre parênteses na fórmula. Este tipo de estimativa é muito usada para métodos mais simples, tal como o método de Euler, com tamanho de passo suficientemente pequeno, tal que  $h_1 \approx h_2$ , e em casos onde a tolerância de aproximação requerida  $\mathcal{E}$  não seja muito pequena [25].

Para que o tamanho do passo não seja fator limitante para computar o erro local, outro critério que pode ser utilizado é a ordem do método. Calcula-se a mesma aproximação  $\mathbf{x}_{n+1}$  utilizando o mesmo método com ordens de aproximação distintas, para então tomar a diferença entre elas. Por exemplo, utilizar-se de métodos de Runge-Kutta de segunda (RK2) e terceira (RK3) ordem para calcular a aproximação no mesmo ponto:

$$e_{n+1} = \left\| (\mathbf{x}_{n+1})_{RK2} - (\mathbf{x}_{n+1})_{RK3} \right\|_{\infty} \quad (2.25)$$

*Solvers* comerciais geralmente fazem uso de um ou mais métodos com ordem variável para computar as soluções, aproveitando-se dessas para estimar o erro local. Muitos deles indicam as ordens utilizadas no próprio nome, tal como o *ode23* (Runge-Kutta de segunda e terceira ordem) e o *ode45* (Runge-Kutta de quarta e quinta ordem) do pacote *ode* do MATLAB 2015a<sup>®</sup>.

Os métodos de múltiplos passos também adotam o critério de ordem, já que calcular soluções para diferentes tamanhos de passos pode ser muito custosa, especialmente para altas ordens. Em geral, fazem uso da estrutura preditor-corretor, como discutido na

Subseção 2.2.3. Como exemplo, considere o BDF de primeira ordem mais simples – o método de Euler implícito – como preditor,  $(\mathbf{x}_{n+1})_p$ , e o método de Milnes, descrito por (2.19), como corretor,  $(\mathbf{x}_{n+1})_c$ . Dessa forma, estima-se o erro local por:

$$(\mathbf{x}_{n+1})_p = \mathbf{x}_n + h\mathbf{f}_{n+1} \quad (2.26a)$$

$$(\mathbf{x}_{n+1})_c = \mathbf{x}_{n-1} + \frac{h}{3}(\mathbf{f}_{n-1} + 4\mathbf{f}_n + \mathbf{f}_{n+1}) \quad (2.26b)$$

$$e_{n+1} = \|(\mathbf{x}_{n+1})_c - (\mathbf{x}_{n+1})_p\|_\infty \quad (2.26c)$$

Como percebido, o erro local depende do tamanho do passo de integração e do método numérico utilizado. De fato, é provado por [13] que, para um método de ordem  $s$ , o erro local relaciona-se assintoticamente com o tamanho do passo  $h$ , através da relação  $e = \kappa h^{s+1}$ , onde  $\kappa$  é uma constante que depende do método e do PVI [18]. Esta relação será utilizada em uma das metodologias desenvolvidas nesta tese.

Por fim, apesar das técnicas aqui discutidas expressarem o erro local de maneira absoluta, é muito comum representá-lo também por em sua forma relativa. Tomando a equação (2.26c) em sua forma relativa, obtém-se:

$$e_{n+1} = \left\| \frac{(\mathbf{x}_{n+1})_c - (\mathbf{x}_{n+1})_p}{(\mathbf{x}_{n+1})_c} \right\|_\infty \quad (2.27)$$

que será a estimativa adotada neste trabalho.

### 2.3.2 Estabilidade

Para que um método numérico tenha um bom comportamento é necessário, além de que este forneça uma solução em um número finito de passos, que se garanta sua estabilidade local, no sentido em que, para dado um problema de valor inicial, e um tamanho de passo  $h$  fixo, as soluções aproximadas pelo método fiquem limitadas à medida que se avança no tempo, mesmo que o sistema a integrar seja instável.

Matematicamente, Silva [26] define estabilidade das seguinte maneira: dado um método numérico  $M$  que forneça uma solução  $\mathbf{x}_{n+1}^{(1)}$  para um PVI com condição inicial  $\mathbf{x}_0^{(1)}$ , e outra solução  $\mathbf{x}_{n+1}^{(2)}$  para o mesmo problema, mas com valor inicial  $\mathbf{x}_0^{(2)}$ , então o método

$M$  é estável se, existirem constantes  $0 < C_n < 1$ , para todo instante  $n \in [0, N]$ , tal que:

$$\left\| \mathbf{x}_{n+1}^{(1)} - \mathbf{x}_{n+1}^{(2)} \right\|_{\infty} \leq C_n \left\| \mathbf{x}_0^{(1)} - \mathbf{x}_0^{(2)} \right\|_{\infty} \quad (2.28)$$

Numericamente, em se tratando da implementação de *solvers* para a resolução de **PVIs**, e conhecendo-se uma única condição inicial consistente, deve-se garantir de alguma forma a estabilidade numérica durante a própria determinação da solução. Para tanto, o problema da estabilidade local dos métodos numéricos será discutido nesta seção.

### 2.3.2.1 Estabilidade dos métodos de passo único

Primeiramente aborda-se o comportamento numérico dos métodos de passo único. Considere que o sistema dinâmico em malha fechada (2.4a) a integrar seja representado localmente pela equação (2.29), tal que  $\mathbf{x}$  possua  $m$  estados, e a matriz jacobiana  $A = \frac{d\mathbf{f}}{d\mathbf{x}}(\bar{\mathbf{x}})$ , de tamanho  $m \times m$ , seja constante e represente a função-vetor  $\mathbf{f}$  da dinâmica do sistema em uma vizinhança do ponto de um operação  $\bar{\mathbf{x}}$ , para dadas condições iniciais (2.4c).

$$\dot{\mathbf{x}}(t) = \mathbf{f}(t, \mathbf{x}(t), \mathbf{p}(t)) \cong A\mathbf{x}(t) \quad (2.29)$$

Considere os  $m$  autovalores ( $\lambda \in \mathbb{C}$ ) da matriz  $A$  representados pela matriz diagonal  $\Lambda$  abaixo.

$$\Lambda = \begin{bmatrix} \lambda_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \lambda_m \end{bmatrix} \quad (2.30)$$

A questão que procura-se responder é sob quais condições, em termos de parâmetros de formulação, um método numérico de passo único consegue representar um comportamento estável e limitado das aproximações que este fornece como solução de um sistema **ODE** de primeira ordem.

Desta forma, considera-se a matriz  $\Lambda$ , de tal maneira que em um instante  $n$  de um método numérico discreto, seja possível representar a função dinâmica do problema (2.29) por, [20].:

$$\mathbf{f}_n = \Lambda \mathbf{x}_n \quad (2.31)$$

Tomando como exemplo, o método de Euler explícito de (2.10), e aplicando (2.31), obtém-se:

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\Lambda\mathbf{x}_n \quad (2.32a)$$

$$\mathbf{x}_{n+1} = (I + \bar{h})\mathbf{x}_n, \quad \bar{h} = h\Lambda. \quad (2.32b)$$

Para que um método numérico de passo único seja estável deve-se garantir que as raízes de sua equação característica encontrem-se dentro da hiper-esfera de raio unitário [14]. Tomando-se o caso escalar, para o autovalor  $\lambda_j \in [\lambda_1, \dots, \lambda_m]$ , deve-se satisfazer:

$$|1 + \bar{h}_j| < 1, \quad \bar{h}_j = \lambda_j h \quad (2.33)$$

ou seja, seu domínio de estabilidade pertence ao raio  $\bar{h}_j \in (-2, 0)$ ,  $\forall j$ .

Considere agora, a análise do método de Euler implícito, que pode ser obtido do método ponderado (2.11), fazendo  $\theta = 1$ :

$$\mathbf{x}_{n+1} = \mathbf{x}_n + h\mathbf{f}_{n+1} \quad (2.34)$$

Aplicando a aproximação (2.31):

$$\mathbf{x}_{n+1} = (I - \bar{h})^{-1}\mathbf{x}_n, \quad \bar{h} = h\Lambda. \quad (2.35)$$

e, portanto, para todo autovalor  $\lambda_j \in \mathbb{C}$ , chega-se à seguinte condição de estabilidade:

$$|1 - \bar{h}_j| > 1, \quad \bar{h}_j = \lambda_j h, \forall j \in [1, m]. \quad (2.36)$$

que é o exterior de um disco com um raio de tamanho unitário e centrado em  $\bar{h}_j = 1$ . Note que este domínio compreende todo o semiplano complexo negativo. Portanto, o método de Euler implícito imita a estabilidade assintótica de uma ODE linear ( $\lambda < 0$ ) sem restrição no tamanho do passo  $h$ , já que este é positivo. Esta propriedade pode ser verificada para todos os métodos implícitos, como os BDFs, tornando-os assim ferramentas poderosas para uso em algoritmos de integração numérica, em especial, os que utilizam tamanho do passo variável. Por fim, para o caso de PVI instáveis ( $\lambda > 0$ ), necessita-se adotar um tamanho de passo maior, que satisfaça  $\bar{h}_j > 2, \forall j$ .

Observa-se que esta análise de estabilidade considera sistemas que operam próximo a um ponto de operação definido, caso típico, por exemplo, de processos dinâmicos controlados localmente. A estabilidade local aqui discutida não é válida, portanto, sistemas oscilatórios periódicos, ou sistemas não lineares que operem longe de um ponto de equilíbrio local.



## 2.3.2.2 Estabilidade dos métodos de múltiplos passos

Para fazer uma análise de estabilidade em métodos de múltiplos passos, aplica-se novamente (2.31), mas agora à equação geral destes métodos, dada por (2.20), obtendo-se:

$$\sum_{j=0}^s (\alpha_j I - h\beta_j \Lambda) \mathbf{x}_{n-j+1} = 0 \quad (2.37)$$

Considere  $\lambda_i$ , o  $i$ -ésimo autovalor de  $\Lambda$ , com  $i = 1, \dots, m$ . Dado que a solução  $\mathbf{x}_{n+1}$  da equação à diferenças (2.35) pode ser expressa como uma combinação linear de potências das raízes de sua equação característica associada [5], define-se:

$$\pi(w; \bar{h}) = \rho(w) - \bar{h}_i \sigma(w), \quad \bar{h}_i = \lambda_i h. \quad (2.38a)$$

$$\rho(w) = \sum_{j=0}^s \alpha_j w^{s-j} \quad (2.38b)$$

$$\sigma(w) = \sum_{j=0}^s \beta_j w^{s-j} \quad (2.38c)$$

onde  $\pi(w; \bar{h}_i)$  é denominado polinômio de estabilidade de métodos de múltiplos passos de ordem  $s$  [5].

Um método de múltiplos passos é dito ser estável, se e somente se todas as  $s$  raízes de  $\pi(w; \bar{h}_i)$ , e para todo estado  $x_i, i = 1, \dots, m$ , possuírem módulo menor que 1, isto é [5]:

$$|w_j| = |w_j(\bar{h}_i)| < 1, \quad j = 1, \dots, s. \quad (2.39)$$

Nota-se que a obtenção das raízes de (2.39) pode ser muito trabalhosa para altas ordens do método ( $s$  grande). Considera-se, então, um método alternativo para determinar o intervalo de estabilidade dos métodos de múltiplos passos, através da aplicação dos seguintes passos, do conhecido critério de estabilidade de Schur [5], para o polinômio  $\pi(w; \bar{h}_i)$ :

- (i)  $\pi(w; \bar{h}_i)$  é um polinômio de Schur se todas as suas raízes tiverem módulo menor que a unidade, isto é, satisfizerem a condição (2.39);
- (ii) Se  $\pi$  é um polinômio tal que  $\pi = a_k w^k + \dots + a_1 w + a_0$ , com  $a_k \neq 0$  e  $a_0 \neq 0$ , então  $\hat{\pi} = a_0 w^k + \dots + a_{k-1} w + a_k$  é denotado conjugado de  $\pi$ ;

(iii) Seja  $\pi_1(w, \bar{h}_i) = (1/w)[\hat{\pi}(0, \bar{h}_i)\pi(w, \bar{h}_i) - \pi(0, \bar{h}_i)\hat{\pi}(w, \bar{h}_i)]$ . O polinômio  $\pi(w; \bar{h}_i)$  é um polinômio de Schur se e somente se  $|\hat{\pi}(0; \bar{h}_i)| > |\pi(0; \bar{h}_i)|$ , e  $\pi_1(w, \bar{h}_i)$  for um polinômio de Schur.

Vejamos a aplicação destes passos para definir o intervalo de estabilidade do seguinte exemplo:

$$\mathbf{x}_{n+1} = \mathbf{x}_{n-1} + \frac{h}{2}(\mathbf{f}_n + 3\mathbf{f}_{n-1}) \quad (2.40)$$

onde temos  $s = 2$  com  $\alpha_0 = 1$ ,  $\alpha_1 = 0$ ,  $\alpha_2 = -1$ , e  $\beta_0 = 0$ ,  $\beta_1 = 1/2$  e  $\beta_2 = 3/2$ .

Primeiramente, calcula-se o polinômio de estabilidade e seu conjugado:

$$\rho(w) = w^2 - 1 \quad (2.41a)$$

$$\sigma(w) = \frac{1}{2}(w + 3) \quad (2.41b)$$

e portanto

$$\pi(w; \bar{h}_i) = w^2 - \frac{1}{2}\bar{h}_i w - (1 + \frac{3}{2}\bar{h}_i) \quad (2.42a)$$

$$\hat{\pi}(w; \bar{h}_i) = -(1 + \frac{3}{2}\bar{h}_i)w^2 - \frac{1}{2}\bar{h}_i w + 1 \quad (2.42b)$$

Claramente,  $|\hat{\pi}(0; \bar{h}_i)| > |\pi(0; \bar{h}_i)|$ , se e somente se  $\bar{h}_i \in (-\frac{4}{3}, 0)$ . Como

$$\pi_1(w, \bar{h}_i) = -\frac{1}{2}\bar{h}_i(2 + \frac{3}{2}\bar{h}_i)(3w + 1) \quad (2.43)$$

tem raiz única igual  $w = -\frac{1}{3}$ , sendo portanto um polinômio de Schur, deduz-se que  $\pi(w; \bar{h}_i)$  também o é, se e somente se  $\bar{h}_i \in (-\frac{4}{3}, 0)$ .

Portanto, o intervalo de estabilidade absoluta do método de múltiplos passos (2.40) é  $\bar{h}_i \in (-\frac{4}{3}, 0)$ , dado  $\bar{h}_i = h\lambda_i$ , com  $i = 1, \dots, m$ .

Para um completo detalhamento da definição matemática do critério de Schur, recomenda-se a leitura de [27].

### 2.3.3 Controle do tamanho do passo de integração

A fim de melhorar a eficiência computacional ou mesmo garantir a estabilidade dos métodos de passo único, é muito comum *solvers* comerciais utilizarem-se de um tamanho de passo adaptativo. Motivada pela condição de parada relacionando o erro local

com a tolerância especificada, uma maneira de computar qual o próximo passo  $h_{n+1}$  a ser tomado pelo método é justamente permitir uma mudança proporcional à razão entre a tolerância  $\mathcal{E}$  e erro local  $e_{n+1}$ , levando em conta a ordem (precisão) do método. Assim:

$$h_{n+1} = \gamma \left( \frac{\mathcal{E}}{e_{n+1}} \right)^{1/p} \cdot h_n \quad (2.44)$$

onde  $p$  corresponde à ordem do método, e  $\gamma$  é um “fator de segurança” devendo ser escolhido de tal modo que  $\gamma \leq 1$ . O propósito deste fator é reduzir o risco de rejeitar a solução computada no próximo passo do método, quando o erro local chega a um valor muito próximo da tolerância, evitando assim recalcular tamanhos de passo de magnitude semelhante. No entanto, se o erro local for maior ou muito maior que  $\mathcal{E}$ , este será rejeitado e um novo  $h$  de menor tamanho será calculado. Observe que métodos de maior ordem – e mais precisos – permitem o uso de tamanhos de passos maiores.

Esta estratégia “padrão” para o controle do tamanho do passo de integração normalmente funciona bem. Entretanto, em muitos problemas **DAEs**, esta técnica tem desempenho aquém do esperado apresentando muita oscilação na variável  $h$ , pois muito tempo computacional é perdido recalculando novos tamanhos de passo devido soluções rejeitadas, quando a razão (tolerância por erro local) da equação (2.44) chega próxima da unidade [18], mesmo utilizando-se o fator  $\gamma$ . Isto é especialmente verdade quando se utilizam métodos puramente explícitos na resolução de equações rígidas (stiff) [18], as quais serão estudadas ainda neste capítulo. Por isso, novas estratégias de controle de passo para evitar estes problemas serão propostas no **Capítulo 4** desta dissertação.

### 2.3.4 Equações rígidas

Quando o tamanho do passo de integração  $h$  é limitado pelo intervalo de estabilidade do método, tendo esta dificuldade em cumprir com os requisitos de precisão numérica impostos, a menos que  $h$  seja muito pequeno, o sistema de equações diferenciais e algébricas é dito ser rígido ou “stiff”. Em geral, esse termo é empregado para sistemas de equações nos quais existem estados com escalas de tempo bastante diferentes uma das outras, diga-se a abertura de uma válvula e a dinâmica de uma pressão, por exemplo, provocando dificuldades na resolução numérica do problema. Isso porque a resolução numérica de um sistema **DAE** depende não apenas do

método escolhido mas também do comportamento dinâmico dos estados do PVI a resolver.

Para melhor esclarecer esta propriedade, considere dois estados  $x_1$  e  $x_2$  soluções de uma ODE de primeira ordem, cujas dinâmicas podem ser observadas na Figura 2.3, Considere utilizar o método de Euler explícito (2.10) para computar estas curvas.

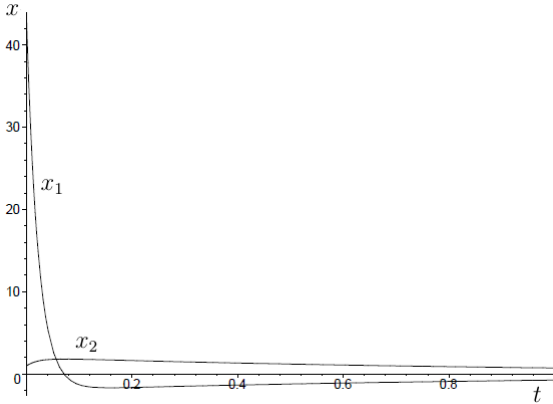


Figura 2.3: O fenômeno da rigidez. (Fonte: [5]).

Observa-se que  $x_1$  possui uma dinâmica muito rápida próximo a  $t = 0$ , e lenta a partir da vizinhança do instante  $t = 0,2$ , enquanto  $x_2$  varia lentamente em todo intervalo  $t = [0, 1]$ . Da condição de estabilidade do método, (2.33), constata-se que o tamanho passo é limitado no intervalo:

$$0 \leq h \leq 2/|\lambda_j|, \quad j = 1, 2. \quad (2.45)$$

onde  $\lambda_1$  e  $\lambda_2$  correspondem aos autovalores dos estados 1 e 2, respectivamente.

Sabe-se que a constante de tempo de  $x_1$  é mais rápida que  $x_2$ , logo  $|\lambda_1| > |\lambda_2|$ . Assim, o tamanho do passo  $h$  fica limitado pela imposição da condição de estabilidade do método, dada pela constante de tempo mais rápida do sistema  $|\lambda_1|$  e, mesmo que na maior parte do tempo (a partir de  $t > 0,2$ ) ambos os estados variem lentamente com dinâmica semelhante, o valor máximo que  $h$  pode assumir será definido por  $h = 2/|\lambda_1|$ . Por esta razão, métodos puramente explícitos quando usados para resolver esse tipo de sistema são muito lentos, embora ainda retornem uma solução após tomar uma quantidade de tempo considerável.

Considere agora o método de Euler implícito para a resolução do mesmo sistema. Dada que sua condição de estabilidade absoluta (2.36) implica em  $h > 0$ , condição que é sempre verdadeira, não há restrição alguma a ser imposta ao tamanho do passo de integração, a não ser requisitos de tempo máximo de amostragem ( $T_s$ ), por exemplo. Esta propriedade pode ser estendida para todos os métodos puramente implícitos, tornando-os muito mais eficientes na resolução numérica de sistemas do tipo rígido, dado que são incondicionalmente estáveis. Por esta razão, são amplamente utilizados em conjunto com a aplicação de metodologias para o controle do tamanho do passo de integração, como a apresentada na Seção 2.3.3, fato que pode ser constatado em diversos *solvers* comerciais (tal como o *ode15s*<sup>5</sup>) que se utilizam de métodos BDFs, puramente implícitos por definição.

Em suma, quanto mais explícito for o método, considerando metodologias híbridas (ou semi-implícitas), menor seu intervalo de estabilidade, e portanto o tamanho de passo utilizado, agregando mais tempo à resolução numérica do problema, sendo desta forma pouco competitivos quando aplicados a PVI que envolvam equações rígidas.

## 2.4 COMENTÁRIOS FINAIS

Neste capítulo apresentou-se uma fundamentação teórica com diversos conceitos a respeito de métodos numéricos para resolução de problemas de valor inicial envolvendo equações diferenciais e algébricas. Alguns dos métodos mais tradicionais na literaturas e presentes na maioria dos *solvers* comerciais foram expostos afim de proporcionar ao leitor uma noção geral de seu funcionamento, uma vez que não é objetivo desta tese demonstrar uma fundamentação matemática rigorosa de cada um deles. O capítulo seguinte discutirá a modelagem matemática do sistema DAE da estação de compressão e discutirá suas complexidades de resolução numérica.

---

<sup>5</sup>Do pacote MATLAB 2015a<sup>®</sup>.



### 3 MODELAGEM DE UMA ESTAÇÃO DE COMPRESSÃO CONTROLADA DE GÁS

Neste capítulo apresenta-se, de maneira geral, o funcionamento operacional da estação de compressão de gás de uma unidade de produção de petróleo, caso de estudo deste trabalho e, expõe-se uma análise quantitativa da estrutura numérica e complexidade do modelo matemático fenomenológico, isto é, do sistema de equações diferenciais e algébricas resultante, problema foco deste trabalho.

#### 3.1 DESCRIÇÃO DO SISTEMA

O processo em estudo contempla um sistema controlado de compressão de gás composta por dois compressores de três estágios cada, tal como mostra a Figura 3.1. Esta é uma típica estação de compressão de uma unidade de produção de gás, cujo modelo descrito neste capítulo é baseado nos trabalhos de [Plucenio et al.\[6\]](#) e [Vetorazzo \[2\]](#).

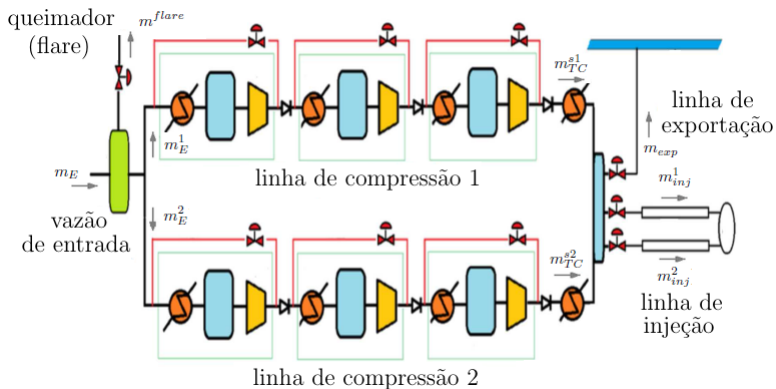


Figura 3.1: Estação de compressão de uma unidade de processamento de gás.

O objetivo da compressão de gás nas plataformas de produção de petróleo é aumentar a pressão do gás que vem do poços, após passar por processo de separação da água e do óleo, para que o gás comprimido possa ser transportado para venda (distribuição), ser usado como combustível ou fonte de energia na própria plataforma, ou ainda para ser injetado em linhas de (*gas-lift*) em processos de elevação artificial de fluidos.

A linha de entrada da estação recebe a vazão de gás proveniente do processo de separação de petróleo. Um vaso de separação à pressão controlada funciona como pulmão para armazenar certa quantidade de gás que o atinge de maneira ininterrupta e irregular. Esta vazão de entrada é vista como uma perturbação pelo sistema de compressão.

Caso uma quantia de gás não puder ser comprimida devido a algum tipo de falha no sistema, este montante deve ser conduzido ao queimador (denominado *flare*) a fim de evitar sobrepressão nas linhas de compressão da estação. A vazão de gás que chega às linhas de compressão é governada pela dinâmica de pressão resultante do balanço de massa no vaso de pressão de entrada.

Cada estágio de compressão é formado por um trocador de calor, que recebe o gás proveniente da linha de entrada ou de seu predecessor, responsável por controlar e reduzir a temperatura do gás para um valor fixo; por um vaso de pressão (ou separação) que acumula parte do gás para remoção, por decantação, de possíveis resíduos de fluido condensado liberado pelo próprio gás; por um compressor centrífugo, que operando a determinada velocidade de rotação, transfere energia ao gás aumentando assim sua pressão e temperatura, e por uma válvula de reciclo, cuja abertura deve garantir uma vazão mínima de gás passando pelo compressor, evitando assim o fenômeno de *surge* e uma operação instável do sistema.

A Figura 3.2 apresenta em detalhe um estágio de compressão e seus componentes.

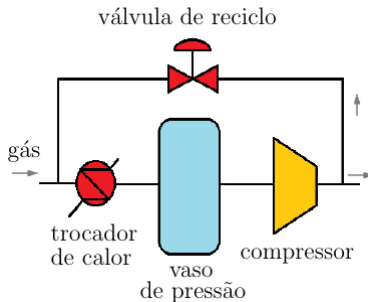


Figura 3.2: Estágio de compressão e seus componentes.

Na saída do terceiro estágio de cada linha encontram-se ainda trocadores de calor atuando para manter a temperatura do gás em um valor admissível para a descarga ou exportação. Imediatamente



após, mais um vaso de pressão de saída (ou *distribution header*) acumula o gás comprimido, e é responsável pela distribuição deste gás a três linhas de transporte: uma para exportação que destina o gás ou para suprimento de energia interno, ou para fora da unidade para fins de comercialização, e duas linhas para injeção de gás nos poços em processos elevação artificial de fluidos.

O controle regulatório deste sistema é parte intrínseca do modelo de equações diferenciais e algébricas desenvolvido, e corresponde ao sistema de controle real existente na estação. Este sistema de controle é composto por um conjunto de controladores PID dispostos a manter em níveis seguros, por exemplo, as pressões na entrada de cada linha de compressão, as temperaturas do gás na entrada de cada estágio, e regular a quantidade de gás a ser transportada para outras linhas.

Além disso, o sistema de controle atua sobre as válvulas de reciclo, quando a vazão de gás que circula pelos compressores compreende-se abaixo de determinado limite de operação, de forma que parte do gás comprimido seja recirculado pelos compressores, a fim de incrementar esta vazão de gás, evitando assim uma operação instável, denominada *surge*.

Em resumo, o controle regulatório modelado dispõe-se a realizar o:

- Controle da pressão de sucção na entrada de cada linha de compressão, atuando na velocidade de rotação do eixo de compressão;
- Controle da pressão de sucção do *header* de distribuição, atuando na válvula da linha de exportação de gás;
- Controle da temperatura do gás em cada estágio atuando na abertura da válvula de admissão de fluido refrigerante nos trocadores de calor;
- Controle da vazão de gás enviada às linhas de injeção, através de atuação sobre as próprias válvulas dessas linhas;
- Controle *anti-surge* atuando nas válvulas de reciclo de cada estágio.

Na seção a seguir apresentar-se-á de forma sucinta a modelagem matemática da estação de compressão aqui descrita.

### 3.2 MODELAGEM MATEMÁTICA FENOMENOLÓGICA

O principal objetivo do modelo matemático desenvolvido é possibilitar uma boa compreensão da dinâmica do sistema contínuo, permitindo assim desenvolver estratégias de controle avançado para a estação de compressão. Algumas equações serão aqui descritas. O modelo completo de equações é apresentado no [Apêndice C](#).

#### 3.2.1 Modelagem termodinâmica

Um modelo termodinâmico minucioso pode ser uma tarefa complexa de realizar devido à falta de informação. Todavia, algumas relações entre variáveis por terem uma dinâmica tão rápida em relação às demais, podem ser consideradas insignificantes em termos de desenvolvimento de técnicas de controle. Além disso, existem variáveis não medidas na estação e que não são utilizadas no projeto do sistema de controle<sup>1</sup>. Assim, algumas hipóteses são levantadas no sentido de simplificar, em parte, um modelo termodinâmico completo:

- Não há perda de energia nos trocadores de calor;
- Não há troca de massa entre fase líquida e gasosa nos vasos de pressão;
- A compressibilidade do gás é considerada constante.

As dinâmicas das pressões de sucção em cada estágio de compressão são resultantes do balanço de massa na entrada e saída dos vasos de pressão, e dependem de características térmicas do gás. Para cada linha de compressão  $i \in [1, 2]$ , e cada estágio de compressão  $j \in [1, 2, 3]$ , obtém-se:

$$\frac{dP_s^{ij}}{dt} = \frac{RZT_s^{ij}}{M_W V_{ij}} \cdot (\sum m_{IN}^{ij} - \sum m_{OUT}^{ij}) \quad (3.1)$$

onde  $R$  é a constante universal dos gases,  $Z$  é o fator de compressibilidade do gás,  $M_W$  é o seu peso molecular.  $T^{ij}$  representa a temperatura medida na mesma posição onde se mede a pressão.  $V_{ij}$  é o volume do vaso e  $m_{IN}^{ij}$  e  $m_{OUT}^{ij}$  representam as vazões mássicas de entrada e saída dos vasos de pressão, respectivamente.

---

<sup>1</sup>A menos que técnicas de estimação de variáveis fossem utilizadas para este fim.

Já as dinâmicas da pressão de sucção no vaso de entrada,  $P_E$ , e no vaso de saída (*header*) da estação,  $P_H$ , seguem a mesma característica, e são dadas por:

$$\frac{dP_E}{dt} = \frac{RZT_E}{M_W V_E} \cdot (m_E - m_{flare} - m_E^1 - m_E^2) \quad (3.2a)$$

$$\frac{dP_H}{dt} = \frac{RZT_E}{M_W V_H} \cdot (m_{TC}^{s1} + m_{TC}^{s2} - m_{exp} - m_{inj}^1 - m_{inj}^2) \quad (3.2b)$$

onde  $m_E$  é a vazão de entrada da linha, vinda da produção dos poços,  $m_{flare}$  representa a vazão de gás sendo levada ao queimador (*flare*), e  $m_E^1$  e  $m_E^2$ , as vazões de entrada das linhas de compressão 1 e 2, respectivamente. Além disso, assume-se volumes de vasos idênticos, isto é,  $V_E = V_H = V_{ij}$ .

Na saída,  $m_{exp}$ ,  $m_{inj}^1$  e  $m_{inj}^2$  são, respectivamente, as vazões enviadas às linhas de exportação e às duas linhas de injeção de gás. Já as vazões  $m_{TC}^{s1}$  e  $m_{TC}^{s2}$  correspondem à saída de gás dos trocadores de calor ao final de cada linha de compressão, cujo balanço de massa fornece, por exemplo, para a linha 1:

$$m_{TC}^{s1} = m_{CP}^{13} - m_{rc}^{13} \quad (3.3)$$

onde  $m_{CP}^{13}$  é a vazão que passa pelo compressor no terceiro estágio, e  $m_{rc}^{13}$  a vazão de reciclo do mesmo estágio.

As vazões de entrada em cada linha de compressão, resultantes das dinâmicas de pressão nos vasos de entrada e do primeiro estágio de cada linha, são modeladas através de equações *ISA-standard*<sup>2</sup> para o cálculo de vazões em válvulas, que são tipicamente usadas em trabalhos de modelagem de compressores, tal como em [28] e [29]. Como exemplo, para a linha 1, primeiro estágio, obtém-se:

$$m_E^1 = k_{TC}^1 \sqrt{\frac{M_W}{RZT_E}} \sqrt{P_E(P_E - P_s^{11})} \quad (3.4)$$

onde  $P_E$  e  $T_E$  são a pressão e temperatura tomadas no vaso de entrada, e  $k_{TC}^1$ , é uma constante que contribui para a modelagem das dimensões do trocador de calor.

A vazão de saída para o queimador,  $m_{flare}$ , é não nula somente se a pressão no vaso de entrada for superior a um limite pré-definido, isto é, se  $P_E > P_E^{max}$ , e é modelada também por uma

<sup>2</sup>Padrão para dimensionamento de válvulas de controle.

equação estática do tipo *ISA-standard*, seguida de uma dinâmica de primeira ordem, tal como equação (3.5).

$$m_{flare}^N = k_{flare} \sqrt{\frac{M_W}{RZT_E}} \sqrt{P_E(P_E - P_E^{max})} \quad (3.5a)$$

$$\frac{dm_{flare}}{dt} = \frac{m_{flare}^N - m_{flare}}{\tau_{flare}} \quad (3.5b)$$

onde  $m_{flare}^N$  representa a vazão de regime destinada ao queimador,  $k_{flare}$  é uma constante que modela a perda de carga na válvula da linha de *flare*, e  $\tau_{flare}$  corresponde à constante de tempo da dinâmica da vazão desta linha.

As pressões de descarga  $P_d^{ij}$  e as vazões  $m_{CP}^{ij}$  de cada estágio de compressor são calculadas a partir de informação obtida dos mapas de compressão, fornecidas pelos fabricantes. Estes mapas apresentam curvas estáticas, como as apresentadas na Figura 3.3 que relacionam, para cada estágio de compressão, a razão (ganho) de compressão  $r^{ij} = P_d^{ij}/P_s^{ij}$ , e a velocidade angular  $w^i$  de rotação do compressor da linha  $i$  com a vazão volumétrica  $Q^{ij}$  que o atravessa.

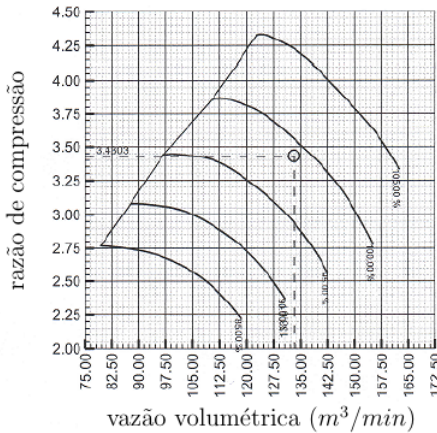


Figura 3.3: Mapa de compressão. A linha que conecta as curvas corresponde ao limite de operação do estágio, denominada linha de *surge*. (Fonte: [2]).

Estas curvas são modeladas da seguinte maneira: considere o primeiro estágio da linha de compressão 1. Definem-se as variáveis

$r^{11} = P_d^{11}/P_s^{11}$ ,  $y_1 = w^1/w_N^1$ , e  $x_{11} = Q^{11}/w_N^1$ , onde  $w_N^1$  é a velocidade angular nominal do compressor da linha 1. Obtém-se então uma curva normalizada para esta velocidade, a partir do seguinte polinômio:

$$r^{11} = a^{11} + b^{11}x_{11} + c^{11}y_1 + d^{11}x_{11}^2 + e^{11}y_1 + f^{11}y_1^2 \quad (3.6)$$

onde os coeficientes  $a^{11}$ ,  $b^{11}$ ,  $c^{11}$ ,  $d^{11}$ ,  $e^{11}$  e  $f^{11}$  são obtidos utilizando interpolação polinomial a partir dos pontos obtidos nos mapas de compressão [6].

A pressão de descarga é obtida simplesmente utilizando-se a razão de compressão, como mostra a equação (3.7).

$$P_d^{11} = r^{11}P_s^1 \quad (3.7)$$

A vazão volumétrica  $Q^{11}$  pode ser obtida computando-se as raízes do polinômio na equação (3.6), para dada razão de compressão  $r^{11}$ . Considerando uma operação normal do compressor, toma-se a raiz maior magnitude,  $x_{11}^*$ . Para obtenção da vazão mássica, multiplica-se  $Q^{11}$  pela massa específica do gás, tal como descrito pelas igualdades (3.8) abaixo.

$$Q^{11} = x_{11}^* w_1^N \quad (3.8a)$$

$$m_{CP}^{11} = \frac{P_s^{11} M_W}{R Z T_s^{11}} Q^{11} \quad (3.8b)$$

A quantidade de energia na entrada dos trocadores de cada estágio,  $W_{in}^{ij}$ , é dada pela soma da energia em forma de calor proveniente do estágio de compressão precedente,  $q_{CP}^{(i-1)(j-1)}$ , via recirculação,  $q_{rc}^{ij}$ , e correspondente ao fluido refrigerante do próprio trocador,  $q_c^{ij}$ , isto é:

$$W_{in}^{ij} = q_{CP}^{(i-1)(j-1)} + q_{rc}^{ij} + q_c^{ij} \quad (3.9a)$$

$$W_{in}^{ij} = m_{CP}^{(i-1)(j-1)} c_g T_d^{(i-1)(j-1)} + m_{rc}^{ij} c_g T_d^{ij} + \dots \\ \dots + \phi_c^{ij} m_c^{\max, ij} c_c T_c^{ij} \quad (3.9b)$$

onde  $c_g$  é o calor específico do gás,  $c_c$ , o calor específico do fluido de resfriamento do trocador de calor,  $T_d^{ij}$  é a temperatura de descarga do compressor no mesmo estágio, e  $m_{CP}^{(i-1)(j-1)}$  é a vazão de gás de

entrada do trocador, proveniente do compressor precedente, ou da entrada da linha ( $m_E^i$ ), para o caso do primeiro estágio.

A variável  $\phi_c^{ij}$  corresponde à abertura da válvula que permite a vazão do fluido de resfriamento no trocador de calor, e  $m_c^{\max,ij}$  é a máxima vazão de fluido de resfriamento. Na Figura 3.4, identifica-se o sentido de circulação das vazões no primeiro de estágio da linha de compressão 1, e apresentam-se os pontos de tomada de pressão e temperatura no estágio.

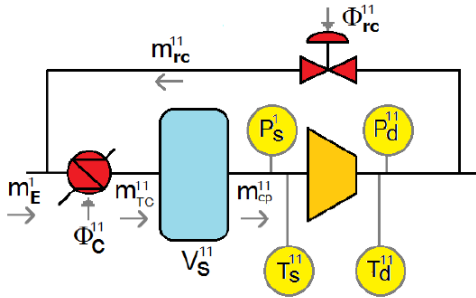


Figura 3.4: Estágio de compressão, identificando-se as vazões que o circulam. (Fonte: [6]).

Em regime permanente, considera-se que a temperatura do fluido refrigerante,  $T_c^{ij}$ , é igual à do gás na saída do trocador de calor, bem como à temperatura do gás na sucção do estágio de compressão correspondente,  $T_{ss}^{ij}$ . Desta forma, a quantidade de energia em forma de calor na saída dos trocadores,  $W_{out}^{ij}$ , corresponde a:

$$W_{out}^{ij} = q_{TC}^{ij} + q_c^{ij} \quad (3.10a)$$

$$W_{out}^{ij} = (m_{TC}^{ij} c_g + \phi_c^{ij} m_c^{\max,ij} c_c) T_{ss}^{ij} \quad (3.10b)$$

onde  $m_{TC}^{ij}$  é a vazão de saída do trocador.

Quanto à dinâmica da temperatura na sucção de cada estágio,  $T_s^{ij}$ , considera-se um modelo primeira ordem, com um ganho estático não linear que deve atender as condições de equilíbrio de energia em regime permanente.

Admitindo-se a ausência de perdas de energia, deve-se satisfazer, no equilíbrio, a condição  $W_{in}^{ij} = W_{out}^{ij}$ . Desta igualdade é possível calcular a temperatura de regime (ou ganho estático) na sucção,  $T_{ss}^{ij}$ . Assim, para o modelo da temperatura na sucção, utiliza-se uma

equação não linear estática e uma dinâmica de primeira ordem, dada por:

$$T_{ss}^{ij} = \frac{m_{rc}^{ij} c_g T_d^{ij} + m_{CP}^{(i-1)(j-1)} c_g T_d^{(i-1)(j-1)} + \phi_c^{ij} m_c^{ij} c_c T_c^{ij}}{m_{TC}^{ij} c_g + \phi_c^{ij} m_c^{\max, ij} c_c} \quad (3.11a)$$

$$\frac{dT_s^{ij}}{dt} = \frac{T_{ss}^{ij} - T_s^{ij}}{\tau_{T_s^{ij}}} \quad (3.11b)$$

onde  $\tau_{T_s^{ij}}$  representa uma constante de tempo, cujo valor adotado considera informação prática da medição em tempo real desta temperatura em estações de compressão.

As temperaturas do gás na saída dos trocadores de calor no terceiro estágio de cada linha,  $T^{s1}$  e  $T^{s2}$ , são computadas seguindo raciocínio análogo ao das temperaturas de sucção. As equações completas estão descritas no [Apêndice C](#).

A modelagem das temperaturas do gás na entrada do *header*,  $T_H$ , e da temperatura de descarga de cada estágio de compressão,  $T_d^{ij}$ , são baseadas nos trabalhos de [Budinis and Thornhill \[8\]](#) e [Helvoirt \[30\]](#) que consideram equações estáticas para estas variáveis.

A temperatura do gás no *header* varia pouco, pois as temperaturas na saída dos trocadores são controladas. Desta forma estima-se sua magnitude através de uma média das vazões dos trocadores de calor, ponderada pelas próprias temperaturas do gás na saída, isto é:

$$T_H = \frac{m_{TC}^{s1} T^{s1} + m_{TC}^{s2} T^{s2}}{m_{TC}^{s1} + m_{TC}^{s2}} \quad (3.12)$$

Já a temperatura de descarga é dependente da temperatura na sucção e da razão de compressão, podendo ser aproximadamente por:

$$T_d^{ij} = T_s^{ij} (r^{ij})^{\sigma^{ij}} \quad (3.13)$$

onde o expoente  $\sigma^{ij}$  é um coeficiente que depende das características térmicas do gás como a razão de calor específico, e da eficiência politrópica de compressão [30].

A modelagem da pressão na linha de exportação é simplificada para evitar a utilização de equações diferenciais parciais. Esta pressão,  $P_{exp}$ , foi modelada como uma combinação de perda de carga  $P_{exp}^{pc}$ , e a pressão devido à acumulação  $P_{exp}^{ac}$  na linha. Assume-se que, na linha de exportação, já flui uma vazão  $m_{Lexp}$  que in-

depende da vazão de saída do *header* de exportação de gás. Além disso, assume-se que a vazão de exportação do *header* na saída da linha de exportação  $m_{exp}^s$  será igual em regime permanente à vazão entrando na mesma,  $m_{exp}$ , mas apresentando uma dinâmica de primeira ordem em relação a entrada. Com essas considerações, descrevem-se as pressões e as vazões da linha de exportação como:

$$P_{exp} = P_{exp}^{ac} + P_{exp}^{pc} \quad (3.14a)$$

$$\frac{dm_{exp}^s}{dt} = \frac{m_{exp} - m_{exp}^s}{\tau_{exp}} \quad (3.14b)$$

$$\frac{dP_{exp}^{ac}}{dt} = k_{exp}^{ac} (m_{exp} - m_{exp}^s) \quad (3.14c)$$

$$P_{exp}^{pc} = k_{exp}^{pc} (m_{exp}^s + m_{Lexp})^2 \quad (3.14d)$$

onde  $\tau_{exp}$  é a constante de tempo relacionada com a dinâmica da vazão na saída da linha de exportação. A constante  $k_{exp}^{pc}$  depende do comprimento da linha de exportação, do diâmetro da linha de exportação, do fator de atrito e da pressão média da linha, e a constante  $k_{exp}^{ac} = \frac{ZRT_H}{M_W V_{Lexp}}$ , sendo  $V_{Lexp}$  o volume da linha de exportação [29].

O cálculo da vazão  $m_{exp}$ , em função da diferença entre as pressões  $P_H$  e  $P_{exp}$ , é realizado seguindo o mesmo procedimento utilizado para a determinação das vazões da linha de entrada e de *flare*. Isto é, utilizando-se equações ISA para vazões em válvulas de controle para fluidos compressíveis.

As vazões  $m_{inj}^1$  e  $m_{inj}^2$  das linhas de injeção são modeladas como vazões controladas para utilização em processos de injeção para gas-lift, por exemplo. As contra-pressões  $P_{L1}^{ac}$  e  $P_{L2}^{ac}$  devido acumulação de fluido na saída das linhas ( $m_{inj}^{s1}$  e  $m_{inj}^{s2}$ ) desses processos são modeladas de forma análoga à linha de exportação, mas considerando-se as pressões nas saídas das linhas,  $P_{L1}^s$  e  $P_{L2}^s$ , pressões altas e constantes durante a simulação. Para a linha 1, modela-se:

$$P_{L1} = P_{L1}^{ac} + k_{L1} (m_{inj}^{s1})^2 + P_{L1}^s \quad (3.15a)$$

$$\frac{dm_{inj}^{s1}}{dt} = \frac{m_{inj}^1 - m_{inj}^{s1}}{\tau_{L1}} \quad (3.15b)$$

$$\frac{dP_{L1}^{ac}}{dt} = k_{L1}^{ac} (m_{inj}^1 - m_{inj}^{s1}) \quad (3.15c)$$

$$k_{L1}^{ac} = \frac{ZRT_H}{M_W V_{L1}} \quad (3.15d)$$



Para a linha 2:

$$P_{L2} = P_{L2}^{ac} + k_{L2} m_{p2-din}^2 + P_{L2}^s \quad (3.16a)$$

$$\frac{dm_{inj}^{s2}}{dt} = \frac{m_{inj}^2 - m_{inj}^{s2}}{\tau_{L1}} \quad (3.16b)$$

$$\frac{dP_{L2}^{ac}}{dt} = k_{L2}^{ac} (m_{inj}^2 - m_{inj}^{s2}) \quad (3.16c)$$

$$k_{L1}^{ac} = \frac{ZRT_H}{M_W V_{L2}} \quad (3.16d)$$

onde  $k_{L1}^{ac}$  e  $k_{L2}^{ac}$  são constantes que dependem do comprimento da linha, diâmetro do duto, fator de atrito e pressão média na linha,  $V_{L1}$  é o volume da linha 1 e  $V_{L2}$  é o volume da linha 2. As constantes de tempo  $\tau_{L1}$  e  $\tau_{L2}$  representam a dinâmica das vazões das linhas de injeção. A modelagem destas linhas de saída da estação, exportação e compressão, foi fortemente baseada nos trabalhos [10, 28, 29, 30].

### 3.2.2 Modelagem do controle regulatório

Conforme já mencionado, os controladores PID existentes na estação de compressão de gás desempenham um controle local regulatório do processo, tal como detalhado a seguir.

#### 3.2.2.1 Controle da pressão de sucção na entrada das linhas de compressão

Sabe-se, das relações obtidas a partir dos mapas de compressão, que é possível manipular a vazão que passa através de um estágio do compressor  $m_{CP}^{ij}$  para levar uma pressão de sucção de entrada  $P_s^{i1}$  a um valor de referência desejado. Por outro lado,  $m_{CP}^{ij}$  varia com a velocidade angular do eixo do compressor  $w_i$ , que por sua vez varia com a potência aplicada ao compressor [2, 6], através das equações:

$$\frac{dw_i}{dt} = \frac{1}{J_i} (\tau_a^i - \sum_{j=1}^3 \tau_c^{ij}) \quad (3.17a)$$

$$\tau_a^i = \frac{u_P^i P_{ot}^i}{w_i} \quad (3.17b)$$

$$\tau_c^{ij} = m_{CP}^{ij} \mu^i w_i R_i^2 \quad (3.17c)$$

onde  $J_i$  e  $\tau_a^i$  são, respectivamente, o momento de inércia e o torque aplicado ao eixo do compressor  $i$ ,  $\tau_c^{ij}$  é o torque exercido pela vazão  $m_{CP}^{ij}$  que atravessa um estágio de compressão  $j$ .  $Pot_N^i$  é a potência nominal, e  $\mu^i$  e  $R_i$  são, respectivamente, o fator de escorregamento e o raio do impelidor do compressor  $i$ . Por fim,  $u_P^i, u_P^i \in (0, 1)$  é a ação de controle que representa um percentual da potencia nominal  $Pot_N^i$  a ser aplicada ao eixo do compressor  $i$ .

A ação de controle tomada a partir da leitura de cada pressão de sucção  $P_s^{i1}$  é dada por:

$$e_{P_s^{i1}} = P_s^{i1SP} - P_s^{i1} \quad (3.18a)$$

$$u_P^{i1} = K_P^{P_s^{i1}} e_{P_s^{i1}} + K_I^{P_s^{i1}} Ie_{P_s^{i1}} \quad (3.18b)$$

$$\frac{dIe_{P_s^{i1}}}{dt} = e_{P_s^{i1}} \quad (3.18c)$$

onde  $e_{P_s^{i1}}$  é o erro de controle entre a pressão de sucção  $P_s^{i1}$  e seu valor de referência  $P_s^{i1SP}$ ,  $K_P^{P_s^{i1}}$  e  $K_I^{P_s^{i1}}$  são, respectivamente, os ganhos proporcional e integral do controlador, e  $Ie_{P_s^{i1}}$  é a integral do erro de controle.

A pressão do header é controlada em um valor desejado,  $P_H^{SP}$ . Isso é feito manipulando-se a vazão enviada para a linha de exportação,  $m_{exp}$ , através do acionamento da abertura ( $\Phi_{exp}$ ) de uma válvula da linha. O controle **Proporcional Integral (PI)** é descrito na equação (3.19).

$$e_{P_H} = P_H^{SP} - P_H \quad (3.19a)$$

$$\Phi_{exp} = K_P^{P_H} e_{P_H} + K_I^{P_H} Ie_{P_H} \quad (3.19b)$$

$$\frac{dIe_{P_H}}{dt} = e_{P_H} \quad (3.19c)$$

tal que  $e_{P_H}$  é o erro entre a leitura da variável  $P_H$  e seu *set-point*  $P_H^{SP}$ .  $K_P^{P_H}$  é o ganho proporcional e  $K_I^{P_H}$  o ganho integral do controlador PI, e  $Ie_{P_H}$  é o estado correspondente à integral do erro  $e_{P_H}$ .

### 3.2.2.2 Controle da temperatura de sucção em cada estágio

O controle das temperaturas de sucção de cada estágio de compressão,  $T_s^{ij}$ , e da saída dos trocadores de calor no último estágio,  $T^{si}$ , é realizado por uma ação **PI**, similar ao controle de pressão,

que atua através da abertura de uma válvula  $\Phi_c$  que permite a vazão de um fluido refrigerante pelo trocador de calor. Estimando-se as temperaturas através da dinâmica de primeira ordem descrita pela equação (3.11), a ação de controle desta grandeza em cada estágio é calculada por:

$$e_{T_s}^{ij} = T_s^{ijSP} - T_s^{ij} \quad (3.20a)$$

$$\Phi_c^{ij} = K_P^{T_s^{ij}} e_{T_s}^{ij} + K_I^{T_s^{ij}} Ie_{T_s}^{ij} \quad (3.20b)$$

$$\frac{dIe_{T_s}^{ij}}{dt} = e_{T_s}^{ij} \quad (3.20c)$$

onde  $e_{T_s}^{ij}$  é o erro de controle entre a temperatura de sucção  $T_s^{ij}$  e seu valor desejado  $T_s^{ijSP}$ ,  $K_P^{T_s^{ij}}$  e  $K_I^{T_s^{ij}}$  são, respectivamente, os ganhos proporcional e integral do controlador PI, e  $Ie_{T_s}^{ij}$  é a integral do erro de controle.

Similarmente, calculam-se as aberturas  $\Phi_c^{si}$  de válvulas para os trocadores de calor na saída das linhas de compressão:

$$e_{T^{si}} = T^{siSP} - T^{si} \quad (3.21a)$$

$$\Phi_c^{si} = K_P^{T^{si}} e_{T^{si}} + K_I^{T^{si}} Ie_{T^{si}} \quad (3.21b)$$

$$\frac{dIe_{T^{si}}}{dt} = e_{T^{si}} \quad (3.21c)$$

tal que  $e_{T^{si}}$  é o erro de controle entre a temperatura de saída da linha  $T^{si}$  e seu correspondente valor de referência  $T^{siSP}$ ,  $K_P^{T^{si}}$  e  $K_I^{T^{si}}$  são os ganhos proporcional e integral do controlador PI, e  $Ie_{T^{si}}$  é a integral do erro de controle da temperatura de saída da linha de compressão  $i$ .

### 3.2.2.3 Controle da vazão das linhas de injeção

As aberturas das válvulas ( $\Phi_{inj1}$  e  $\Phi_{inj2}$ ), que controlam as vazões de injeção destinadas ao processo de gas-lift,  $m_{inj}^1$  e  $m_{inj}^2$ , também são reguladas com controladores PI, com as seguintes leis de controle.

Para a linha 1:

$$e_{inj1} = m_{inj1}^{SP} - m_{inj1}^1 \quad (3.22a)$$

$$\Phi_{inj1} = K_P^{inj1} e_{inj1} + K_I^{inj1} Ie_{inj1} \quad (3.22b)$$

$$\frac{dIe_{inj1}}{dt} = e_{inj1} \quad (3.22c)$$

onde  $m_{inj1}^{SP}$ : é o set-point da vazão mássica de gás na linha 1,  $K_P^{inj1}$  é o ganho proporcional, e  $K_I^{inj1}$  é o ganho integral do controlador PI de vazão na linha 1.

Para a linha 2:

$$e_{inj2} = m_{inj2}^{SP} - m_{inj}^2 \quad (3.23a)$$

$$\Phi_{inj2} = K_P^{inj2} e_{inj2} + K_I^{inj2} I e_{inj2} \quad (3.23b)$$

$$\frac{dI e_{inj2}}{dt} = e_{inj2} \quad (3.23c)$$

onde  $m_{inj2}^{SP}$  é o set-point da vazão mássica de gás na linha 2,  $K_P^{inj2}$  é o ganho proporcional, e  $K_I^{inj2}$  é o ganho integral do controlador PI de vazão na linha 2.

#### 3.2.2.4 Controle *anti-surge*

Compressores centrífugos trabalham com uma faixa limitada de vazão volumétrica ( $Q$ ) em sua sucção. Isso porque para baixas vazões um fenômeno denominado *surge* pode ocorrer, resultando em uma operação instável do compressor com reversão de vazão e flutuações de pressão no sistema [6]. Como já mencionado, o funcionamento de um compressor pode ser analisado por meio de seu mapa de compressão. Os pontos de cada curva do mapa para os quais a razão entre o ganho de pressão (razão de compressão) e a vazão é nula, isto é  $dr/dQ = 0$ , indicam o limite de operação estável, livre de *surge*, do compressor. Ao conectarem-se os pontos de cada curva por uma linha, encontra-se a chamada *linha de surge*, como mostra a Figura 3.5. Assim, todos os pontos de operação situados à esquerda desta curva levam o compressor a uma operação em *surge*.

Uma técnica de controle *anti-surge* típica consiste em traçar uma linha paralela à linha de *surge*, estabelecendo um afastamento, ou folga em relação a esta. Ou seja, é uma linha de *surge* de folga, ou de segurança, tal como a linha tracejada na Figura 3.5.

Se, no funcionamento do compressor, a vazão volumétrica  $Q$  situar-se ao lado esquerdo da linha de segurança, uma vazão de reciclo  $Q_{rc}$  deve ser implementada para levar a vazão  $Q$  até um valor de segurança (um *set-point*) situado sobre esta linha. Já se o ponto situar-se à direita da linha, não há necessidade de atuação. A ideia é garantir que a vazão no compressor nunca seja inferior à vazão estabelecida pela linha de segurança [2].

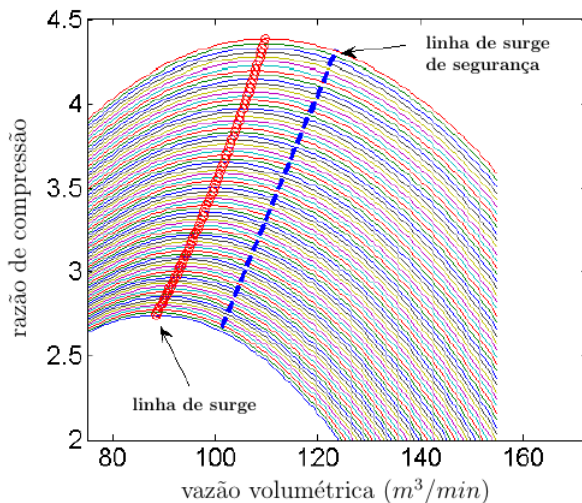


Figura 3.5: Mapa de compressão indicando a linha de surge e a linha de segurança.

Isso deverá assegurar um afastamento do limiar (linha) de *surge* que compense a incerteza do modelo utilizado, bem como a velocidade de resposta do sistema de controle. A quantidade de gás necessária para levar uma vazão  $Q^{ij}$  que passa por um estágio de compressão, para a linha de segurança é obtida manipulando-se a abertura da válvula de reciclo ( $\Phi_r^{ij}$ ) do estágio de compressão em questão.

Para a obtenção da linha de segurança de *surge*, primeiramente determinam-se os pontos tais que  $dr^{ij}/dQ^{ij} = 0$ , utilizando-se os polinômios modelados para as curvas de compressão, a fim de obter a vazão sobre a linha de *surge*, denominada  $Q_{surge}^{ij}$ , para cada estágio  $j$  do compressor  $i$ . Com esta informação, a vazão sobre a linha de segurança, denominada  $Q_{surge}^{ijSP}$  e que é *set-point* do sistema de controle *anti-surge*, é obtida utilizando-se uma folga  $f^{ij}$ , de maneira que:

$$Q_{surge}^{ijSP} = Q_{surge}^{ij}(1 + f^{ij}) \quad (3.24)$$

Para os cenários de simulação deste trabalho, utilizou-se uma folga constante de 10 % para todos os estágios. É interessante observar que, no contexto do sistema de controle de compressão, existe um compromisso entre robustez e desempenho. Quanto maior a

folga utilizada, maior será a vazão de reciclo e conseqüentemente menor será a eficiência do sistema. Entretanto, quanto menor a folga, aumenta-se a probabilidade de ocorrência de *surge*, e de uma possível falha no compressor. O que se busca com o desenvolvimento de técnicas de controle avançado é justamente atuar sobre o sistema de controle regulatório a fim de permitir uma operação mais eficiente, com menos vazão de reciclo, mas mantendo-se a segurança do processo, evitando paradas não programadas dos compressores.

A lei de controle *anti-surge* calculada a partir da leitura da vazão de compressão  $Q^{ij}$ , para cada estágio de compressão é dada por:

$$e_{surge}^{ij} = Q_{surge}^{ijSP} - Q^{ij} \quad (3.25a)$$

$$\Phi_r^{ij} = K_P^{s^{ij}} e_{surge}^{ij} + K_I^{s^{ij}} I e_{surge}^{ij} \quad (3.25b)$$

$$\frac{dI e_{surge}^{ij}}{dt} = e_{surge}^{ij} + d_{AW}^{ij} \quad (3.25c)$$

onde  $e_{surge}^{ij}$  é o desvio entre a vazão  $Q^{ij}$  que passa pelo estágio  $j$  do compressor  $i$ , e a vazão calculada para a linha de segurança  $Q_{surge}^{ijSP}$ . A abertura da válvula de reciclo  $\Phi_r^{ij}$  é dada por uma ação de controle **PI** com ganhos proporcional  $K_P^{s^{ij}}$  e integral  $K_I^{s^{ij}}$ . A variável correspondente à integral do erro de controle é dada por  $I e_{surge}^{ij}$ .

Devido aos limites de saturação da abertura  $\Phi_r^{ij} \in [0, 1]$  da válvula, a ação integral é corrigida por uma ação *anti wind-up*,  $d_{AW}^{ij}$ , que computa o desvio entre a abertura calculada pela lei de controle e a abertura real da válvula para dada limitação por saturação [31]. Esta ação *anti wind-up* resultante é obtida por:

$$d_{AW}^{ij} = \begin{cases} (1 - \Phi_r^{ij}), & \text{se } \Phi_r^{ij} > 1 \\ -\Phi_r^{ij}, & \text{se } \Phi_r^{ij} < 0 \end{cases} \quad (3.26)$$

O cálculo do ganho estático da vazão de reciclo  $m_r^{ij}$  resultante, deve seguir a expressão *ISA* para válvulas de controle para gases e vapor:

$$m_r^{ij} = K_{ss}^{ij} \Phi_r^{ij} C V_N P_d^{ij} \sqrt{\frac{M_W x_P}{Z R T_d^{ij}}} \quad (3.27)$$

onde  $K_{ss}^{ij}$  é uma constante característica da válvula,  $C V_N$  é o coeficiente nominal da válvula (100 % aberta), e  $x_P$  é a razão de queda

de pressão a montante e à jusante da válvula.

A velocidade de abertura das válvulas de reciclo foi suposta seguir uma dinâmica de primeira ordem, descrita pela seguinte equação diferencial:

$$\frac{dm_{rc}^{ij}}{dt} = \frac{m_{rc}^{ij} - m_r^{ij}}{\tau_{m_r^{ij}}} \quad (3.28)$$

Os valores dos parâmetros utilizados em todas as equações apresentadas neste capítulo são informados no [Apêndice B](#). Para mais informações sobre a obtenção das equações do modelo bem como a justificativa da utilização de cada uma, consultar [2].

### 3.3 COMPLEXIDADE NUMÉRICA DO MODELO

O modelo da estação de compressão abordado neste capítulo envolve uma série de dificuldades ou obstáculos numéricos que o tornam complexo de resolver. Cada um deles será discutido nesta seção.

O primeiro ponto a ser levantado é a quantidade de equações do problema a resolver. Como pode ser verificado no [Apêndice C](#), o modelo fenomenológico completo contempla 50 equações diferenciais [ODEs](#) de primeira ordem e 84 equações algébricas, que envolvem termos não lineares (tal como no cálculo das vazões nas equações (3.4), (3.6) e (3.27)). As equações algébricas atuam como restrição à resolução numérica das [ODEs](#), devendo cada solução computada satisfazer estas equações. Desta forma, deve-se resolver, a cada ciclo de amostragem do modelo, um [PVI](#)s dado por um sistema de [DAE](#) não linear de índice unitário, apresentado pelo sistema de equações (2.4).

O fato do sistema ser matematicamente extenso implica que a cada chamada do modelo [DAE](#), um grande esforço computacional seja gasto, pois para todo estado computado através de um método numérico que deve aproximar as derivadas a partir do cálculo da dinâmica  $f$  diversas vezes, é necessário ainda resolver um sistema de equações não lineares, descrito por  $g$ , para computar as variáveis algébricas. Fortuitamente, este esforço pode ser reduzido. Como visto na seção 2.1 do [Capítulo 2](#), o fato do sistema [DAE](#) possuir índice 1 faz com que este possa ser resolvido como um sistema [ODE](#) com restrições. Desta forma, é possível simplesmente aplicar um método numérico para obter o estado  $x$ , resolvendo o sistema [ODE](#) (2.4a),

e substituir o resultado em  $g$  para aferir as variáveis algébricas  $p$ , em todo intervalo de integração  $t, t \in [t_0, t_f]$ .

A segunda dificuldade na resolução encontra-se no fato de que muitos estados possuem dinâmica diferente. Do ponto de vista de sistemas de controle, isto é essencial. Toma-se como exemplo o controle da pressão de sucção, o qual para ser possível, torna-se necessário que a dinâmica de atuação na velocidade angular do compressor seja mais rápida que a dinâmica da grandeza controlada.

Entretanto, numericamente o problema de integração a ser resolvido torna-se rígido, sendo portanto necessário o uso de métodos numéricos implícitos ou semi-implícitos para que se reduza a carga computacional resultante da limitação de tamanho de passo de integração por requisitos de estabilidade quando métodos puramente explícitos são utilizados.

Por fim, para avaliar e simular o sistema de controle avançado, perturbações são impostas à vazão de entrada, retirando o sistema, em determinado intervalo de tempo, de seu ponto de operação, variando-se assim sua dinâmica. Isto faz com que um método numérico deva reduzir o passo de integração de modo a capturar com precisão as variações de dinâmica impostas pela perturbação. Esta tarefa torna-se particularmente mais custosa quando requisitos de precisão são muito apertados, tal como uma tolerância para o erro local de aproximação,  $\mathcal{E} < 10^{-3}$  ou menor, que serão estudadas neste trabalho. Assim sendo, a fim de tornar o *solver* de integração numérica mais eficiente, é proposto o uso de métodos numéricos de baixa ordem, para reduzir o número de cálculos do sistema, ao calcular cada aproximação, e com tamanho do passo variável (controlado) sendo capaz de adaptar-se a possíveis modificações da dinâmica do sistema impostas pela aplicação de perturbações, fornecendo assim soluções eficientes e precisas.

### 3.4 COMENTÁRIOS FINAIS

Este capítulo apresentou de forma sucinta a modelagem matemática da estação de compressão de gás de uma plataforma de produção, a fim de proporcionar ao leitor uma visão geral do sistema real, suas relações termodinâmicas e seu controle regulatório existente. O sistema DAE completo, bem como suas condições iniciais estão descritos no Apêndice C e Apêndice B, respectivamente. A fim de proporcionar ao leitor uma noção da magnitude do problema abordado, foram identificados os aspectos numéricos que



---

agregam complexidade ao problema de integração, tomados como base para o desenvolvimento de metodologias mais eficientes, principal contribuição deste trabalho. No próximo capítulo, será descrito o desenvolvimento de metodologias mais eficientes para resolver **PVIs** complexos e rígidos, tal como o sistema de compressão.



## 4 CONTRIBUIÇÕES QUANTO À EFICIÊNCIA COMPUTACIONAL DE MÉTODOS NUMÉRICOS

Este capítulo apresenta as principais contribuições desta dissertação: são implementados algoritmos de integração numérica com elementos típicos tais como o uso de um método para calcular as aproximações e uma fórmula para estimar sua precisão. Além disso, para as soluções obtidas fora da precisão especificada, realiza-se uma ação corretiva nos parâmetros da formulação do método: aplica-se a teoria de controle linear para que o tamanho do passo de integração  $h$  utilizado no método forneça uma solução mais próxima de atender a precisão especificada.

Os algoritmos desenvolvidos são customizados em relação ao sistema DAE a resolver, ou seja, é obtida informação prévia do modelo matemático, visando a melhora da eficiência da integração numérica quando exigidas tolerâncias muito precisas para o erro local das soluções estimada.

### 4.1 ESCOLHENDO FÓRMULAS PARA CALCULAR APROXIMAÇÕES E ESTIMAR O ERRO LOCAL

Dentre tantas metodologias existentes na literatura, das quais algumas foram expostas no [Capítulo 2](#), quais seriam as mais eficientes para simular um modelo complexo?

Para responder esta questão é preciso levar em conta dois fatores que são os principais responsáveis pela elevada carga de processamento computacional durante a resolução de PVI's:

- (i) O sistema DAE apresenta uma estrutura numérica complexa: grande quantidade de equações diferenciais e algébricas, ODEs do tipo rígidas e não linearidades;
- (ii) Uma alta precisão numérica é exigida para o cálculo das aproximações – entenda-se por “alta”, tolerâncias menores que um milésimo, ou  $\mathcal{E} < 10^{-3}$ , para a estimativa do erro local.

Do item (i) já surgem dois pontos a considerar: primeiro, precisa-se conhecer previamente o modelo para escolher uma metodologia adequada para a integração. Isto não é algo novo, já que devido a grande quantidade de *solvers* existentes em pacotes comerciais, necessita-se algum conhecimento quanto à estrutura matemática e complexidade do sistema para definir-se qual *solver* de integração seria mais adequado para resolver tal problema.

Segundo, se o sistema apresenta uma estrutura numérica complexa, escolher um método de alta ordem, isto é, que necessite realizar chamadas do sistema em vários pontos para calcular cada aproximação  $\mathbf{x}_{n+1}$  em todo instante  $n$  do intervalo de integração, certamente agregará um elevado custo computacional à resolução numérica.

Além disso, adotar um método puramente explícito que não faça o uso de um tamanho de passo variável para resolver um sistema rígido, seguramente incrementará (e muito) o tempo de processamento do algoritmo, devido à limitação no tamanho de passo utilizado por estes métodos, conforme justificado na Seção 2.3.4.

Tomando agora o item (ii), por exigir-se um erro local muito pequeno, será necessário ou o uso de um algoritmo de mais alta ordem, obtendo-se assim mais informação para computar cada aproximação, calculando-se as funções  $\mathbf{f}$  e  $\mathbf{g}$  do sistema em mais pontos, ou realizar um número maior de passos (ou iterações) em cada subintervalo  $n$  utilizando-se tamanhos de passo  $h$  menores, a fim de estimar um erro local tão pequeno quanto mais próximas forem as soluções computadas adjacentes.

Por fim, mas não menos importante, estes itens em conjunto normalmente fazem com que o método numérico exceda o número máximo de iterações ( $N_{max}$ ), retornando aproximações pouco precisas ou confiáveis [14, 32].

Como mencionado anteriormente, *solvers* comerciais adotam em geral uma ordem variável do método, cujo valor máximo pode ser definido pelo usuário. Todavia, mantêm como padrão uma ordem consideravelmente alta de maneira a garantir como padrão uma aproximação precisa, mas em geral lenta. Exemplos destes *solvers* tradicionais apresentam algoritmos que utilizam os métodos de Runge-Kutta de 4ª ordem, a família de métodos preditores-corretores (de múltiplos passos) de Adams-Bashforth de 3ª e 4ª ordem, e os métodos de colocação de Gauss-Radau, com polinômios de Lagrange de 3ª a 5ª ordem.

Baseado neste conhecimento, adota-se neste trabalho o método ponderado, ou  $\theta$ -method, já apresentado pela equação (2.11). Nesta metodologia, a ponderação pode ser variada ( $\theta \in [0, 1]$ ), sendo assim uma metodologia adaptativa que compreende uma família de técnicas de baixa ordem, compreendendo, por exemplo, o método de Euler implícito quando  $\theta = 0$ , e o método de Euler puramente explícito quando  $\theta = 1$ , sendo um método semi-implícito no restante do intervalo.

Pode-se, portanto, resolver sistemas rígidos agregando-se um

controle de passo e/ou de ponderação, não necessitando assim um número elevado de chamadas do sistema, isto é, cálculos das equações das funções  $f$  e  $g$  do modelo [DAE](#).

Neste trabalho foi desenvolvida apenas uma metodologia com o controle da ponderação dos gradientes, pois verificou-se que a manipulação desta variável contribuía muito pouco com a eficiência do algoritmo quando comparada ao controle do tamanho do passo.

#### 4.1.1 Estabilidade do método ponderado

Retomando as ideias da subseção [2.3.2.1](#) e aplicando-se a aproximação linear local  $f_n = \Lambda x_n$ , no método ponderado [\(2.11\)](#), obtém-se:

$$x_{n+1} = x_n + h[\theta \Lambda x_n + (1 - \theta) \Lambda x_{n+1}] \quad (4.1a)$$

$$x_{n+1} = \frac{I + \theta h \Lambda}{I - (1 - \theta) h \Lambda} x_n \quad (4.1b)$$

A função de estabilidade do  $\theta$ -method é, portanto, dada por:

$$R(\bar{h}, \theta) = \frac{I + \theta \bar{h}}{I - (1 - \theta) \bar{h}}, \quad \bar{h} = h \Lambda \quad (4.2)$$

Para assegurar a estabilidade local deste método, necessita-se satisfazer:

$$\|R(\bar{h}, \theta)\| < I, \quad \bar{h} = h \Lambda \quad (4.3)$$

Observa-se que o resultado dessa condição fornece os valores localmente estáveis de tamanho do passo  $h$  e ponderação  $\theta$  podendo ser, portanto, manipulados por uma ação de controle.

Pode-se argumentar aqui que o fato de utilizar-se conhecimento prévio da dinâmica do sistema [DAE](#) pode favorecer o desempenho dos métodos desenvolvidos em relação a seus concorrentes. Lembra-se aqui, entretanto, que esta é uma alternativa ao fato de que estes algoritmos não são favorecidos por recursos computacionais de processamento e compilação, como os utilizados por muitos *solvers* comerciais e que, além disso, as metodologias implementadas são customizadas a resolver eficientemente um sistema [DAE](#) complexo de compressão, problema o qual as técnicas comerciais disponíveis não se apresentaram eficientes.

### 4.1.2 Estimativa do erro local

Afim de estimar o erro local, utiliza-se neste trabalho a estimativa do erro em sua forma relativa, tal como exposto pela equação (4.4) abaixo, onde são tomadas duas estimativas para computar cada solução, chamadas preditor  $((\mathbf{x}_{n+1})_p)$  e corretor  $((\mathbf{x}_{n+1})_c)$ .

Esta fórmula foi adotada porque no modelo da estação de compressão existem variáveis bastante dispares umas das outras em termos de magnitude, como por exemplo a pressão e a temperatura. Assim, para que o critério de precisão tenha o mesmo significado para todos os estados, o erro relativo torna-se mais conveniente, dado que desta forma normaliza-se os estados quanto à sua magnitude.

$$e_{n+1} = \left\| \frac{(\mathbf{x}_{n+1})_c - (\mathbf{x}_{n+1})_p}{(\mathbf{x}_{n+1})_c} \right\|_{\infty} \quad (4.4)$$

## 4.2 MÉTODO PONDERAÇÃO VARIÁVEL COM CONTROLE DO TAMANHO DO PASSO

Tomando-se o método ponderado ( $\theta$ -method) como padrão para calcular as aproximações, este algoritmo propõe-se a realizar um controle adaptativo muito simples, não somente do tamanho do passo de integração  $h$ , isto é, da distância entre as soluções subsequentes computadas, mas também da ponderação  $\theta$  sobre os gradientes (cálculos da função)  $f$  da dinâmica do sistema. Ou seja, realiza-se também o controle sobre a direção da predição linear realizada pelo método.

Esta solução foi implementada com o objetivo de tomar-se pouco tempo com algebrismos, e chamadas de sistema desnecessárias, visando atender a precisão especificada o mais rápido possível, tomando apenas informação do comportamento numérico da aproximação computada no passo anterior.

A Figura 4.1 exemplifica a aplicação do método. Note que, geometricamente, a solução computada toma a direção compreendida entre os gradientes (ou inclinação) da curva-solução, calculados nos instantes  $t_i$  e  $t_{i+1}$ .

O controle do tamanho do passo é baseado na condição de parada do método. Enquanto o erro local estiver acima da tolerância, diminui-se o tamanho do passo  $h$  em determinada quantidade percentual. Caso contrário, e ainda se o número de iterações do método for pequeno, aumenta-se  $h$  em outra determinada quantidade.

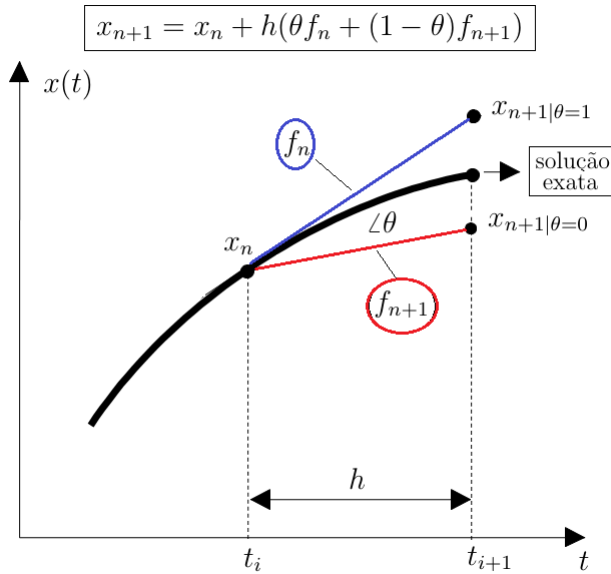


Figura 4.1: Exemplificação do método ponderado.

Assim, o tamanho do passo é decrescido de maneira a satisfazer rapidamente a condição de precisão, ou incrementado levando-se em conta a agilidade com que se obteve a aproximação passada. Os limites de magnitude desta variável levam em conta o período de amostragem  $T_s$  do modelo, tempo máximo disponibilizado para calcular cada solução. Adota-se como valor mínimo um centésimo deste intervalo, e como máximo o próprio valor de  $T_s$  decrescido do instante atual, isto é, dado o instante  $t_{i+1} = t_i + h_n$ , então  $h_{n+1} \in [T_s/100, T_s - t_{i+1}]$ .

O controle da ponderação  $\theta$  dos gradientes é realizado de maneira semelhante, mas completamente baseada no número de iterações ( $j$ ) tomadas pelo método: caso um número pequeno de iterações seja dado para aceitar-se uma solução precisa ( $e_{n+1} < \mathcal{E}$ ), permite-se aumentar a ponderação, de maneira a tornar o método mais explícito. Caso contrário, e se a solução tomar mais iterações do que um determinado valor  $N_i$ , é possível que uma parte rígida do sistema esteja sendo numericamente computada. Assim, permite-se decrementar a ponderação  $\theta$ , tornando o método ponderado mais implícito, e portanto, mais eficiente.

É muito importante salientar que ao alterar-se os valores tanto do tamanho do passo de integração  $h$ , quanto da ponderação dos gradientes  $\theta$ , modifica-se o domínio de estabilidade do método numérico, segundo a condição apresentada na equação (4.3). Portanto, é essencial garantir que o algoritmo não viole essa condição, pois, tal como visto na Seção 2.3.2, a instabilidade do método pode trazer resultados incorretos, falta de precisão das soluções obtidas e iterações demasiadas do algoritmo [21].

Da função de estabilidade dada por (4.2) observa-se que ao aumentar-se os valores do passo de integração  $h$  e da ponderação  $\theta$ , diminui-se a região de estabilidade do método numérico. Considerando o pior caso, isto é, tomando-se os autovalores mínimos da matriz  $\Lambda$ , computa-se a condição (4.3), sempre que houver alteração dos valores de  $h$  e  $\theta$  pelo algoritmo. É necessário garantir-se, então, que os novos valores calculados estejam sempre dentro do intervalo de estabilidade, que varia conforme equação (4.2).

Caso algum valor calculado para as variáveis manipuladas  $h$  e/ou  $\theta$  viole a condição de estabilidade (4.3), o algoritmo proposto decrementa o valor da ponderação  $\theta$  em uma pequena porção, tornando o método mais implícito, e por conseguinte aumentando o domínio de estabilidade. Este decremento é realizado até que o valor de  $\theta$  calculado satisfaça a condição (4.3) e o laço do método possa então aceitar o tamanho do passo  $h_{n+1}$  calculado.

O pseudo-código desta metodologia é apresentado pelo Algoritmo 1.

Observe que na primeira iteração de cada chamada, o preditor nada mais é do que um método de Euler simples. A atualização do gradiente (descrita na linha 24 do pseudo-código) faz com que, a menos do primeiro passo, seja necessário realizar apenas uma chamada do sistema ( $\mathbf{f}_{n+1}$ ) por iteração, contribuindo com sua eficiência.

Nota-se pelo algoritmo que os seguintes parâmetros devem ser previamente fornecidos: os incrementos ( $i_h, i_\theta$ ), e decrementos ( $d_h, d_\theta$ ) das variáveis manipuladas; o número máximo de iterações  $N_{max}$ , bem como o valor  $N_i$  (necessário para reduzir a ponderação tornar o método mais implícito); e ainda quanto aos valores iniciais do tamanho do passo  $h$  e da ponderação  $\theta$ . Isso implica ser necessário estudar a parametrização destes valores no algoritmo.

Sendo assim, primeiramente foi desenvolvido um algoritmo que permitisse uma escolha automática destes parâmetros. Fornecia-se a este uma faixa de valores para cada parâmetro e, para todas as



---

**Algoritmo 1:** Ponderação variável com controle do tamanho do passo

---

**Entrada:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n, \theta_n$   
**Dados:** tempo  $\leftarrow 0, n \leftarrow 0, \mathbf{f}_{n+1} \leftarrow \mathbf{f}_n$ ;

- 1 **enquanto** tempo  $< T_s$  **faça**
- 2      $j \leftarrow 0, e_{n+1} \leftarrow \mathcal{E}$ ;
- 3     **enquanto**  $e_{n+1} \geq \mathcal{E}$  **faça**
- 4          $h_{j+1} \leftarrow \max(h_j(1 - d_h/100), T_s/100)$ ;
- 5          $\bar{h} \leftarrow h_{j+1}\lambda_{\max}$  ; calcule  $\|R(\bar{h}, \theta_j)\|$ ;
- 6         **enquanto**  $\|R(\bar{h}, \theta_j)\| \geq I$  **faça**
- 7              $\theta_{j+1} \leftarrow \max(\theta_j(1 - d_\theta/100), 0)$ ;
- 8         **fim**
- 9         compute  $(\mathbf{x}_{n+1})_p$  e  $\mathbf{p}_{n+1}$  pelo  $\theta$ -method (2.11);
- 10         calcule  $\mathbf{f}_{n+1} = \mathbf{f}(\text{tempo} + h_{j+1}, (\mathbf{x}_{n+1})_p, \mathbf{p}_{n+1})$ ;
- 11         compute  $(\mathbf{x}_{n+1})_c$  pelo  $\theta$ -method (2.11);
- 12         calcule o erro local relativo  $e_{n+1}$  por (4.4);
- 13         **se**  $j \geq N_i$  **então**
- 14              $\theta_{j+1} \leftarrow \max(\theta_j(1 - d_\theta/100), 0)$ ;
- 15         **senão**
- 16             **fim**
- 17         **se**  $j > N_{\max}$  **então**
- 18             | fim-enquanto;
- 19         **senão**
- 20             | Retorne ao passo 4;
- 21         **fim**
- 22         **atualize:**
- 23              $h_j \leftarrow h_{j+1}, \theta_j \leftarrow \theta_{j+1}$ ;
- 24              $\mathbf{f}_n \leftarrow \mathbf{f}_{n+1}, \mathbf{p}_n \leftarrow \mathbf{p}_{n+1}$ ;
- 25              $j \leftarrow j + 1$ ;
- 26     **fim**
- 27     **se**  $j < N_i$  **então**
- 28          $h_j \leftarrow \min(h_j(1 + i_h/100), T_s - \text{tempo})$ ;
- 29          $\theta_j \leftarrow \min(\theta_j(1 + i_\theta/100), 1)$ ;
- 30     **senão**
- 31     **fim**
- 32      $\mathbf{x}_n \leftarrow (\mathbf{x}_{n+1})_c$ ;
- 33      $h_n \leftarrow h_j, \theta_n \leftarrow \theta_j$ ;
- 34     tempo  $\leftarrow \text{tempo} + h_j$ ;
- 35      $n \leftarrow n + 1$ ;
- 36 **fim**

**Saída:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n, \theta_n$

---

combinações possíveis entre esses valores, o programa simulava a integração numérica do modelo DAE com um determinado conjunto de parâmetros, retornando aquele grupo de menor custo computacional em termos de tempo de simulação.

A utilização deste algoritmo para tal tarefa é interessante, porém como este é altamente dependente do modelo, e dado que alterações nos simuladores são frequentes, especialmente na fase de testes dos algoritmos de controle, executar esse programa muitas vezes torna-se custoso. Além disso, em termos de comparação com outros métodos, sua utilização não é academicamente interessante, haja vista que os métodos concorrentes desconhecem o modelo a resolver. Logo, uma heurística foi adotada para realizar a parametrização desta metodologia, levando-se em conta que a eficiência computacional do algoritmo é um compromisso entre sua precisão numérica e sua rapidez.

O passo de integração inicial  $h$  pode ser um valor tão grande quanto seu intervalo de estabilidade, definido pela condição (4.3), o permitir, pois é assumido que o modelo encontra-se inicialmente em seu ponto de operação, mas mesmo que não estivesse, o controle de passo reduziria seu valor na primeira iteração do método. Também por partir-se o sistema do regime permanente, a ponderação  $\theta$  pode ser inicializada em um valor igual ou próximo do máximo ( $\theta \cong 1$ ), isto é, mais explícita.

Aumentar o número máximo de iterações  $N_{max}$  pode ser uma opção para reduzir o número de soluções fora da tolerância. O número  $N_i$  de passos mínimo para tornar o método mais implícito (diminuir  $\theta$ ) pode ser tomado como uma fração de  $N_{max}$ , lembrando que ao usar um valor alto, pode-se deixar o método lento ao integrar sistemas rígidos. Por fim, para endereçar os valores de escala, o quanto aumentar ( $i_h\%$  e  $i_\theta\%$ ) ou diminuir ( $d_h\%$  e  $d_\theta\%$ ) o tamanho do passo  $h$  e das ponderações  $\theta$  respectivamente, deve-se escolher valores pequenos para os incrementos, uma vez que, como já mencionado, um aumento destes parâmetros provoca uma redução no domínio de estabilidade do método. Nestes termos é necessário ser conservativo, a fim de evitar comportamentos indesejados quando o método aproxima-se dos limites de sua região de estabilidade.

### 4.3 MÉTODO CONTROLE PI DO ERRO LOCAL

Esta metodologia foi baseada no trabalho de [18] que toma como premissa a utilização dos critérios típicos de controle do tama-

nho do passo descritos na Seção 2.3.3, mas estudados sob o ponto de vista da teoria clássica de controle. Esta dissertação utiliza as mesmas ideias desenvolvidas por [18], e acrescenta melhorias com relação à dinâmica das variáveis manipuladas e controladas, a fim de contribuir com uma melhora na eficiência do cálculo do tamanho do passo  $h$  utilizado pelo algoritmo de integração numérica.

Primeiramente, sabe-se que o fornecimento de sequências suaves de tamanho do passo  $h_{n+1}$  diminuem o comportamento oscilatório da dinâmica desta variável, bem como na dinâmica do erro local [18]. Isso é particularmente importante quando o erro local encontra-se próximo de satisfazer a condição de precisão especificada, pois com menos oscilação nesta região, diminui-se o número de aproximações imprecisas, e reduz-se também a quantidade de passos rejeitados. Dessa forma, espera-se uma melhora na eficiência do algoritmo de integração.

Como motivação adicional para a aplicação de técnicas de controle linear, recorre-se ao fato de que um controle de passo típico tal como o da equação (2.44) provoque oscilações de grande magnitude no tamanho do passo calculado  $h_{n+1}$ , quando métodos explícitos puros são usados para integrar partes rígidas do modelo, ocasionando um conflito entre requisitos de precisão e de estabilidade numérica do método, como visto na seção 2.3.4. Isso faz com que o uso indevido de tamanhos de passo fora da faixa de estabilidade, seja porque o método não previu uma atitude corretiva para tal situação, ou por uma má escolha de um *solver* comercial por desconhecimento do usuário, provoque oscilações e mesmo a instabilidade numérica do método [25, 26].

Dado que, através da utilização do método ponderado, seja possível deparar-se com esta situação constantemente, é de interesse o desenvolvimento de técnicas que evitem ou minimizem esse tipo de comportamento, de maneira que a suavidade do sinal  $h$  transpareça na própria solução numérica.

Assume-se, inicialmente, que o erro local relaciona-se assintoticamente ao tamanho do passo, isto é,  $e \sim h^{p+1}$ , tal que no instante  $n$ , a próxima estimativa do erro seja dada por:

$$e_{n+1} = \kappa h_{n+1}^{p+1} \quad (4.5)$$

onde  $p$  é a ordem da formulação da aproximação, relacionada diretamente com o número de chamadas da função dinâmica  $f$  do sistema. O coeficiente  $\kappa$  depende da solução do sistema DAE abordado, cujo comportamento não linear pode ser tomado como irrele-

vante desde que se garanta a estabilidade local do do sistema DAE e do método numérico [18]. Mais detalhes podem ser estudados em [15, 14, 33].

Afim de tomar tamanhos de passo  $h_{n+1}$  tão grandes quanto possível, sem que erro local  $e_{n+1}$  viole a tolerância  $\mathcal{E}$ , o tamanho do passo deve ser escolhido tal que se satisfaça:

$$e_{n+1} \cong \mathcal{E} \quad (4.6)$$

Recorda-se aqui a estratégia de controle de passo tradicional, admitindo-se a condição do erro local atender (ficar abaixo) a tolerância (4.6):

$$h_{n+1} \leq \gamma \left( \frac{\mathcal{E}}{e_{n+1}} \right)^{1/p} \cdot h_n \quad (4.7)$$

onde  $\gamma \leq 1$  é um coeficiente destinado a evitar múltiplas mudanças no sinal, quando o erro encontra-se muito próximo da tolerância.

Para evitar pequenas mudanças no tamanho do passo, uma zona-morta é frequentemente implementada: se a razão entre tolerância e erro local ( $\mathcal{E}/e_{n+1}$ ) for muito próxima de 1, nenhuma mudança no tamanho do passo é realizada. Ainda pode-se limitar a magnitude desta razão de maneira a evitar mudanças abruptas no sinal  $h_{n+1}$  resultante.

Essa estratégia tradicional de controle de passo geralmente funciona muito bem [25]. Entretanto, como já mencionado, para o caso de uso de métodos explícitos para integração de sistemas rígidos, oscilações podem persistir e com isso muito tempo computacional ser despendido recalculando novos tamanhos de passo [18].

### 4.3.1 Análise do problema de controle

Considere a escolha do tamanho de passo como um problema de controle cuja malha pode ser vista na Figura 4.2. O controlador  $C_h$  tem a tarefa de calcular um tamanho de passo, tal que o erro local estimado  $e_{n+1}$  siga (o mais perto possível) uma referência dada por sua própria tolerância  $\mathcal{E}$ , a fim de satisfazer a relação (4.6).

A planta  $P$  consiste no modelo e no método de integração. Ela toma um tamanho de passo  $h_{n+1}$  como entrada e produz uma estimativa do erro local  $e_{n+1}$ , dada pela equação (4.4). Naturalmente, a solução numérica do sistema também é fornecida pela planta, mas esta não é usada para o projeto do controlador de passo [18].

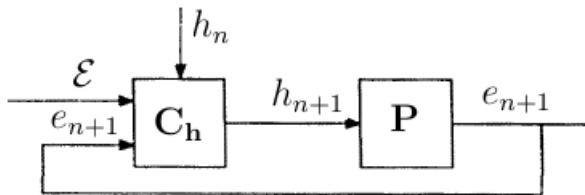


Figura 4.2: Malha de controle do erro local.

Neste trabalho, a planta  $\mathbf{P}$  é não linear e suas propriedades dependem das variações de comportamento (dinâmico) das soluções computadas do PVI (2.4). Uma parte da não linearidade, porém, pode ser tratada levando-se em conta o comportamento assintótico da relação (4.5). Ao aplicar-se o logaritmo em ambos os lados desta equação, a parte exponencial da não linearidade é transformada em uma relação afim entre o logaritmo do tamanho do passo e o logaritmo da estimativa do erro local:

$$\log(e_{n+1}) = (p + 1)\log(h_{n+1}) + \log(\kappa) \quad (4.8)$$

Considerando-se, então,  $\log(h_{n+1})$  como entrada e  $\log(e_{n+1})$  como saída da planta  $\mathbf{P}$ , e ainda assumindo-se que as variações impostas pela componente não linear  $\log(\kappa)$  (dependente do método de integração utilizado e do problema DAE a resolver), contribuam pouco nesta relação – o que é verdade, desde que as variações do tamanho do passo sejam suaves e limitadas pelo intervalo de estabilidade do método [33], então pode-se projetar um controlador linear para satisfazer a relação desejada (4.6).

A estratégia típica de controle de passo descrita pela equação (4.7) pode ser vista então como um controlador integrador, e o logaritmo de  $h_n$  a variável manipulada [18]. Aplicando o logaritmo em ambos os lados desta equação, obtém-se:

$$\log(h_{n+1}) = \log(h_n) + \frac{1}{p} \left( \log(\gamma^p \cdot \mathcal{E}) - \log(e_{n+1}) \right) \quad (4.9)$$

Observa-se que o tamanho do passo  $h_{n+1}$  irá variar enquanto o desvio, ou erro de controle,  $\log(\gamma^p \cdot \mathcal{E}) - \log(e_{n+1})$  não for nulo. Adotar um valor de segurança  $\gamma$  é equivalente a utilizar uma tolerância menor (mais apertada) para o erro local.

Levando-se em conta  $\log(\gamma^p \cdot \mathcal{E})$  como sendo o *set-point* da malha, o sinal de controle  $\log(h_{n+1})$  é obtido pela soma acumulada do sinal de erro de controle, multiplicado por um fator  $1/p$ , com o logaritmo do passo anterior  $\log(h_n)$ . Este fator, que depende da ordem do método, é também conhecido como ganho integral e determina o quão rápido o controlador responderá a variações não nulas no sinal de desvio que o multiplica.

### 4.3.2 Projeto do algoritmo de controle PI filtrado com *anti wind-up*

O controlador do tamanho do passo é discreto, pois ele enxerga a planta como um sistema amostrado pelo método numérico, que recebe uma sequência de tamanhos de passos  $h_n$  e fornece uma sequência de estimativas para o erro local  $e_{n+1}$ . Para o projeto do algoritmo de controle, deve-se considerar que existem restrições no tamanho do passo  $h_n$ , tal como os limites dados pelo intervalo de estabilidade numérica do método, e pelo próprio período de amostragem utilizado.

Portanto, caso a variável ultrapasse esses limites, ou seja, *sature*, e o valor correspondente seja diferente do valor calculado pela lei de controle, é necessário que o algoritmo de controle tome uma ação corretiva de maneira a prevenir-se de um comportamento inadequado em malha fechada.

Esta atitude preventiva, tomada quando um sinal de controle é limitado, denomina-se ação *anti wind-up* e corresponde à adição da diferença entre o sinal saturado e o sinal calculado pela lei de controle à parcela integral do controlador, tal como apresentado na equação (4.10d). Desta forma, os incrementos posteriores da ação de controle são reduzidos, tornando a resposta mais conservativa, evitando-se assim a saturação do sinal. Pode-se ainda filtrar a contribuição dessa diferença através de um ganho  $K_{AW}$ , corrigindo-se dinamicamente a ação integral quando a ação opera de maneira não linear.

A sequência de passos realizadas pelo algoritmo de controle: soma acumulada (parcela integral), cálculo da ação de controle, saturação, e adição da parcela *anti wind-up*, contribuição deste trabalho visando uma ação de controle mais suave, são descritas respectivamente pelo conjunto de equações (4.10), onde  $\rho$  é uma função que representa o conjunto de restrições de saturação do sinal de controle.

$$I_{n+1|temp} = I_n + K_I(\log(\gamma^p \cdot \mathcal{E}) - \log(e_{n+1})) \quad (4.10a)$$

$$h_{n+1|temp} = e^{(I_{n+1|temp})} \quad (4.10b)$$

$$h_{n+1} = \rho(h_{n+1|temp}) \quad (4.10c)$$

$$I_{n+1} = I_{n+1|temp} + K_{AW}(\log(h_{n+1}) - \log(h_{n+1|temp})) \quad (4.10d)$$

onde  $K_I = 1/p$  é o ganho integral e  $I_n = \log(h_n)$ . Em geral, o filtro da ação *anti wind-up* é escolhido empiricamente, tal que  $K_{AW} > K_I$ , mas com valores não muito elevados, evitando-se a ampliação de possíveis ruídos presentes na variável manipulada [31].

A despeito de integrar-se numericamente um sistema rígido por um método explícito, oscilações podem aparecer no sinal de controle  $h_{n+1}$  resultantes do fato de um controlador integrador puro possuir características de estabilidade muito fracas<sup>1</sup>, e ainda que um ganho integral de valor  $1/p$  pode ser muito elevado, em especial, quando a integração numérica é realizada por métodos de baixa ordem.

Em vista disso, [Gustafsson et al. \[18\]](#) adiciona à lei de controle (4.10) uma parcela proporcional, descrita pela equação (4.11), de maneira a generalizar o projeto para um controlador do tipo **PI**, cujo ajuste de ganhos é dependente do sistema **DAE** a integrar.

$$P_{n+1} = K_P(\log(\gamma^p \cdot \mathcal{E}) - \log(e_{n+1})) \quad (4.11a)$$

$$h_{n+1|temp} = e^{(P_{n+1} + I_{n+1|temp})} \quad (4.11b)$$

Neste trabalho, ajusta-se o ganho proporcional  $K_P$  de maneira a compensar uma diminuição do ganho integral, isto é,  $K_I < 1/p$ , com o objetivo de reduzir as oscilações. Esse ajuste é determinado heurísticamente, através da observação da resposta oscilatória no sinal de erro local  $e_{n+1}$  (de saída da malha), e da consequente melhora da eficiência computacional do algoritmo.

Embora a adição da ação proporcional colabore para amenizar da magnitude das oscilações (que são componentes de alta frequência), as contribuições da parcela não linear  $\log(\phi)$  não modeladas do sistema **DAE** podem ser significativas quando houver a aplicação de perturbações no sistema, ou seja quando este possa se afastar de seu ponto de operação. Assim sendo, neste trabalho

---

<sup>1</sup>Para maiores detalhes, consultar [31].

adiciona-se um filtro passa-baixas à saída do controlador PI de maneira a filtrar o sinal de controle  $h_{n+1}$  calculado e eliminar as componentes de mais alta frequência, a fim de minimizar as oscilações remanescentes. A saída do filtro, e portanto a ação de controle a ser enviada à planta, corresponde ao sinal  $h_{n+1|f}$  na equação (4.12) abaixo.

$$h_{n+1|f} = \alpha h_{n|f} + (1 - \alpha)h_{n+1} \quad (4.12)$$

onde a frequência de corte do filtro  $\alpha$  é ajustável e  $\alpha < 1$ .

Desta forma, neste trabalho, o projeto resultante para o controle do erro local é de um controle PI, semelhante ao desenvolvido por Gustafsson et al.[18], mas com a adição de um filtro à variável manipulada, e uma ação *anti wind-up* na parcela integral, em vista à obtenção de um algoritmo de integração com uma resposta mais suave e eficiente. A malha de controle final desenvolvida é mostrada pela Figura 4.3

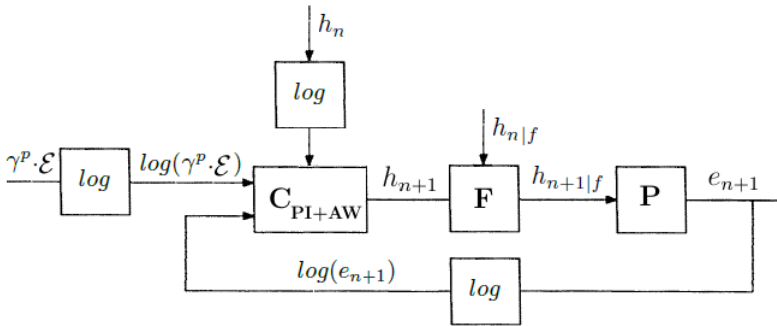


Figura 4.3: Malha de controle PI final.

O pseudo-código completo desta metodologia de integração é apresentado em Algoritmo 2.

### 4.3.3 Considerações a cerca da estabilidade do método controle PI do erro local

A estabilidade numérica do método é garantida através da adição do intervalo de estabilidade, obtido através da condição (4.3) com ponderação  $\theta$  fixa, à função de saturação  $\rho$  do algoritmo de controle, cabendo, portanto, ao próprio controlador assegurar a manutenção do tamanho do passo  $h_{n+1}$  dentro deste intervalo.

Podemos levantar que, quando utilizado um método explícito para resolver um sistema ODE rígido, a relação assintótica (4.5)



**Algoritmo 2:** Controle PI do erro local.

---

**Entrada:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n$   
**Dados:**  $tempo \leftarrow 0, n \leftarrow 0, \mathbf{f}_{n+1} \leftarrow \mathbf{f}_n, I_n \leftarrow \log(h_n),$   
 $h_{n|f} \leftarrow 0;$

- 1 **enquanto**  $tempo < T_s$  **faça**
- 2      $j \leftarrow 0, e_{n+1} \leftarrow \mathcal{E};$
- 3     **enquanto**  $e_{n+1} \geq \mathcal{E}$  **faça**
- 4         compute  $(\mathbf{x}_{n+1})_p$  e  $\mathbf{p}_{n+1}$  pelo  $\theta$ -method (2.11);
- 5         atualize  $\mathbf{f}_{n+1} = \mathbf{f}(tempo + h_j, (\mathbf{x}_{n+1})_p, \mathbf{p}_{n+1});$
- 6         compute  $(\mathbf{x}_{n+1})_c$  pelo  $\theta$ -method (2.11);
- 7         calcule o erro local relativo  $e_{n+1}$  por (4.4);
- 8         **início-controle PI**
- 9             compute a ação integral:  $I_{j+1|temp}$  com (4.10a);
- 10            compute a ação proporcional:  $P_{j+1}$  com (4.11a);
- 11            calcule a ação de controle:  $h_{j+1|temp}$  por (4.10b);
- 12            compute a ação de controle saturada:  $h_{j+1}$  com (4.10c);
- 13            calcule a correção anti wind-up:  $I_{j+1}$  por (4.10d);
- 14            compute o filtro da ação de controle:  $h_{j+1|f}$  por (4.12);
- 15            **fim-controle PI**
- 16            **atualize:**
- 17             $h_{j|f} \leftarrow h_{j+1|f}, h_j \leftarrow h_{j+1|f};$
- 18             $I_j \leftarrow I_{j+1};$
- 19             $\mathbf{f}_n \leftarrow \mathbf{f}_{n+1}, \mathbf{p}_n \leftarrow \mathbf{p}_{n+1};$
- 20             $j \leftarrow j + 1;$
- 21            **se**  $j > N_{max}$  **então**
- 22                | fim-enquanto;
- 23            **senão**
- 24                | Retorne ao passo 4;
- 25            **fim**
- 26     **fim**
- 27      $\mathbf{x}_n \leftarrow (\mathbf{x}_{n+1})_c;$
- 28      $tempo \leftarrow tempo + h_j;$
- 29      $n \leftarrow n + 1;$
- 30      $h_n \leftarrow h_{j|f};$
- 31 **fim**

**Saída:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n$

---

do erro local usada no projeto do controlador do tamanho de passo é perdida, e parcelas não modeladas do erro podem instabilizar o sistema. Entretanto, como mostrado por [18], esta relação assintótica pode ser considerada representativa quanto à parte dominante do erro local real, desde que seja garantida a estabilidade numérica do método, através da condição (4.3). Mais detalhes sobre essa asserção podem ser encontradas em [33].

#### 4.4 MÉTODO DE CONTROLE DO ERRO LOCAL POR CANCELAMENTO DE NÃO LINEARIDADE

A metodologia apresentada nesta seção também envolve a manipulação do tamanho do passo de integração, utilizando o método ponderado com  $\theta$  fixo, e baseia-se em determinar experimentalmente uma função característica não linear que descreva a dinâmica do sistema DAE a integrar, dentro de um intervalo de operação numericamente estável. Esta metodologia é, portanto, totalmente baseada e dependente do PVI considerado.

Seja uma função  $\Phi^{-1}$  não linear contínua e inversível que descreva matematicamente o tamanho do passo  $h_{n+1}$  em função de um sinal de erro local auxiliar  $\bar{e}_{n+1}$ . Pode-se utilizar essa relação em uma malha de controle, como mostra a Figura 4.4 e, por meio de um cancelamento de não-linearidades, obter-se uma relação linear entre este sinal auxiliar  $\bar{e}_{n+1}$  de erro local e a estimativa do erro local  $e_{n+1}$ . Ou seja:

$$e_{n+1} = \Phi(h_{n+1}) \quad (4.13a)$$

$$h_{n+1} = \Phi^{-1}(\bar{e}_{n+1}) \quad (4.13b)$$

Assim, o sinal  $\bar{e}_{n+1}$ , criado a partir da identificação de uma função inversa  $\Phi^{-1}$  pode ser usado como uma variável manipulada para projetar-se um controlador linear  $C$  do erro local  $e_{n+1}$ . Ou seja, manipula-se indiretamente o tamanho do passo  $h_{n+1}$  através desse sinal auxiliar  $\bar{e}_{n+1}$ .

Desta forma, admite-se como planta  $P$  a relação entre a função  $\Phi^{-1}$  e a equação do erro local descrita pela equação (4.4). O diagrama de blocos da Figura 4.4 ilustra o sistema realimentado resultante.

O controlador  $C$  pode ser projetado utilizando-se técnicas clássicas de controle linear, manipulando-se indiretamente o tamanho do passo  $h_{n+1}$  através variável auxiliar  $\bar{e}_{n+1}$ , e tomando como entrada a diferença entre a tolerância  $\mathcal{E}$ , e o erro local estimado

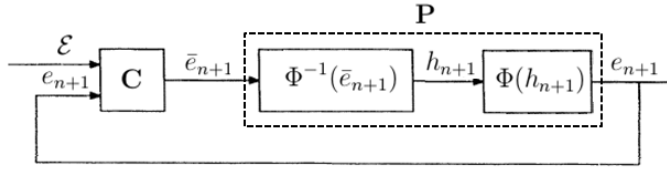


Figura 4.4: Malha de controle do erro local por cancelamento de não linearidade.

$e_{n+1}$ , variável de saída calculado pela equação (4.4). Previamente ao projeto de controle, entretanto, é necessário que se efetue a identificação de uma função não linear  $\Phi^{-1}$ , e do correspondente modelo para a planta  $P$ .

#### 4.4.1 Identificação do modelo discreto

Para identificação de um modelo discreto para a planta, considere o sistema DAE de compressão a ser integrado com o método ponderado com  $\theta = 0,9$  fixo<sup>2</sup>. Primeiramente, deve-se estabelecer seu intervalo de estabilidade.

Aplicando-se a condição de estabilidade (4.3) em sua forma escalar, obtém-se o conjunto de valores que o tamanho de passo  $h_{n+1}$  pode assumir, de maneira a garantir um comportamento numérico estável do método ponderado.

$$0 < h_{n+1} < \frac{5}{2|\lambda_{max}|} \quad (4.14)$$

Substituindo-se o autovalor de maior magnitude identificado no modelo (o pior caso),  $|\lambda_{max}| \cong 5/3$ , na equação (4.14) acima, obtém-se o seguinte intervalo:

$$0 < h_{n+1} < \frac{3}{2} \quad (4.15)$$

onde a unidade do tamanho do passo é dada em segundos.

Assim, para o levantamento da função  $\Phi^{-1}$ , toma-se um conjunto de valores de  $h_{n+1}$  que pertençam ao intervalo (4.15), e executa-se para cada valor do conjunto, o simulador do sistema de controle PNMPC, utilizando-se o método ponderado para integração do

<sup>2</sup>Esta escolha para a ponderação  $\theta$  é dependente do modelo DAE de compressão e justificada na seção 5.2 do Capítulo 5.

modelo DAE que foi empregado tanto para simular a planta como para calcular as previsões. Para cada simulação, armazenam-se os valores correspondentes de erro local calculados através da equação (4.4).

Para o conjunto de valores  $\bar{e}_{n+1}$  obtidos, obtém-se a curva que melhor descreva a relação inversa, isto é,  $(e_{n+1} \times h_{n+1})$  através do uso de algoritmos de ajuste de funções a conjuntos de pontos (*fitting*), disponíveis em softwares de análise de dados<sup>3</sup>.

A curva equivalente  $\Phi^{-1}$  é descrita pela equação (4.16) e é apresentada na Figura 4.5.

$$h_{n+1} = \Phi^{-1}(\bar{e}_{n+1}) \tag{4.16a}$$

$$\Phi^{-1} = 6,425 \cdot 10^{-9} \cdot \exp(1,88 \cdot 10^7 \bar{e}_{n+1}) \tag{4.16b}$$

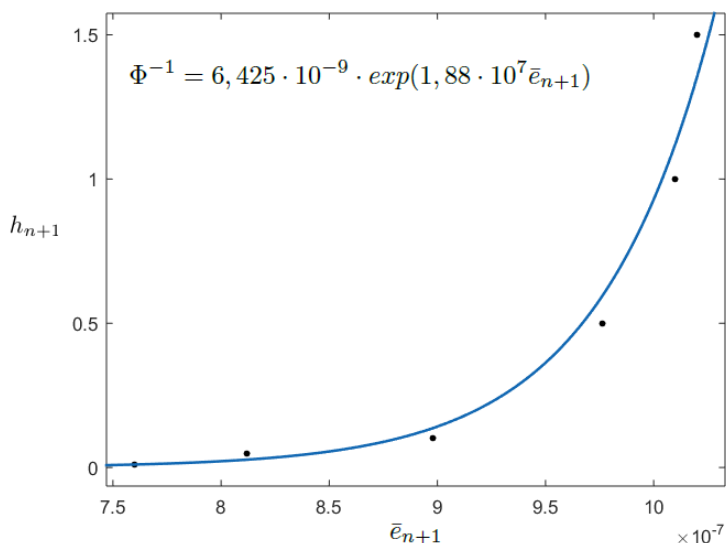


Figura 4.5: Curva  $(h_{n+1} = \Phi^{-1}(\bar{e}_{n+1}))$  relacionando o erro local ao tamanho do passo.

Observa-se que esta curva é dependente tanto do ponto de operação do sistema DAE como da fórmula utilizada para computar as aproximações da integração numérica deste sistema. A aplicação

<sup>3</sup>Tal como a ferramenta *cftool* do MATLAB 2015a<sup>®</sup>.

de perturbações de grande magnitude ou o uso de diferentes métodos numéricos alteram a curva  $\Phi$  identificada por esse processo.

Obtida uma relação linear entre erro local  $e_{n+1}$  e a variável auxiliar  $\bar{e}_{n+1}$ , deve-se agora identificar um modelo discreto para a planta **P**. Como deseja-se que o erro local siga ou seja menor que a tolerância, isto é,  $e_{n+1} \leq \mathcal{E}$ , aplica-se na planta um degrau, de valor inicial  $\bar{e}_{n+1} = 0,748 \cdot \mathcal{E}$ , e de magnitude  $\bar{e}_{n+1} = 0,25 \cdot \mathcal{E}$ , de maneira a incluir toda a faixa de valores de  $h_{n+1}$  determinada pelo intervalo de estabilidade (4.15).

Calcula-se, então, a resposta do sistema a uma entrada do tipo degrau no sinal  $bare_{n+1}$  a partir da execução do algoritmo de simulação. O resultado pode ser visto na Figura 4.6.

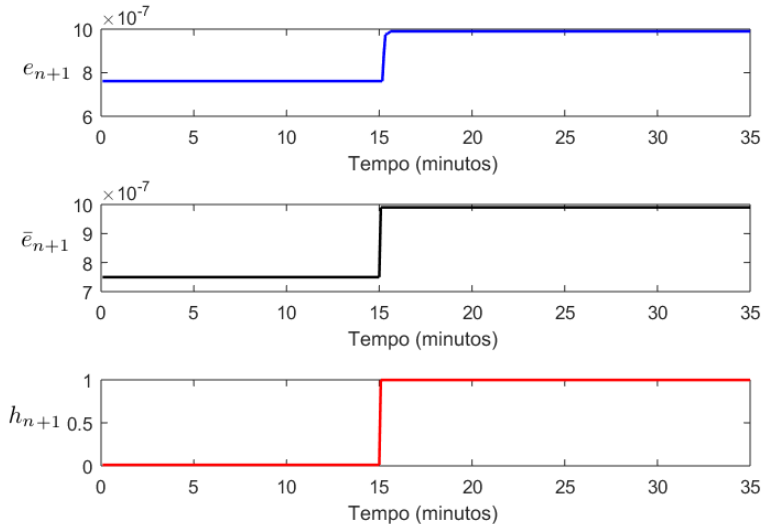


Figura 4.6: Resposta ao degrau (sinal em azul) da planta **P** à entrada  $\bar{e}_{n+1}$  (em preto) e o correspondente passo de integração  $h_{n+1}$  (em vermelho)

A partir da resposta dinâmica obtida, classifica-se a planta **P** como um sistema de primeira ordem, através da técnica de identificação por resposta ao degrau<sup>4</sup>. Com o auxílio da transformada  $\mathcal{Z}$ , é obtido o seguinte modelo discreto:

<sup>4</sup>Maiores detalhes desta técnica podem ser assimilados em [34].

$$\frac{e(z)}{\bar{e}(z)} = \tilde{P}(z) = \frac{0.3615}{z - 0.6211} \quad z = e^{-t/\tau_{MA}} \quad (4.17)$$

onde  $\tau_{MA}$  é a constante de tempo em malha aberta de  $\mathbf{P}$  (que corresponde ao intervalo em que 63,2 % do valor de regime permanente da resposta é alcançado), identificada na Figura 4.6 como sendo aproximadamente 15 segundos, e  $t = T_s = 5$  segundos.

Sobrepondo-se a resposta ao degrau do modelo representado por (4.17) com a resposta ao degrau do sistema  $\mathbf{P}$  na Figura 4.7, é possível notar-se a similaridade das dinâmicas das respostas.

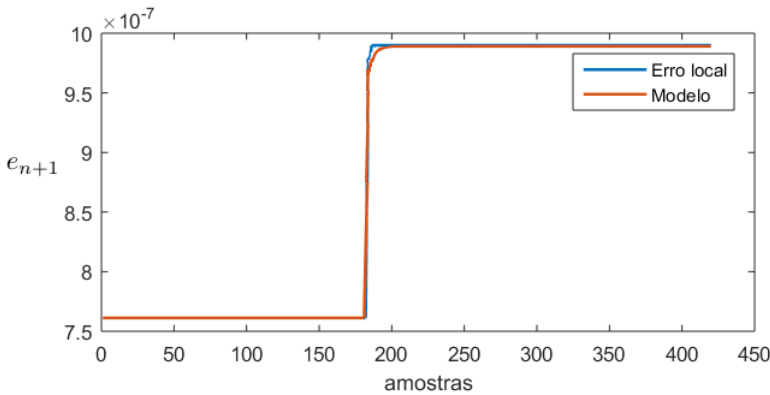


Figura 4.7: Sobreposição da resposta temporal de  $\mathbf{P}$  (em azul) e o correspondente modelo identificado (em vermelho).

Considerando-se o modelo satisfatório segue-se, então, para o projeto do controlador.

#### 4.4.2 Projeto e sintonia do controlador

Para projeto do controlador, adota-se neste trabalho (dentre as diversas técnicas de projeto de controle linear existentes na literatura) a técnica de Controle por Modelo Interno – em inglês **Internal Model Control (IMC)**, desenvolvida por [35], que toma o modelo do sistema  $\tilde{P}$  como parte integrante do controlador. Esta escolha deve-se a dois motivos:

- O projeto leva em consideração as incertezas do modelo e a presença de perturbações não medidas;

- Permite-se contrabalancear a performance e a robustez do sistema de controle com as variações do processo e os erros de modelagem.

Considere que o modelo identificado do sistema **DAE** a integrar  $\tilde{P}$  contenha incertezas (erros de modelagem), e que estas somadas às perturbações não medidas do sistema sejam representadas pela variável  $\tilde{d}_{n+1}$  da Figura 4.8. Dado que a variável manipulada  $\tilde{e}_{n+1}$  aplicada ao modelo  $\tilde{P}$  produza uma resposta  $\tilde{e}_{n+1}$ , pode-se subtrair a resposta da planta  $P$ , isto é, a estimativa do erro local  $e_{n+1}$  dada pela equação (4.4) somadas às incertezas e perturbações  $d_{n+1}$ , para determinar a diferença, ou o desvio com relação ao modelo  $\tilde{P}$ . Esta informação, denominada  $\tilde{d}_{n+1}$ , pode ser então realimentada ao controlador para se compensar os desvios correspondentes aos erros de modelagem e às perturbações não medidas no sistema. O procedimento de projeto consiste nas seguintes etapas [35]:

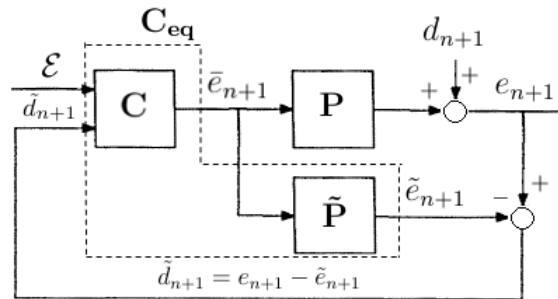


Figura 4.8: Malha de controle **IMC** de erro local.

**Etapla 1:** Fatora-se o modelo  $\tilde{P}(z)$  em duas partes: uma que possa ser invertida:  $\tilde{P}_-(z)$ ; e outra que não possa ser invertida:  $\tilde{P}_+(z)$  e que contenha quaisquer zeros de fase não mínima ou atrasos de transporte. Como estes termos não se apresentam no modelo identificado através da relação (4.17), obtém-se:

$$\tilde{P}(z) = \tilde{P}_+(z) \cdot \tilde{P}_-(z) \quad (4.18a)$$

$$\tilde{P}(z) = 1 \cdot \frac{0.3615}{z - 0.6211} \quad (4.18b)$$

**Etapla 2:** Relaciona-se o controlador **IMC** pelo inverso de  $\tilde{P}_-(z)$ ,

garantindo-se assim que  $C$  não contenha elementos que o tornariam instáveis ou dificultariam o projeto. Portanto:

$$C(z) = \frac{1}{\tilde{P}_-(z)} \quad (4.19)$$

**Etapa 3:** Para que o controlador seja próprio e fisicamente realizável<sup>5</sup>, adiciona-se um filtro  $G_F(z)$  ao controlador, tal que a alocação da constante  $z_f$  determine a velocidade da resposta do sistema em malha fechada:

$$C(z) = \frac{1}{\tilde{P}_-(z)} \cdot G_F(z) \quad (4.20a)$$

$$C(z) = \frac{(z - 0.6211)}{0.3615} \cdot \frac{(1 - z_f)}{z - z_f} \quad (4.20b)$$

**Etapa 4:** Determina-se o controlador equivalente, oriundo da álgebra das malhas que contêm o controlador  $C$  e o modelo  $\tilde{P}$  da Figura 4.8. Neste trabalho projetou-se  $z_f$  de maneira que a dinâmica de malha fechada fosse aproximadamente duas vezes mais rápida que a de malha a aberta, isto é  $\tau_{MF} = \tau_{MA}/2$ , e  $z_f = e^{-T_s/\tau_{MF}} \cong 0.51$ . Esta dinâmica foi utilizada pois, apesar de não existirem restrições quanto a um projeto de controle mais veloz (embora mais agressivo), não foi observado ganho significativo da aplicação de dinâmicas mais rápidas que esta, quanto à eficiência computacional do algoritmo de integração.

$$C_{eq}(z) = \frac{C(z)}{1 - C(z)\tilde{P}(z)} \quad (4.21a)$$

$$C_{eq}(z) = \frac{(1 - z_f)}{0.3615} \cdot \frac{(z - 0.6211)}{z - 1} \quad (4.21b)$$

Observa-se que o ganho do controlador é inversamente proporcional à constante  $z_f$ , sendo que quanto maior a velocidade da malha, isto é, menor o valor de  $z_f$ , maior o valor do ganho do controlador. Além disso, nota-se que, dispondo de um modelo para a planta  $P$ , o único parâmetro de controle a se ajustar é justamente a constante de tempo  $z_f$  do filtro, o que faz o projeto de controle IMC ser bastante simples. Por fim, convém notar que o controlador equivalente  $C_{eq}$  nada mais é do que um controlador PI [35].

<sup>5</sup>Uma função de transferência  $G(z) = num(z)/den(z)$  é dita própria se a ordem do polinômio  $p$  do denominador for maior ou igual à do numerador, ou  $p(den(z)) \geq p(num(z))$ .



### 4.4.3 Análise da estabilidade do controlador IMC de erro local

A fim de estudar a estabilidade do sistema de controle, calcula-se a resposta do sistema realimentado em malha fechada esboçado na Figura 4.5, isto é, a saída dada pelo erro local  $e_{n+1}$  em função da entrada  $\mathcal{E}$  de referência, e das incertezas e perturbações não medidas, representadas por  $d_{n+1}$ . Assim, obtém-se

$$e(z) = \frac{C(z)P(z)}{1+C(z)(P(z)-\tilde{P}(z))} \cdot \mathcal{E} + \frac{1-\tilde{P}(z)C(z)}{1+C(z)(P(z)-\tilde{P}(z))} \cdot d(z) \quad (4.22)$$

Considera-se para esta análise que o modelo seja fiel ou muito parecido à planta, isto é,  $(P(z) - \tilde{P}(z) \cong 0)$ , o que pode ser justificado pelas as similaridades entre as respostas observadas na Figura 4.7. A partir desta premissa, e baseando-se na resposta do sistema (4.22), apresenta-se o teorema da estabilidade interna do modelo, tal como estabelecido em [35]:

- Dado um modelo perfeito  $P(z) = \tilde{P}(z)$ , o sistema de controle em malha fechada é internamente estável se e somente se  $P(z)$  e  $C(z)$  são ambos estáveis.
- Assumindo-se que  $P(z)$  é estável e que  $P(z) = \tilde{P}(z)$ , então o sistema realimentado da Figura 4.8 é internamente estável se e somente se  $C(z)$  é estável.

Dado que nesta tese considera-se a planta **P**, isto é, o sistema **DAE** em malha fechada juntamente com seu método numérico, estável (considerando que as não-linearidades sejam compensadas estaticamente), então para que sistema de controle **IMC** seja internamente estável em malha fechada, basta que o controlador  $C(z)$  da equação (4.20) tenha polos contidos no interior do círculo unitário, ou:

$$|z_f| < 1 \quad (4.23)$$

Como projetou-se  $z_f = 0.51$ , a estabilidade interna do sistema de controle é garantida.

Quanto à estabilidade numérica do método, esta é garantida limitando-se os valores do tamanho de passo aos compreendidos dentro de seu intervalo de estabilidade (4.15), bem como feito na metodologia 4.3 desenvolvida.

O pseudo-código desta metodologia é apresentado no Algoritmo 3. As equações envolvendo funções de transferência discreta foram descritas como equações a diferenças.

---

**Algoritmo 3:** Controle PI do erro local por cancelamento de não linearidade.

---

**Entrada:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n$   
**Dados:**  $tempo \leftarrow 0, n \leftarrow 0, \mathbf{f}_{n+1} \leftarrow \mathbf{f}_n,$   
 $\tilde{e}_n \leftarrow 0, \bar{e}_n \leftarrow 0, \bar{e}_{n+1} \leftarrow 0, e_{n|c} \leftarrow 0;$

- 1 **enquanto**  $tempo < T_s$  **faça**
- 2      $j \leftarrow 0, e_{j+1} \leftarrow \mathcal{E};$
- 3     **enquanto**  $e_{n+1} \geq \mathcal{E}$  **faça**
- 4         compute  $(\mathbf{x}_{n+1})_p$  e  $\mathbf{p}_{n+1}$  pelo  $\theta$ -method (2.11);
- 5         atualize  $\mathbf{f}_{n+1} = \mathbf{f}(tempo + h_j, (\mathbf{x}_{n+1})_p, \mathbf{p}_{n+1});$
- 6         compute  $(\mathbf{x}_{n+1})_c$  pelo  $\theta$ -method (2.11);
- 7         calcule o erro local (saída de  $\mathbf{P}$ ):  $e_{j+1}$  por (4.4);
- 8         **início-controle**
- 9             calcule a resposta do modelo  $\tilde{P}(z):$   
 $\tilde{e}_{j+1} = 0,6211\tilde{e}_j + 0,3615\bar{e}_{j+1};$
- 10            compute o desvio:  $\tilde{d}_{j+1} = e_{j+1} - \tilde{e}_{j+1};$
- 11            calcule o erro de controle:  $e_{j+1|c} = \mathcal{E} - \tilde{d}_{j+1};$
- 12            calcule a ação de controle:  
 $\bar{e}_{j+1} = 0,51\bar{e}_j + 1,36e_{j+1|c} - 0,87e_{j|c};$
- 13            calcule o tamanho de passo pela função  $\Phi^{-1}:$   
 $h_{j+1} = 6,425 \cdot 10^{-9} \exp(1,88 \cdot 10^7 \bar{e}_{j+1});$
- 14            **fim-controle**
- 15            **atualize:**
- 16             $\bar{e}_j \leftarrow \bar{e}_{j+1}, \tilde{e}_j \leftarrow \tilde{e}_{j+1}, e_{j|c} \leftarrow e_{j+1|c};$
- 17             $h_j \leftarrow h_{j+1}, \mathbf{f}_n \leftarrow \mathbf{f}_{n+1}, \mathbf{p}_n \leftarrow \mathbf{p}_{n+1};$
- 18             $j \leftarrow j + 1;$
- 19            **se**  $j > N_{max}$  **então**
- 20                 | fim-enquanto;
- 21            **senão**
- 22                 | retorne ao passo 4;
- 23            **fim**
- 24     **fim**
- 25      $\mathbf{x}_n \leftarrow (\mathbf{x}_{n+1})_c;$
- 26      $h_n \leftarrow h_j;$
- 27      $tempo \leftarrow tempo + h_j;$
- 28      $n \leftarrow n + 1;$
- 29 **fim**

**Saída:**  $\mathbf{x}_n, \mathbf{f}_n, \mathbf{p}_n, h_n$

---

## 4.5 COMENTÁRIOS FINAIS

Neste capítulo foram apresentados três algoritmos customizados a contribuir para a resolução eficiente o sistema **DAEs** de compressão complexo complexo descrito no **Capítulo 3**, mesmo sem considerar recursos de processamento computacional adicionais. Os métodos focaram especialmente na manipulação do tamanho do passo de integração a fim de que o erro local computado em cada passo seguisse ou ficasse abaixo da tolerância especificada. No próximo capítulo serão expostos os resultados da aplicação destas metodologias, comparando-as com *solvers* comerciais típicos, descritos no **Capítulo 2**, através de indicadores de eficiência numérica e computacional.



## 5 RESULTADOS E DISCUSSÕES

Este capítulo traz os resultados da aplicação das metodologias de integração numérica desenvolvidas à simulação do sistema de controle avançado da estação de compressão. Serão apresentados os cenários de simulação, bem como os parâmetros utilizados em cada método. Os resultados obtidos são comparados com metodologias típicas de resolução de sistemas DAEs quanto à robustez, precisão e eficácia da solução fornecida, a partir de indicadores de eficiência computacional.

### 5.1 CENÁRIOS DE SIMULAÇÃO

A aplicação das metodologias de integração foi realizada utilizando-se de um simulador de sistema de controle PNMPC, que faz uso de um modelo DAE da estação de compressão para calcular as previsões. Neste trabalho são considerados dois cenários: no primeiro, além de ser usado como modelo de previsão, o sistema DAE é também empregado para simular a planta. Desta forma é contemplada a fase de desenvolvimento e testes de algoritmos de controle, quando informações sobre a planta ainda não estão disponíveis, necessitando-se um modelo do processo para representá-la, e que precisa também ser resolvido por um *solver* de integração de maneira eficiente, a fim de que não seja tomado muito tempo com testes e avaliações de sintonias do controlador preditivo.

No segundo cenário, utiliza-se o sistema DAE apenas como modelo de previsão, portanto não se considera para uma análise de eficiência, a carga computacional despendida com a obtenção da resposta da planta. Neste cenário, objetiva-se avaliar o desempenho dos algoritmos desenvolvidos quando da aplicação em tempo real, onde o controlador PNMPC deve calcular as previsões e fornecer uma ação de controle para a ser aplicada no processo eficientemente dentro do período de amostragem considerado.

Para avaliar a sintonia dos algoritmos de integração numérica desenvolvidos, perturbações são inseridas na vazão de entrada de gás a fim de simular, em instantes de tempo distintos, uma queda de 50 % e um aumento de 15 % da vazão nominal. O objetivo de modificar a vazão recebida pela estação é, além de simular um evento abrupto capaz de ocorrer em uma operação real, avaliar a qualidade e a rapidez das soluções fornecidas pelos *solvers* quando mudanças na dinâmica do sistema são impostas.

O arquivo de simulação principal bem como todos os algo-

ritmos desenvolvidos foram escritos sobre a plataforma MATLAB 2015a<sup>®</sup>. Além de ser um software muito comum para estudo e análise de sistemas de controle, este fornece um pacote que disponibiliza um conjunto de *solvers* para integração numérica de sistemas DAEs, além de ferramentas para análise quantitativa e estatística em tempo real da simulação dos algoritmos desenvolvidos, tal como a quantidade de passos tomadas, o número de soluções entregues sem a precisão especificada, a quantidade de chamadas do sistema para aproximar as derivadas, e para calcular as variáveis algébricas.

Ademais, o número de chamadas do *solver* de otimização utilizado para calcular ações de controle, além do tempo computacional total decorrido. Todas as simulações foram realizadas em uma máquina com processador de quatro núcleos e 4 GB de memória RAM.

O intervalo de simulação utilizado foi de 35 minutos, e o período de amostragem de  $T_s = 5$  segundos. Este ciclo de amostragem corresponde ao montante de tempo necessário para computar as tarefas do algoritmo do sistema de controle avançado, tais como cálculo de predições, da resposta do sistema, da ação de controle, e todas as demais. Quanto à sintonia do controlador PNMPC, admitiu-se um horizonte de predição de 15 amostras, ou seja, para cada ciclo de amostragem da simulação, o controlador deve calcular a resposta do sistema DAE quinze vezes a frente da amostra atual.

Para cada cenário foram adotadas duas especificações de precisão para o erro local: primeiramente foi utilizado o valor padrão na maioria dos pacotes de *solvers* comerciais, estabelecida a  $\mathcal{E}_1 = 10^{-3}$ . Ou seja, para aceitar-se a solução calculada, o erro local relativo deve ser igual ou inferior a 0,1 %.

Para testar a eficiência dos *solvers* para tolerâncias mais apertadas, e levando-se em conta que as respostas obtidas dos modelos DAEs para cálculo de ações de controle devem apresentar alta confiabilidade em tempo real, especifica-se uma tolerância máxima de  $\mathcal{E}_2 = 10^{-6}$ . Este valor corresponde, portanto, a um milésimo do estabelecido como padrão em pacotes de *solvers* comerciais, tais como os encontrados na plataforma MATLAB 2015a<sup>®</sup>. Portanto, aceitam-se somente soluções com alta precisão, com um erro local relativo não maior que 0,0001 %. A descrição completa de todos os parâmetros de simulação encontram-se disponíveis no [Apêndice B](#).

### 5.1.1 Métodos utilizados para comparação e análise

Os métodos de integração utilizados para comparação com os desenvolvidos neste trabalho foram obtidos do pacote *ode*, ferramenta do próprio software de simulação, que disponibiliza sete *solvers*, quatro dos quais são indicados para resolução de sistemas do tipo rígido (*stiff*).

A fim de verificar a baixa eficiência e precisão de métodos puramente explícitos na integração de um problema não-rígido, também é simulada a integração com estes métodos do pacote *ode*. Adicionalmente, um método de Colocação é testado, cujo algoritmo foi implementado tal como descrito por [24], o qual faz uso de um polinômio de Lagrange de terceira ordem, e cujos pontos de colocação são obtidos por quadratura gaussiana, através da estratégia LGR (Legendre-Gauss-Radau)<sup>1</sup>. Por fim, o *solver* implementado por [18], utilizando como fórmula o método ponderado (2.11), cujo controle do erro local é adaptada e melhorada neste trabalho para a integração do sistema DAE de compressão, descrito pela Seção 4.3, também foi implementado para comparação.

Os *solvers* utilizados estão listados na Tabela 5.1. Todos eles utilizam algum tipo de controle do tamanho de passo em seus algoritmos. Os números após o identificador “*ode*” ou “*coloc*” expressam a ordem mínima e máxima de cada método, que varia a depender da distância que a solução computada esteja da precisão especificada, mas cujo critério não é especificado.

Tabela 5.1: Métodos numéricos utilizados para comparação.

Solver	Método	Tipo	Aplicação
ode45	Runge-Kutta	Passo Único Explícito	Não-rígido
ode23	Runge-Kutta	Passo Único Explícito	Não-rígido
ode113	Adams Bashforth-Moulton	Múltiplos Passos Semi-Implicito	Não-rígido
ode23t	Regra do Trapézio	Passo Único Semi-Implicito	Rígido
ode23tb	Regra do Trapézio e método BDF	Múltiplos Passos Semi-Implicito	Rígido
ode23s	Rosenbrock Modificado	Passo Único Implícito	Rígido
ode15s	Backward Differentiation Formulas	Múltiplos Passos Implícito	Rígido
coloc34	Collocation Method	Passo Único Implícito	Rígido
gPondPI	PI Step-Size Control [18]	Passo Único Semi-Implicito	Rígido

A última coluna da tabela, “Aplicação”, refere-se a uma sugestão quanto ao tipo de problema que o *solver* está apto a resolver.

<sup>1</sup>O software MATLAB 2015a<sup>®</sup> disponibiliza uma função para o cálculo dos pontos de colocação por este método.

Em relação ao pacote *ode*, esta informação foi obtida da própria documentação do pacote. Para maiores detalhes, consultar [36].

O método *ode23s* utiliza uma fórmula de Rosenbrock modificada. Este método nada mais é do que um método de Runge-Kutta implícito o qual, em vez de realizar chamadas da função dinâmica  $f$  do sistema, estima um jacobiano a cada passo para computar as aproximações das derivadas, necessitando para tal tarefa calcular inversas de matrizes, operação computacionalmente custosa [37].

### 5.1.2 Parametrização dos métodos numéricos desenvolvidos

Conforme visto no [Capítulo 4](#), alguns parâmetros devem ser informados aos algoritmos de integração tanto para a configuração destes, como para inicialização de algumas variáveis. Alguns destes parâmetros são comuns em todos os algoritmos. São eles: o número máximo de iterações  $N_{max}$  para computar uma solução, mantido em  $N_{max} = 15$  iterações; e o tamanho de passo  $h$  inicial, definido como um sexto do limite superior do intervalo de estabilidade do  $\theta$ -method, no pior caso ( $\theta = 1$ ), isto é,  $h = 2/(6|\lambda_{max}|) = 0,2$  segundos.

O número de iterações adotado é um valor médio observado de simulações com todos os métodos, tal que estes apresentassem o mínimo de soluções sem a precisão requerida. Já o tamanho de passo inicial foi escolhido por apresentar um comportamento menos oscilatório (mais conservativo) nas primeiras amostras.

Quanto à ineficiência de utilizar um tamanho de passo tão menor que o período de amostragem, recorda-se que seu valor inicial não contribui muito com eficiência do algoritmo, já que o controle de erro local projetado nos algoritmos, o manipulará de maneira a fazer com que o erro local busque rapidamente a referência especificada.

Os parâmetros utilizados nos algoritmos dos métodos estão descritos nas Tabelas 5.2, 5.3 e 5.4 respectivamente. A escolha destes foi realizada conforme exposto também no [Capítulo 4](#).

### 5.1.3 Indicadores de desempenho

As seguintes métricas de desempenho são utilizadas para avaliar e comparar a eficiência computacional de cada método:

- **Chamadas do Sistema:** a quantidade média de chamadas do modelo, isto é, o número de cálculos das funções  $f$  e  $g$  por ciclo;



Tabela 5.2: Parametrização método ponderação variável.

Parâmetro	Descrição	Valor
$\theta_{ini}$	Ponderação inicial	1
$d_h\%$	Decremento de $h_n$	25 %
$i_h\%$	Incremento de $h_n$	10 %
$d_\theta\%$	Decremento de $\theta_n$	10 %
$i_\theta\%$	Incremento de $\theta_n$	5 %
$N_i$	Mínimo de passos para decrementar $\theta_n$	5

Tabela 5.3: Parametrização método controle PI do erro local.

Parâmetro	Descrição	Valor
$\gamma$	Escalamento da referência $\mathcal{E}$	1
$K_P$	Ganho proporcional	0,75
$K_I$	Ganho integral	0,25
$K_{AW}$	Ganho do filtro <i>anti wind-up</i>	1,00
$\alpha$	Dinâmica do filtro de $h_n$	0,95

Tabela 5.4: Parametrização método controle do erro local por cancelamento de não linearidade.

Parâmetro	Descrição	Valor
$z_f$	Dinâmica de malha fechada	0,51

- **Quantidade de Passos:** o número de passos médio por ciclo de amostragem para computar uma solução;
- **Número de Imprecisões:** o valor total percentual de soluções entregues fora da precisão especificada (quando  $j > N_{max}$ );
- **Contribuição Temporal da Integração:** a carga temporal despendida com a tarefa de integração ( $t_{INT}$ ) em relação ao tempo tomado para simular um ciclo de amostragem ( $t_{CICLO}$ ) completo. Ou seja, para um número  $N_c$  de ciclos de amostragem ( $T_s$ ), este indicador corresponde, em termos percentuais a:

$$\frac{1}{N_c} \sum_{i=1}^{N_c} \frac{t_{INT}(i)}{t_{CICLO}(i)} \times 100 \quad (5.1)$$

## 5.2 RESULTADOS DE SIMULAÇÃO

### 5.2.1 Primeiro Cenário

No primeiro cenário admite-se o sistema DAE para simular a planta e calcular as predições.

As Tabelas 5.5 e 5.6 apresentam os resultados de simulação para as tolerâncias  $\mathcal{E}_1$  e  $\mathcal{E}_2$ , respectivamente. Na coluna “DAE Solver”, os algoritmos desenvolvidos seguem as seguintes abreviaturas: método Ponderação Variável com controle do tamanho de passo (mPV), método controle PI do erro local (mPI) e método controle por cancelamento de Não Linearidade (mNL).

Tabela 5.5: Cenário 1: resultados de simulação para uma tolerância de  $\mathcal{E}_1 = 10^{-3}$ .

DAE Solver	Chamadas do Sistema	Quantidade de Passos	Número de Imprecisões (%)	Contribuição Temporal da Integração (% $T_s$ )
ode45	23	13	15,82	96,12
ode23	20	10	8,56	92,80
ode113	22	13	10,12	95,62
ode23t	19	12	1,42	85,56
ode23tb	19	14	1,50	85,72
ode23s	28	10	0,00	96,74
ode15s	18	11	1,13	84,31
coloc34	19	12	0,95	85,27
gPondPI	17	10	2,52	85,30
mPV	16	11	0,00	84,40
mPI	11	10	0,00	82,35
mNL	12	9	0,00	82,48

Tabela 5.6: Cenário 1: resultados de simulação para uma tolerância de  $\mathcal{E}_2 = 10^{-6}$ .

DAE Solver	Chamadas do Sistema	Quantidade de Passos	Número de Imprecisões (%)	Contribuição Temporal da Integração (% $T_s$ )
ode45	36	23	17,92	97,87
ode23	32	20	9,09	96,95
ode113	34	23	12,05	97,73
ode23t	30	22	3,42	95,56
ode23tb	30	24	3,54	95,78
ode23s	44	16	3,65	96,91
ode15s	29	21	2,91	95,41
coloc34	30	23	3,96	95,54
gPondPI	27	10	3,52	95,30
mPV	24	18	0,00	92,40
mPI	16	15	0,95	88,35
mNL	17	14	1,12	89,48

Chamadas do sistema rígido de compressão são muito custosas. Como esperado, os solvers explícitos puros (os três primeiros

das Tabelas 5.5 e 5.6) obtiveram um desempenho muito insatisfatório, tanto pela alta quantidade de chamadas do sistema, como pelo alto número de soluções entregues fora da precisão. Já os métodos semi-implícitos testados do pacote *ode* obtiveram melhor desempenho por conseguirem, com esta estrutura, reduzir o número de chamadas do sistema, mesmo que o melhor destes (o *ode15s*) ainda tenha tomado mais de 85 % do tempo de um ciclo e entregue pelo menos 1% de soluções fora da faixa de precisão. O *solver* desenvolvido por [18] apresentou um desempenho quase que semelhante ao o *ode15s*, embora sempre com mais imprecisões, devido a presença oscilações no tamanho do passo. O *solver* *ode23s* apresentou a melhor precisão nas estimativas dentre os tradicionais testados. Para este método, entretanto, foi necessário fornecer uma estimativa inicial do jacobiano do sistema, que é atualizada a cada passo, tomando muito tempo de processamento.

Neste cenário, nota-se alta carga computacional da integração para ambas as tolerâncias. O método *mPI* apresentou o melhor resultado, mas com pelo menos 82 % do tempo de ciclo despendido com a tarefa de integração, ainda que a especificação  $\mathcal{E}_1$  fosse 1000 vezes maior que  $\mathcal{E}_2$ . Todavia, foi obtida uma quantidade de imprecisões muito maior para  $\mathcal{E}_2$ , já que os métodos precisaram realizar muito mais chamadas e passos para cumprir com esta especificação, ultrapassando muitas vezes o número máximo de passos permitido nos algoritmos numéricos. Observa-se que, para cumprir com  $\mathcal{E}_1$ , todos os métodos precisaram realizar um número consideravelmente menor de iterações a cada passo, do que para satisfazer com  $\mathcal{E}_2$ , e que para  $\mathcal{E}_1$ , especificação padrão dos *solvers* comerciais, não foi obtido um ganho de performance tão relevante em relação à integração, sendo este menos de 3 % comparando-se os métodos (comercial e desenvolvido) de melhor desempenho, isto é, *mPI* com *ode15s*. Os algoritmos implementados tomaram, em geral, menos passos por ciclo que qualquer outro, calculando muito menos vezes as funções *f* e *g* do sistema *DAE*, o que contribuiu com a redução em pelo menos 2 % a contribuição temporal da integração por ciclo de amostragem, avaliando-se os resultados de ambas as especificações em sua totalidade. Controlar o tamanho do passo de integração e utilizar um método de baixa ordem mostrou-se uma maneira efetiva de diminuir o número de chamadas do sistema para calcular cada solução, mantendo o erro local próximo à tolerância de referência, e reduzindo assim as imprecisões observadas.

## 5.2.2 Segundo Cenário

No segundo cenário admite-se o sistema DAE apenas para calcular as predições, ou seja, este é usado como modelo de predição, com o objetivo de avaliar o desempenho dos *solvers* quando embarcados em um controlador PNMPC para a atuação em um processo físico em tempo real.

As Tabelas 5.7 e 5.8 apresentam os resultados de simulação para as tolerâncias  $\mathcal{E}_1$  e  $\mathcal{E}_2$ , respectivamente.

Tabela 5.7: Cenário 2: Resultados de simulação para uma tolerância de  $\mathcal{E}_1 = 10^{-3}$ .

DAE Solver	Chamadas do Sistema	Quantidade de Passos	Número de Imprecisões (%)	Contribuição Temporal da Integração (% $T_s$ )
ode45	17	12	9,82	82,73
ode23	16	10	4,56	79,71
ode113	18	12	8,12	83,88
ode23t	12	9	0,36	75,66
ode23tb	13	10	0,53	75,28
ode23s	22	4	0,00	86,87
ode15s	14	12	0,00	72,73
coloc34	14	10	1,57	73,45
gPondPI	15	7	1,22	73,22
mPV	8	10	0,00	71,42
mPI	5	4	0,00	69,33
mNL	7	4	0,00	70,42

Tabela 5.8: Cenário 2: resultados de simulação para uma tolerância de  $\mathcal{E}_2 = 10^{-6}$ .

DAE Solver	Chamadas do Sistema	Quantidade de Passos	Número de Imprecisões (%)	Contribuição Temporal da Integração (% $T_s$ )
ode45	24	23	13,92	86,83
ode23	21	20	7,09	84,63
ode113	25	23	11,05	85,10
ode23t	20	22	2,36	83,70
ode23tb	20	24	2,75	83,85
ode23s	29	16	2,35	89,70
ode15s	19	21	2,31	83,60
coloc34	21	23	2,16	83,69
gPondPI	19	10	2,72	85,53
mPV	17	20	0,00	81,60
mPI	11	10	0,95	76,90
mNL	12	8	1,12	78,06

Neste segundo cenário, observa-se diferença quanto à contribuição da integração, tendo-se reduzindo em média 10 % o tempo por ciclo quando comparado ao cenário 1. De fato, e como esperado, todos os índices reduziram em magnitude, já que foi necessário resolver um PVI a menos por ciclo, ou seja, uma chamada

a menos dos *solvers* de integração. Novamente, os três algoritmos implementados obtiveram resultados melhores do que seus concorrentes. Quanto à tolerância  $\mathcal{E}_2$ , o ganho em eficiência é mais significativo, reduzindo-se em quase 7% a carga computacional da tarefa de integração. Somado a um *solver* de otimização eficiente, isto se traduz em um algoritmo de controle com melhor desempenho, visto que com menos tempo do período de amostragem sendo utilizado pelo controlador, a parcela restante pode ser utilizada para incrementar a robustez do controle, por exemplo com ações que garantam sua estabilidade, ou ainda reduzir o período de amostragem do algoritmo, tornando assim a chamada do controlador e o cálculo de suas tarefas mais veloz.

Quanto ao desempenho geral de cada uma das três metodologias desenvolvidas neste trabalho, para ambos os cenários: o algoritmo **mPV** apresenta um controle simples (on/off) do tamanho de passo  $h$ , e o  $\theta$  é único com controle da ponderação  $\theta$  (da direção das soluções). Isso porque, como pode-se observar na Figura 5.1, a magnitude de variação obtida com relação à ponderação foi muito pequena, mantendo-se próximo a 1, apresentando mudanças basicamente quando as amplitudes de variação do tamanho do passo foram maiores, para manter o método numericamente estável. Desta forma, na implementação das outras metodologias, adotou-se um valor constante de  $\theta = 0,9$ , podendo-se usar passos maiores do que os obtidos com o método **mPV**. Observa-se também, que esta metodologia manteve o erro local sempre abaixo da especificação, ou seja, nunca superou o número máximo de passos ( $N_{max}$ ) mesmo sob efeito de perturbações no sistema.

Por outro lado, esta metodologia apresentou muita oscilação no tamanho do passo e por conseguinte no erro local. Entretanto, isto era esperado já que nenhuma ação para mitigar tal comportamento foi implementada, com o propósito de manter o algoritmo matematicamente simples, com poucos algebrismos. Enfatiza-se que, independentemente da especificação de precisão, não existem garantias que estas oscilações não transpareçam nas soluções, podendo provocar pequenos ruídos, especialmente na ocorrência de variações bruscas das dinâmicas dos estados. Esta metodologia foi, dentre as desenvolvidas, a que mais realizou chamadas do sistema, e a que, em média, realizou mais passos para computar cada solução.

O método **mPI** alcançou o melhor desempenho entre as metodologias desenvolvidas, por realizar o menor número de chamadas do sistema, chegando a reduzir em até aproximadamente 7 % o tempo de integração com relação a um ciclo, com relação a seu con-

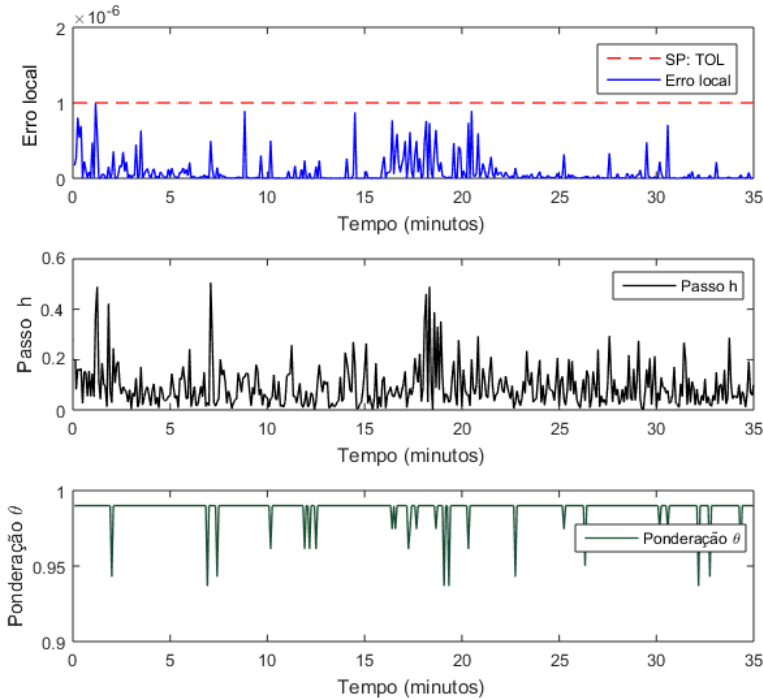


Figura 5.1: Cenário 2: método **mPV**: dinâmicas do erro local  $e_{n+1}$ , tamanho de passo  $h_{n+1}$  e ponderação  $\theta_{n+1}$ .

corrente de melhor performance (ode15s). De fato, para cada passo dado do método, em média apenas 1,07 chamadas do modelo são realizadas, isto é, dado que os gradientes dos passos imediatamente anteriores ( $f_n$ ) ao atual são armazenados, para calcular o método ponderado pela equação (2.11), não é necessária mais do que uma chamada do sistema por passo, a menos na primeira chamada do método, caso o sistema não se encontre inicialmente em seu ponto de operação, quando  $f_0 = 0_{n_x \times 1}$ .

Observando-se a Figura 5.2 é possível inferir que mais avaliações são calculadas justamente quando a variável controlada não segue a referência: no início da simulação, em resposta à atuação de controle com elevados ganhos, e posteriormente, como consequência da dinâmica do sistema em resposta à segunda perturbação aplicada, de maior magnitude.

Observa-se, também, a suavidade do comportamento da dinâmica das variáveis controlada (erro local) e manipulada (tamanho do passo) obtida com esta sintonia, contribuindo para garantir a fidelidade da solução numérica à analítica.

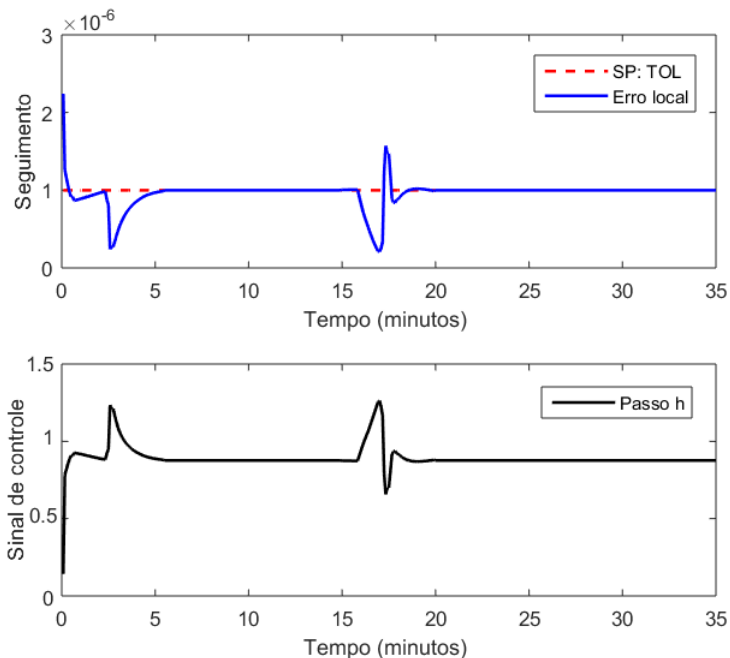


Figura 5.2: Cenário 2: método **mPI**: dinâmicas do erro local  $e_{n+1}$  e do tamanho de passo  $h_{n+1}$ .

Quanto à precisão, manteve-se o baixo número de soluções fora da faixa, que podem ser observadas claramente na Figura 5.2 nos instantes em que o erro local encontra-se acima da referência, chegando a pouco mais de 1 %, ou seja, das 420 soluções calculadas<sup>2</sup> (cada amostra correspondendo a 1 ciclo,  $T_s = 5$  segundos), somente 4 foram obtidas fora da precisão especificada.

É possível corrigir-se as imprecisões observadas adotando-se, por exemplo, um set-point mais apertado, tal como  $\mathcal{E}_{novo} = 0,9 \cdot \mathcal{E}_2$  no segundo cenário, adicionando um controle *feed-forward*<sup>3</sup> de

<sup>2</sup>Para um ciclo de amostragem  $T_s = 5$  segundos e um tempo de simulação de 35 minutos, tem-se  $(35 \times 60)/5 = 420$  amostras.

<sup>3</sup>Ou antecipatório. Detalhes em [31].

maneira a minimizar os efeitos da resposta à perturbação, e evitar que estas levem o erro local a ultrapassar a tolerância  $\mathcal{E}$ .

Quanto ao terceiro método implementado, **mNL**, obteve-se o segundo melhor resultado quanto à redução do tempo da tarefa de integração por ciclo, chegando a 5,54 % de diminuição, ficando próximo ao método **mPI** em desempenho, e realizando, em geral, apenas uma chamada do sistema a mais que este. Foi também o método que realizou a menor média de passos para calcular as aproximações.

A Figura 5.3 ilustra o desempenho dinâmico do erro local e do tamanho do passo. Observa-se um comportamento mais agressivo a o obtido com o método **mPI**, mantendo-se, entretanto, o seguimento do erro local à referência. Novamente, encontram-se imprecisões no início da simulação, e como resposta à ocorrência das duas perturbações aplicadas. Apenas uma amostra a mais foi observada, resultando também em um número próximo a 1 % do total de soluções entregues (5 em 420).

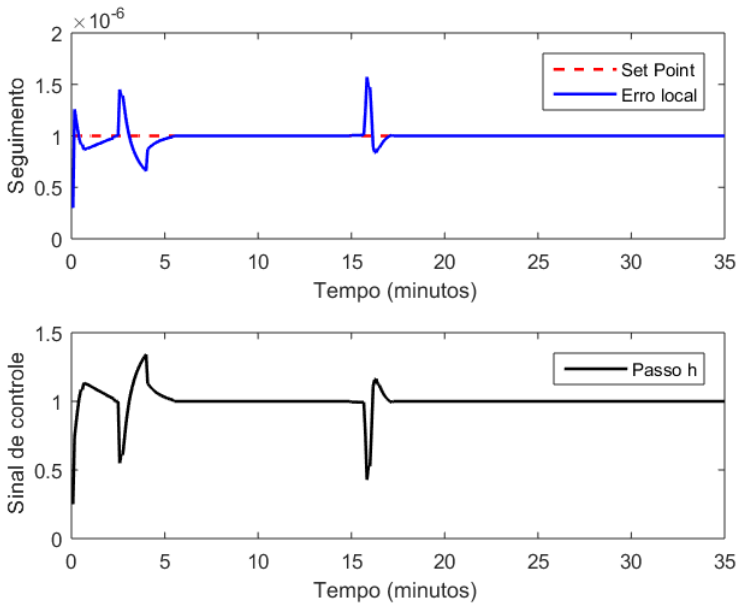


Figura 5.3: Método **mNL**: dinâmicas do erro local  $e_{n+1}$  e do tamanho de passo  $h_{n+1}$ .



Enfatiza-se, que estas imprecisões podem ser reduzidas adotando-se um set-point menor do que  $\mathcal{E}$  em conjunto com controladores antecipativos, bem como projetando-se uma dinâmica de malha fechada mais conservativa, através de um valor maior para o parâmetro  $z_f$ . Além disso, recorda-se que em todo projeto de controle é necessário levar em conta um compromisso entre robustez e rapidez do sistema. Neste sentido, o objetivo das metodologias implementadas é sobretudo melhorar a eficiência da simulação do algoritmo de controle avançado como um todo, mas buscando entregar soluções equilibradas quanto à rapidez, precisão e estabilidade numérica de integração.

### 5.3 COMENTÁRIOS FINAIS

Este capítulo apresentou os resultados da aplicação dos algoritmos de integração numérica desenvolvidos ao simulador de controle avançado de compressão de gás. Os três algoritmos obtiveram melhor desempenho em relação a metodologias típicas encontradas em pacotes de integração numérica, mesmo não fazendo uso de qualquer recurso extra de processamento computacional ou de compilação. Observou-se um resultado de maior evidencia quando empregou-se uma tolerância de precisão mais apertada (menor). Neste caso, e quando o sistema DAE é utilizado apenas como modelo de predição, foi obtida uma redução de até 7 % na contribuição temporal por ciclo da tarefa de integração do algoritmo de controle PNMPC. Naturalmente, a adição de uma chamada a mais do integrador por ciclo agregou uma carga computacional adicional, observando-se, em média, um custo de 10% a mais no tempo de um ciclo de amostragem. Foram obtidos melhores resultados também quanto à confiabilidade, observando-se um máximo de pouco mais de 1 % de soluções obtidas fora da faixa de precisão especificada, durante todo o período de simulação. Comparando-se os algoritmos desenvolvidos entre si, o método mPI foi o que apresentou melhor custo-benefício, considerando não somente a melhor eficiência obtida, mas sua simplicidade na sintonia do algoritmo, seguido do método mNL, alcançando uma eficiência semelhante, embora com implementação e sintonia mais onerosas. Por fim, o método mPV, embora menos eficiente, foi o que apresentou melhor precisão e confiabilidade. No próximo capítulo serão apresentadas as conclusões desta dissertação, as contribuições obtidas com relação aos resultados aqui evidenciados, e algumas sugestões de trabalhos futuros.



## 6 CONCLUSÕES

Esta dissertação abordou o estudo, aplicação de algoritmos de integração numérica, implementados de maneira a melhorar a eficiência de desempenho e simulação de um sistema de controle avançado. Como caso de estudo, o algoritmo de um controlador **PNMPC** aplicado a um sistema complexo de compressão de gás foi investigado.

O sistema de compressão foi modelado matematicamente por um conjunto de equações diferenciais e algébricas rígidas (**DAEs**), muitas das quais são não lineares. O sistema de controle **PNMPC** utiliza este modelo não linear para computar previsões das saídas do sistema que serão utilizadas para obtenção da ação de controle a ser enviada para a planta.

Para que sejam obtidas as respostas (saídas) do modelo de previsão, necessita-se resolvê-lo numericamente, em cada ciclo de amostragem do algoritmo, por meio de um *solver* integração numérica, que nem sempre apresentam bom desempenho devido complexidades na estrutura do Por meio deste resultado, este trabalho de pesquisa foi direcionado ao estudo de técnicas e metodologias de integração numérica que viessem a melhorar o desempenho computacional do algoritmo de controle, a fim de reduzir os tempos tomados com simulações, quando na fase de desenvolvimento e testes do sistema de controle **PNMPC** e melhorar o desempenho do controlador quando aplicado efetivamente em uma planta real.

Três algoritmos foram propostos para resolver o problema de integração numérica (**PVI**) eficientemente. Em comum, todas utilizam o método ponderado ( $\theta$ -method) com controle do tamanho passo de integração. Optou-se por esta técnica de aproximação da integral, pois foi constatado que ao utilizar um método simples e de baixa ordem, fornecendo apenas uma aproximação linear, mas adaptando o tamanho do passo com informações do comportamento dinâmico do sistema, e com dados quantitativos do próprio algoritmo de integração, obtinha-se um algoritmo mais rápido e tão ou mais preciso quanto técnicas típicas, que computavam aproximações polinomiais de mais alta ordem para calcular cada solução.

Isto devido ao fato de que chamadas do sistema para computar as aproximações e atualizar as variáveis algébricas eram muito custosas computacionalmente, em se tratando de um modelo complexo, com inúmeras equações rígidas e não lineares e, adicionalmente, por ser exigida uma especificação de alta precisão para cada solução. Adaptar-se ao comportamento dinâmico do sistema refere-

se ao ajuste do tamanho do passo ( $h$ ) de acordo com a similaridade das soluções. Com o sistema em regime permanente, em seu ponto de operação, habilita-se ao método utilizar tamanhos de passos maiores.

O primeiro algoritmo levou em conta não apenas o controle de passo, mas também a utilização de uma ponderação  $\theta$  variável, regulando-se assim a direção das aproximações e ajustando-se a implicidade do método de acordo com o número de passos tomados, ou a dificuldade para se calcular uma solução coerente com os requisitos de precisão, levando-se em conta a estabilidade numérica do método utilizado.

Na segunda estratégia, sintonizou-se um controlador PI para regular a estimativa de erro local,  $e_{n+1}$ , tomando como referência as tolerâncias ( $\mathcal{E}_1$  e  $\mathcal{E}_2$ ) especificadas para esta variável, manipulando-se o tamanho de passo de integração. Definiu-se uma ponderação  $\theta$  fixa para o método, pois os resultados de simulação com a primeira metodologia apresentaram pouca variação neste parâmetro, obtendo-se um valor médio mais próximo da unidade, ou seja, mais explícito. Quanto à estabilidade numérica da técnica, esta foi assegurada através da limitação da magnitude do tamanho de passo. Para limitar a ação integral, foi adicionada ao controle da malha uma parcela *anti wind-up*.

Já na terceira estratégia realizou-se, empiricamente, o levantamento de uma função estática não linear que relaciona o tamanho do passo de integração com o erro local, ou seja, uma função inversa ( $\Phi^{-1}$ ) à utilizada para descrever a estimativa do erro local. Desta forma, através do cancelamento de não linearidades foi possível identificar um modelo linear resultante que descrevesse o erro local como uma função de um sinal auxiliar de erro, através da transformada  $\mathcal{Z}$ . Este sinal auxiliar foi então manipulado para controlar o erro local, por meio do projeto de um controlador por modelo interno, ou IMC. Optou-se por esta técnica clássica, pois ela leva em consideração no projeto de controle possíveis incertezas, erros de modelagem e perturbações não medidas na malha de controle. Além disso, a sintonia do controlador é simples, tendo apenas um parâmetro a ajustar, e a estabilidade numérica do método é garantida também pela limitação do tamanho do passo de integração.

Foram realizados ensaios de simulação visando avaliar as estratégias de integração propostas. Dois cenários foram propostos: no primeiro considerou-se a fase de desenvolvimento e simulação de um algoritmo de controle, onde o sistema DAE foi empregado tanto como modelo de predição, como para simular a resposta da

planta. No segundo cenário, procurou-se estudar a aplicação pura do controlador **PNMPC**, avaliando-se a eficiência somente quanto à obtenção das respostas do modelo de predição. Os resultados obtidos foram comparados com *solvers* comerciais, aproveitando-se inclusive do pacote de *solvers* disponível na própria plataforma de simulação utilizada.

Através de indicadores de desempenho e eficiência computacional de algoritmos numéricos, verificou-se que as metodologias desenvolvidas não somente apresentaram melhor rapidez, como entregaram uma solução mais precisa e confiável, independente do cenário de simulação e especificação de precisão. A estratégia **mPI** foi a que apresentou o melhor custo-benefício, sendo não apenas a mais eficiente, obtendo o melhor desempenho em todos os cenários de simulação, como também a de ajuste mais simples, tendo somente os ganhos do controlador **PI** e a frequência de corte do filtro passa-baixas a sintonizar.

Embora os três algoritmos desenvolvidos dependam de conhecimento prévio do modelo **DAE** a resolver, em especial, para calcular o intervalo de estabilidade das variáveis manipuladas, considera-se a implementação destes válida haja vista a eficiência computacional obtida com a aplicação destas, a qual, independente da quantidade de chamadas dos *solvers* e da especificação de precisão empregada, foi sempre melhor que a dos concorrentes, e com alta confiabilidade. Isto motiva a aplicação de modelos de predição mais complexos e ricos em informação do processo a controlar, visando uma resposta de controle o mais robusta e fiel possível à sintonia especificada, assim portanto melhorando o desempenho de operação da planta.

As principais contribuições deste trabalho são relacionadas a seguir:

- A implementação de rotinas de integração numérica para a resolução dedicada e eficiente de um sistema de compressão de gás modelado por um conjunto complexo de equações diferenciais e algébricas rígidas;
- A identificação de um modelo dinâmico linear para o erro local de aproximação através da técnica de cancelamento de não linearidade;
- O desenvolvimento de controladores **PI** e a aplicação da teoria de controle linear para melhorar a eficiência de rotinas de integração numérica.

Deve-se ressaltar a inexistência da aplicação de técnicas de identificação de sistemas e de projeto de controladores lineares para o controle do erro local de integração, bem como a aplicação de métodos numéricos adaptativos com variação da direção da aproximação futura, a partir do método ponderado. Destaca-se quanto ao projeto do controlador **IMC**, como uma vantagem para aplicação em outros problemas, sua robustez pela incorporação de incertezas do modelo **DAE** e de distúrbios não mensuráveis na malha de controle, e ainda, o uso de filtragem de sinais para melhorar o comportamento dinâmico das variáveis de controle, com o objetivo de melhorar a qualidade e confiabilidade das soluções computadas.

Como contribuição para o estudo de desenvolvimento de técnicas de controle aplicadas a rotinas de integração para a resolução numérica de problemas **DAEs** complexos, este trabalho gerou a seguinte publicação:

- ROMANO, J.M. et al. A semi-implicit dae solver applied to a stiff gas compression system. AADECA: Asociación Argentina de Control Automático: 25° Congreso Argentino de Control Automático. v. 1, p.1-6, 2016.

## 6.1 TRABALHOS FUTUROS

O sistema de compressão de gás estudado neste trabalho representa somente uma parte dos processos existentes em plataformas de petróleo. Existe muito trabalho a ser realizado na área de modelagem fenomenológica dos sistemas e equipamentos envolvidos no processamento (separação e compressão) de gás.

O uso de metodologias que contribuam para melhorar a eficiência computacional da simulação de sistemas complexos motiva também o estudo e a implementação de modelos fenomenológicos que agreguem um bom entendimento das relações físico-matemáticas dos processos, sem apreensão com a complexidade envolvida. Acredita-se que uma modelagem rigorosa aliada ao conhecimento prático possa contribuir para a diminuição de incertezas e melhorar o projeto de sistemas de controle avançado em plataformas de produção.

O desenvolvimento de algoritmos simples e computacionalmente eficientes e capazes de adaptar-se a qualquer tipo de problema complexo de valor inicial continua sendo um desafio. Alguns trabalhos de pesquisa podem contribuir para a elaboração de novas metodologias. Sugestões de temas são feitas a seguir:

- Aplicar e testar as metodologias desenvolvidos nestes trabalhos e suas sintonias em outros exemplos de modelos DAEs e sistemas de controle complexos;
- Empregar outras formas de medição da precisão que não somente o erro local;
- Aferir a estabilidade numérica de outras maneiras, por exemplo através do método de Lyapunov, para que esta não dependa do ponto de operação nem da linearização do sistema;
- Utilizar outras técnicas de controle de erro local nos métodos numéricos, cujas sintonias sejam assistidas por técnicas de inteligência artificial tais como controle *fuzzy* ou redes neurais;
- Dividir problemas de valor inicial rígidos em parte *rígida* e não-rígida, utilizando-se de um ou mais métodos numéricos resolvam mais eficientemente cada uma dessas partes. Um exemplo motivacional pode ser consultado em [5];
- Considerar a extensão do estudo, bem como o desenvolvimento e aplicação de metodologias que melhorem a eficiência numérica da integração em problemas de valor de contorno – **Boundary Value Problems (BVPs)**, podendo assim envolver aplicações que possuam condições finais, ou de chegada, para os estados.





## REFERÊNCIAS

- 1 THOMAS, J. E. *Fundamentos de Engenharia de Petróleo*. 2. ed. Rio de Janeiro, RJ: Interciência, 2004.
- 2 VETORAZZO, C. M. *Model Predictive Control of Gas Compression Stations in Off-Shore Production Platforms*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2016.
- 3 MENDES, P. R. C. *Controle avançado de um sistema de separação trifásica e tratamento de água*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2012.
- 4 AGUIAR, M. A. S. *An Augmented Lagrangian Method for Optimal control of continuous time DAE systems*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2016.
- 5 SÜLLI, E. *Numerical Solution of Ordinary Differential Equations*. [S.l.]: Mathematical Institute, University of Oxford, 2014.
- 6 PLUCENIO, A. et al. Modeling and control of a gas compression station used in a oil platform. *Anais do XXI Congresso Brasileiro de Automática*, 2016.
- 7 STOKKE, S.; STRAND, S.; SJONG, D. *Model Predictive Control (MPC) of a Gas-Oil-Water Separator Train*. [S.l.]: Springer, 2012. 701-713 p. ISBN 9401040613.
- 8 BUDINIS, S.; THORNHILL, N. F. *2nd IFAC Workshop on Automatic Control in Offshore Oil and Gas Production*.
- 9 CAMACHO, E. F.; ALBA, C. B. *Model Predictive Control (Advanced Textbooks in Control and Signal Processing)*. [S.l.]: Springer, 2007. ISBN 978-0-85729-398-5.
- 10 PLUCENIO, A. *Desenvolvimento de técnicas de controle não linear para elevação de fluidos multifásicos*. Tese (Doutorado) — Universidade Federal de Santa Catarina, 2010.
- 11 PLUCENIO, A. et al. Model Predictive Control Algorithm with Stability Step. *Anais do XX Congresso Brasileiro de Automática*, p. 2230–2237, 2014.
- 12 THOME, F. F. *Controle preditivo robusto de sistemas não lineares com atraso*. Dissertação (Mestrado) — Universidade Federal de Santa Catarina, Florianópolis, 2013.

- 13 HAIRER, E.; LUBICH, C.; ROCHE, M. *The Numerical Solution of Differential-Algebraic Systems by Runge-Kutta Methods*. 2. ed. [S.l.]: Springer, 1989. ISBN 978-3-540-46832-5.
- 14 AIKEN, I.; RICHARD, C. *Stiff Computation*. [S.l.]: Oxford University Press, 1985. ISBN 0195034538.
- 15 HAIRER, E.; WANNER, G. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Equations*. [S.l.]: Springer, 1996. ISBN 978-3-642-05221-7.
- 16 ASCHER, U. M.; PETZOLD, L. R. *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*. [S.l.]: SIAM: Society for Industrial and Applied Mathematics, 1998. ISBN 0898714125.
- 17 ATKINSON, K.; HAN, W.; STEWART, D. E. *Numerical Solution of Ordinary Differential Equations*. [S.l.]: Wiley, 2009. ISBN 047004294X.
- 18 GUSTAFSSON, K.; LUNDH, M.; SÖDERLIND, G. A PI Stepsize Control for the Numerical Solution of Ordinary Differential Equations. *BIT Computer Science and Numerical Mathematics*, Lawrence, KS, USA, v. 28, n. 2, p. 270–287, 1988. ISSN 0006-3835.
- 19 TORELLI, F. et al. A new approach for solving DAE systems applied to distribution networks. *49th International Universities Power Engineering Conference (UPEC)*, p. 1–6, 2014.
- 20 ANTONOV, A. et al. *Numerical Solution of Differential Algebraic Equations*. 1. ed. Lyngby, Denmark: IMM - Department of Mathematical Modelling, 1999.
- 21 GOPAL, G.; Sacks-Davis, R.; TISCHER, P. A Review of Recent Developments in Solving ODEs. *Computing Surveys*, v. 17, n. 1, 1985.
- 22 BIEGLER, L. T. *Nonlinear Programming: Concepts, Algorithms, and Applications to Chemical Processes*. [S.l.]: Springer, 2010. ISBN 978-0-89871-702-0.
- 23 LYCHE, T.; M., K. *Spline Methods Draft*. Department of Informatics Centre of Mathematics for Applications University of Oslo: [s.n.], 2008.
- 24 WANG, Z.; GUO, B. Legendre-Gauss-Radau Collocation Method for Solving Initial Value Problems of First Order Ordinary Differential Equations. *Journal of Scientific Computing*, v. 52, n. 1, p. 226–255, 2012. ISSN 1573-7691.

- 25 SIMOS, T. *Recent Advances in Computational and Applied Mathematics*. [S.l.]: Springer, 2010.
- 26 SILVA, K. R. de Aguiar da. *Regiões de Estabilidade de Métodos Numéricos para Equações Diferenciais*. Tese (Doutorado) — Universidade federal de Santa Catarina, 2007.
- 27 FUKUMA, N.; MORI, T. Schur Stability of Linear Combinations of Polynomials. *T.IEEJ - The Institute of Electrical Engineers of Japan*, Japan, v. 113-C, n. 6, 1993.
- 28 GRAVDAHL, J.; EGELAND, O. Speed and surge control for a low order centrifugal compressor model. *IEEE*, 1998.
- 29 HANSEN, C. *Dynamic Simulation of Compressor Control Systems*. Tese (Doutorado), 2008.
- 30 HELVOIRT, J. van. Centrifugal compressor surge: modeling and identification for control. Technische Universiteit Eindhoven, 2007.
- 31 OGATA, K. *Modern Control Engineering*. 4th. ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001. ISBN 0130609072.
- 32 ENRIGHT, W. H.; HULL, T. E.; LINDBERG, B. Comparing numerical methods for stiff systems of ODEs. *BIT Numerical Mathematics*, v. 15, n. 1, p. 10–48, 1975.
- 33 HIGHAM, D.; HALL, G. Embedded Runge-Kutta formulas with stable equilibrium states. *Journal of Computational and Applied Mathematics*, v. 29, n. 1, p. 25–33, 1990. ISSN 0377-0427.
- 34 COELHO, A.; COELHO, L. dos S. *Identificação de sistemas dinâmicos lineares*. [S.l.]: Editora da UFSC, 2004. (Didáctica (Editora da UFSC)). ISBN 9788532802804.
- 35 RIVERA, D.; MORARI, M.; SKOGESTAD, S. Internal Model Control. 4. PID Controller Design. *Industrial I& Engineering Chemistry, Process Design and Development*, v. 25, n. 1, p. 252–265, 1 1986. ISSN 0019-7882.
- 36 SHAMPINE, L. F.; REICHEL, M. W. The MATLAB ODE Suite. *SIAM J. Sci. Comput.*, v. 18, n. 1, p. 1–22, jan. 1997. ISSN 1064-8275.
- 37 PAN, V. Complexity computations with matrices and polynomials. *SIAM*, v. 34, n. 2, p. 225–262, 6 1992.



## APÊNDICE A - CONTROLE PREDITIVO PNMPC

O controle PNMPC (MPC Prático para Sistemas Não-Lineares) desenvolvido em [10] é baseado em algoritmos de controle preditivo que utilizam a representação de um vetor de predições,  $\hat{\mathbf{Y}}$  ao longo de um horizonte de (tempo) predição finito  $p$ , como uma função de um vetor com  $m$  mudanças na ação de controle  $\Delta\mathbf{u}$ , e um vetor  $\mathbf{F}$  que corresponde à resposta livre do sistema, isto é, a resposta do sistema quando não há variações no sinal de controle  $\mathbf{u}$ . Quando o modelo é linear, expressam-se as predições tal como a equação 1.

$$\hat{\mathbf{Y}} = \mathbf{G}\Delta\mathbf{u} + \mathbf{F} \quad (1)$$

onde  $\mathbf{G}$  corresponde a uma matriz contendo a resposta ao degrau do sistema, tal que o produto  $\mathbf{G}\Delta\mathbf{u}$  represente a resposta forçada do sistema.

Para modelos lineares, é possível calcular  $\hat{\mathbf{Y}}$  de forma simples, devido ao princípio de superposição das respostas livre e forçada, tal como expresso em 1. Entretanto, quando o modelo utilizado é não linear, é necessário a obtenção de uma aproximação para o vetor de predições  $\hat{\mathbf{Y}}$ .

A técnica PNMPC difere-se de outras técnicas NMPC (*nonlinear MPC*) principalmente pelo fato de utilizar modelos linearizados independentes dos pontos de equilíbrio do sistema. É proposta uma forma de descrever a evolução das saídas do sistema ao longo do horizonte de predição utilizando-se uma representação linear das saídas em relação aos incrementos de controle futuros. Para isto, não se utiliza o conceito de ponto de equilíbrio, assumindo-se que as predições  $\mathbf{Y}_p$  dependam apenas das entradas passadas  $\tilde{\mathbf{u}}$ , das saídas passadas  $\tilde{\mathbf{y}}$  e dos incrementos de entradas futuros  $\Delta\mathbf{u}$ , isto é,

$$\mathbf{Y}_p = \mathbf{h}(\tilde{\mathbf{y}}, \tilde{\mathbf{u}}, \Delta\mathbf{u}) \quad (2)$$

O vetor com as predições é reescrito como

$$\mathbf{Y}_p = \mathbf{G}_{\text{PNMPC}}\Delta\mathbf{u} + \mathbf{F} \quad (3)$$

onde

$$\mathbf{F} = \mathbf{h}(\tilde{\mathbf{y}}, \tilde{\mathbf{u}}) \quad (4)$$

e

$$\mathbf{G}_{\text{PNMPC}} = \frac{\partial \mathbf{Y}_p}{\partial \Delta\mathbf{u}} \quad (5)$$

A matriz  $\mathbf{G}_{\text{PNMPC}}$  é o Jacobiano de  $\mathbf{Y}_p$ . Esta representação pode ser utilizada tanto para sistemas representados por modelos lineares como para aqueles representados por modelos não lineares desde que as saídas sejam contínuas e diferenciáveis em relação às entradas [3].

A vantagem e importância da técnica PNMPC concentra-se no fato de que com esta solução alternativa de expressar as previsões torna-se possível resolver um problema de otimização utilizando apenas programação quadrática para a obtenção do sinal de controle, tornando o algoritmo de controle preditivo mais eficiente, e numericamente mais simples de resolver.

### OBTENÇÃO DAS MATRIZES $\mathbf{G}_{\text{PNMPC}}$ E $\mathbf{F}$

No controlador PNMPC a matriz  $\mathbf{G}_{\text{PNMPC}}$  e o vetor de resposta livre  $\mathbf{F}$  são obtidos de forma numérica. Para isso executa-se um algoritmo que calcule o vetor com as  $p$  previsões  $\mathbf{Y}_p$  quando se fornece os valores das entradas e saídas passadas, e o vetor com os  $m$  incrementos de entrada futura  $\Delta\mathbf{u}$ . Se denominarmos o conjunto de entradas passadas de  $\mathbf{u}_{\text{passado}}$  e as entradas atuais e passadas de  $\mathbf{y}_{\text{passado}}$ , então as previsões no instante  $k$  podem ser reescritas como na expressão 6.

$$\begin{aligned}
 \mathbf{y}_p(k+1) &= \mathbf{h}(\mathbf{y}_{\text{passado}}, \mathbf{u}_{\text{passado}}, \Delta\mathbf{u}(k)) \\
 \mathbf{y}_p(k+2) &= \mathbf{h}(\mathbf{y}_{\text{passado}}, \mathbf{u}_{\text{passado}}, \Delta\mathbf{u}(k), \Delta\mathbf{u}(k+1)) \\
 &\vdots \\
 \mathbf{y}_p(k+p) &= \mathbf{h}(\mathbf{y}_{\text{passado}}, \mathbf{u}_{\text{passado}}, \Delta\mathbf{u}(k), \Delta\mathbf{u}(k+1), \dots, \\
 &\quad \Delta\mathbf{u}(k+m-1))
 \end{aligned} \tag{6}$$

Dado que  $\mathbf{F}$  é o vetor de predições que seria obtido para  $\Delta\mathbf{u} = 0$  e  $\mathbf{G}_{\text{PNMPC}}$  é o gradiente das saídas previstas com relação ao vetor dos incrementos de controle, a expressão 3 pode ser escrita para um sistema de uma entrada e uma saída como:

$$\begin{bmatrix} y_p(k+1) \\ y_p(k+2) \\ \vdots \\ y_p(k+p) \end{bmatrix} \cong \begin{bmatrix} \frac{\partial \hat{y}(k+1)}{\partial \Delta u(k)} & 0 & \dots & 0 \\ \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k)} & \frac{\partial \hat{y}(k+2)}{\partial \Delta u(k+1)} & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial y_p(k+p)}{\partial \Delta u(k)} & \frac{\partial y_p(k+p)}{\partial \Delta u(k+1)} & \dots & \frac{\partial y_p(k+p)}{\partial \Delta u(k+m-1)} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m-1) \end{bmatrix} + \dots \quad (7)$$

$$\dots + \begin{bmatrix} f_1(y, u)_{passado} \\ f_2(y, u)_{passado} \\ \vdots \\ f_p(y, u)_{passado} \end{bmatrix}$$

Nota-se nesse caso a forma triangular inferior da matriz Jacobiana devido a causalidade do sistema [10], ou seja  $\frac{\partial y_p(k+j)}{\partial \Delta u(k+i)} = 0$  para  $i \geq j$ , e  $i \in [1, p]$ ,  $j \in [1, m]$ . A derivada parcial  $\frac{\partial y_p(k+j)}{\partial \Delta u_k}$  é definida como:

$$\frac{\partial y_p(k+j)}{\partial \Delta u_k} = \lim_{\Delta u_k \rightarrow 0} \frac{y_p(k+j)(u_{k-1} + \Delta u_k) - y_p(k+j)(u_{k-1})}{\Delta u_k} \quad (8)$$

Para um sistema monovariável, por exemplo, a cada iteração executa-se o seguinte procedimento para a obtenção numérica de **F** e **G<sub>PNMPC</sub>**:

- (i) Obtém-se o vetor  $\mathbf{Y}_p^0$  (dimensão  $p \times 1$ ) executando o modelo com as entradas e saídas passadas e com  $\Delta \mathbf{u} = [0 \ 0 \dots 0]^T$ .  $\mathbf{F} = \mathbf{Y}_p^0$ .
- (ii) Calcula-se a primeira coluna da matriz **G<sub>PNMPC</sub>**. Obtém-se o vetor  $\mathbf{Y}_p^1$  (dimensão  $p \times 1$ ) executando o modelo com as entradas e saídas passadas e com  $\Delta \mathbf{u} = [\epsilon \ 0 \dots 0]^T$ , onde  $\epsilon$  é um valor muito pequeno,  $\frac{u_{k-1}}{1000}$ , por exemplo.  $\mathbf{G}_{PNMPC}(:, 1) = \frac{\mathbf{Y}_p^1 - \mathbf{Y}_p^0}{\epsilon}$ .
- (iii) Calcula-se a segunda coluna de **G<sub>PNMPC</sub>**. Obtém-se o vetor  $\mathbf{Y}_p^2$  (dimensão  $p \times 1$ ) executando o modelo com as entradas e saídas passadas e com  $\Delta \mathbf{u} = [0 \ \epsilon \dots 0]$ .  $\mathbf{G}_{PNMPC}(:, 2) = \frac{\mathbf{Y}_p^2 - \mathbf{Y}_p^0}{\epsilon}$ .
- (iv) Prossegue-se com o cálculo das demais colunas de **G<sub>PNMPC</sub>** até a última coluna, onde obtém-se o vetor  $\mathbf{Y}_p^p$  executando o modelo com as entradas e saídas passadas e com  $\Delta \mathbf{u} = [0 \ 0 \dots \epsilon]^T$ .  $\mathbf{G}_{PNMPC}(:, m) = \frac{\mathbf{Y}_p^p - \mathbf{Y}_p^0}{\epsilon}$ .

Para a aplicação da técnica PNMPC em sistemas multivariáveis é necessário montar os vetores de predição e incrementos de ação de controle aumentados. Assim, para um sistema de  $ne$  entradas e  $ns$  saídas teríamos os seguintes vetores:

$$\begin{aligned} \mathbf{Y}_p &= [\mathbf{Y}_{p1} \mathbf{Y}_{p2} \dots \mathbf{Y}_{pns}]^T \\ \Delta \mathbf{u} &= [\Delta \mathbf{u}_1 \Delta \mathbf{u}_2 \dots \Delta \mathbf{u}_{ne}]^T \end{aligned} \quad (9)$$

A obtenção de  $\mathbf{F}$  e  $\mathbf{G}_{PNMPC}$  acontece da mesma forma como descrito anteriormente. A diferença é que agora a matriz  $\mathbf{G}_{PNMPC}$  passa a ter  $ne \times ns$  blocos calculados como no caso monovariável para cada par de entrada e saída.

### OBTENÇÃO DA AÇÃO DE CONTROLE

A ação de controle visa cumprir o objetivo geral de um controle MPC, que é fazer com que a saída futura  $y_p$  siga um determinado sinal de referência  $w$  em um determinado horizonte de tempo, penalizando o esforço incremental de controle  $\Delta u$ . Uma função para esse objetivo é geralmente expressa matematicamente por:

$$\begin{aligned} J = \sum_{j=N_1}^p \delta(j) [y_p(t+j|t) - w(t+j)]^2 + \dots \\ \dots + \sum_{j=1}^m \lambda(j) [\Delta u(t+j-1)]^2 \end{aligned} \quad (10)$$

onde a expressão  $y_p(t+j|t)$  corresponde à saída futura no instante  $(t+j)$ , calculada no instante  $t$ . Os índices  $N_1$  e  $p$  são os horizontes de predição mínimo e máximo enquanto  $m$  é o horizonte de controle. Esses índices definem os instantes em que deseja-se que a referência  $w$  siga a saída e onde é importante limitar a ação de controle [3]. Esses horizontes podem ser usados para compensar atrasos de transporte e fase não-mínima. Nos casos com um atraso de transporte  $d$ , deve-se escolher  $N_1 > d$  pois não haverá resposta do sistema a um sinal de entrada  $u(t)$  antes de  $t = d$ . Em casos onde o processo não possui atraso de transporte utiliza-se  $N_1 = 1$ . A variação de  $N_u$  permite penalizar por mais ou menos tempo a ação de controle. Os coeficientes  $\delta(j)$  e  $\lambda(j)$  são as sequências de ponderação do erro e do esforço de controle e geralmente são escolhidas constantes ou exponenciais, dependendo se queremos considerar mais instantes iniciais ou finais da resposta.



O sinal de controle a cada iteração é proveniente da minimização de 10. À função objetivo, é possível ainda estabelecer restrições às variáveis de entrada e saída, pois na prática, todos os sistemas reais estão sujeitos a limites físicos e de segurança. Exemplos são os limites máximos e mínimos impostos aos atuadores (como, por exemplo, válvulas), ou os valores limites que podem ser atingidos pelas saídas de um sistema devido a questões de segurança. Além disso, existem restrições do tipo econômico para o funcionamento dos sistemas que, em geral, levarão a escolher pontos de operação muito próximos destes limites. Assim, otimiza-se a qualidade e a relação custo-benefício do processo produtivo.

Utilizando-se a equação 3 como forma de expressar as predições, pode-se reescrever a função objetivo da seguinte forma:

$$J = (\mathbf{Y}_p - \mathbf{W})^T \mathbf{R} (\mathbf{Y}_p - \mathbf{W}) + \Delta \mathbf{u}^T \mathbf{Q} \Delta \mathbf{u}$$

ou

$$J = (\mathbf{G}_{\text{PNMPC}} \Delta \mathbf{u} + \mathbf{F} - \mathbf{W})^T \mathbf{R} (\mathbf{G}_{\text{PNMPC}} \Delta \mathbf{u} + \mathbf{F} - \mathbf{W}) + \Delta \mathbf{u}^T \mathbf{Q} \Delta \mathbf{u} \quad (11)$$

onde  $\mathbf{W}$  representa o vetor de referências futuras,  $\mathbf{R}$  a matriz de ponderação do erro e  $\mathbf{Q}$  a matriz de ponderação do controle. A minimização da função custo pode ser obtida, então, resolvendo-se um problema de otimização do tipo QP (quadrático), da mesma forma que no caso linear. Pode-se expressar as restrições nas entradas e saídas em função dos incrementos de controle  $\Delta \mathbf{u}$ , de maneira que o problema possa ser expresso matematicamente por:

$$\begin{aligned} \min_{\Delta \mathbf{u}} J &= \frac{1}{2} \Delta \mathbf{u}^T \mathbf{H} \Delta \mathbf{u} + \mathbf{b} \Delta \mathbf{u} + \mathbf{f}_0 \\ \text{Sujeito a } & \mathbf{A} \Delta \mathbf{u} \leq \mathbf{b} \end{aligned} \quad (12)$$

onde

$$\begin{aligned} \mathbf{H} &= 2\mathbf{G}_{\text{PNMPC}}^T \mathbf{R} \mathbf{G}_{\text{PNMPC}} + \mathbf{Q} \\ \mathbf{b} &= 2(\mathbf{F} - \mathbf{W})^T \mathbf{R} \mathbf{G} \\ \mathbf{f}_0 &= (\mathbf{F} - \mathbf{W})^T \mathbf{R} (\mathbf{F} - \mathbf{W}) \end{aligned} \quad (13)$$

O sinal de controle a ser enviado para a planta é, portanto, dado por:

$$\mathbf{u}(k) = \mathbf{u}(k-1) + \Delta \mathbf{u}(k) \quad (14)$$

Detalhes e alternativas de implementação da técnica PNMPC podem ser estudados em [10].

## APÊNDICE B - PARÂMETROS DE SIMULAÇÃO

### Tempo de simulação e período de amostragem

$$t_{sim} = 2100 \text{ s}$$

$$T_s = 5 \text{ s}$$

### Parâmetros físico-químicos da mistura gasosa

$$R = 8,314 \text{ J/K} \cdot \text{mol}$$

$$Z = 0,95$$

$$M_W = 0,019 \text{ kg/mol}$$

$$c_g = 3,0 \text{ J/kg} \cdot \text{K}$$

$$c_c = 4,2 \text{ J/kg} \cdot \text{K}$$

$$\sigma^{ij} = 0,258$$

### Dados da linha de entrada da estação

$$m_E = 21,58 \text{ m}^3/\text{s}$$

$$T_E = 320 \text{ K}$$

$$V_E = 6,0 \text{ m}^3$$

$$P_E^{max} = 713 \text{ kPa}$$

$$w_1^N = 1485,55 \text{ rad/s}$$

$$w_2^N = 1485,55 \text{ rad/s}$$

$$k_{flare} = 100$$

$$\tau_{flare} = 2,0 \text{ s}$$

### Dados das linhas de compressão

#### Trocadores de Calor

$$k_{TC}^i = 1,198 \cdot 10^6$$

$$T_c^{ij} = 277 \text{ K}$$

$$m_c^{\max,11} = m_c^{\max,21} = 40 \text{ m}^3/\text{s}$$

$$m_c^{\max,12} = m_c^{\max,22} = 40 \text{ m}^3/\text{s}$$

$$m_c^{\max,13} = m_c^{\max,23} = 60 \text{ m}^3/\text{s}$$

$$m_c^{\max,s1} = m_c^{\max,s2} = 40 \text{ m}^3/\text{s}$$

$$\tau_{T_s}^{ij} = 6,0 \text{ s}$$

$$\tau_{T_{si}} = 6,0 \text{ s}$$

Vasos de Pressão

$$V_{11} = 4,0 \text{ m}^3$$

$$V_{12} = 3,0 \text{ m}^3$$

$$V_{13} = 1,85 \text{ m}^3$$

$$V_{21} = 4,0 \text{ m}^3$$

$$V_{22} = 3,0 \text{ m}^3$$

$$V_{23} = 1,85 \text{ m}^3$$

$$V_H = 1,5 \text{ m}^3$$

Compressores

$$J_i = 150 \text{ kg} \cdot \text{m}^2$$

$$P_N^i = 8,42 \text{ MW}$$

$$w_N^i = 1485,55 \text{ rad/s}$$

$$\mu^i = 0,9$$

$$R_i = 0,9017 \text{ m}$$

Válvulas de Reciclo

$$K_{ss}^{11} = K_{ss}^{21} = 3,0$$

$$K_{ss}^{12} = K_{ss}^{22} = 1,0$$

$$K_{ss}^{13} = K_{ss}^{23} = 3,0$$

$$CV_N = 1,24$$

$$x_P = 0,885$$

$$\tau_{m_r}^{ij} = 2 \text{ s}$$

**Curvas de compressão: coeficientes dos polinômios**Primeiro estágio

$$a^{11} = a^{21} = 1,752$$

$$b^{11} = b^{21} = 0,08592$$

$$c^{11} = c^{21} = -14,18$$

$$d^{11} = d^{21} = -0,0004272$$

$$e^{11} = e^{21} = 0,003541$$

$$f^{11} = f^{21} = 11,65$$

Segundo estágio

$$a^{12} = a^{22} = 1,404$$

$$b^{12} = b^{22} = 0,1861$$

$$c^{12} = c^{22} = -9,422$$

$$d^{12} = d^{22} = -0,002173$$

$$e^{12} = e^{22} = -0,05412$$

$$f^{12} = f^{22} = 9,619$$

Terceiro estágio

$$a^{13} = a^{23} = 0,9599$$

$$b^{13} = b^{23} = 0,1881$$

$$c^{13} = c^{23} = -2,569$$

$$d^{13} = d^{23} = -0,00876$$

$$e^{13} = e^{23} = -0,05318$$

$$f^{13} = f^{23} = 4,139$$

**Dados das linhas de saída da estação**Linha de exportação

$$m_{Lexp} = 51,25 \text{ m}^3/\text{s}$$

$$k_{exp}^{pc} = 3,42559 \cdot 10^3$$

$$\tau_{exp} = 3,0 \text{ s}$$

$$V_{Lexp} = 100 \text{ m}^3$$

Linha de injeção

$$k_{L1} = 27,5 \cdot 10^5$$

$$k_{L2} = 16,0 \cdot 10^5$$

$$P_{L1}^s = 5,0 \text{ MPa}$$

$$P_{L2}^s = 6,0 \text{ MPa}$$

$$\tau_{L1} = 4,0 \text{ s}$$

$$\tau_{L2} = 3,0 \text{ s}$$

$$V_{L1} = V_{L2} = 100 \text{ m}^3$$

## Sintonia do Controle Regulatório

### Controle da Pressão de Sucção e do Header

$$P_H^{SP} = 17,6 \text{ MPa}$$

$$K_P^{PH} = 0,1$$

$$K_I^{PH} = 0,033$$

$$P_s^{11SP} = P_s^{21SP} = 602,59 \text{ kPa}$$

$$K_P^{P_s^{11}} = K_P^{P_s^{21}} = 1,25$$

$$K_I^{P_s^{11}} = K_I^{P_s^{21}} = 0,017$$

### Controle da Temperatura de Sucção e de Saída

$$T_s^{11SP} = T_s^{21SP} = 298,15 \text{ K}$$

$$T_s^{12SP} = T_s^{22SP} = 338,25 \text{ K}$$

$$T_s^{13SP} = T_s^{23SP} = 307,45 \text{ K}$$

$$T^{s1SP} = T^{s2SP} = 340 \text{ K}$$

$$K_P^{T_s^{11}} = K_P^{T_s^{21}} = 0,05$$

$$K_I^{T_s^{11}} = K_I^{T_s^{21}} = 0,05$$

$$K_P^{T_s^{12}} = K_P^{T_s^{22}} = 0,1$$

$$K_I^{T_s^{12}} = K_I^{T_s^{22}} = 0,05$$

$$K_P^{T_s^{13}} = K_P^{T_s^{23}} = 0,2$$

$$K_I^{T_s^{13}} = K_I^{T_s^{23}} = 0,1$$

$$K_P^{T^{s1}} = K_P^{T^{s2}} = 0,1$$

$$K_I^{T^{s1}} = K_I^{T^{s2}} = 0,05$$

### Controle da vazão das linhas de injeção

$$m_{inj1}^{SP} = 2,0 \text{ m}^3/\text{s}$$

$$m_{inj2}^{SP} = 2,5 \text{ m}^3/\text{s}$$

$$K_P^{inj1} = K_P^{inj2} = 2,0$$

$$K_I^{inj1} = K_I^{inj2} = 1,0$$

Controle Anti-Surge

$$K_P^{s^{ij}} = 0,175$$

$$K_I^{s^{ij}} = 0,058$$

$$f^{ij} = 0,1$$

**Condições Iniciais**Linha de entrada

$$P_E = 620 \text{ kPa}$$

$$m_{flare} = 0$$

$$w_1 = 1485,55 \text{ rad/s}$$

$$w_2 = 1485,55 \text{ rad/s}$$

Estágio 11

$$P_s^{11} = 602,59 \text{ kPa}$$

$$Ie_{P_s}^{11} = 0$$

$$T_s^{11} = 298,15 \text{ K}$$

$$Ie_{T_s}^{11} = -327,60 \text{ K}$$

$$m_{rc}^{11} = 0$$

$$Ie_{surge}^{11} = 0$$

Estágio 12

$$P_s^{12} = 2,07 \text{ MPa}$$

$$T_s^{12} = 338,25 \text{ K}$$

$$Ie_{T_s}^{12} = 0$$

$$m_{rc}^{12} = 0$$

$$Ie_{surge}^{12} = 0$$

Estágio 13

$$P_s^{13} = 6,40 \text{ MPa}$$

$$T_s^{13} = 307,46 \text{ K}$$

$$Ie_{T_s}^{13} = 0$$

$$m_{rc}^{13} = 0$$

$$Ie_{surge}^{13} = 0$$

Estágio 21

$$P_s^{21} = 602,59 \text{ kPa}$$

$$Ie_{P_s}^{21} = 0$$

$$T_s^{21} = 298,15 \text{ K}$$

$$Ie_{T_s}^{21} = -327,60 \text{ K}$$

$$m_{rc}^{21} = 0$$

$$Ie_{surge}^{21} = 0$$

Estágio 22

$$P_s^{22} = 2,07 \text{ MPa}$$

$$T_s^{22} = 338,25 \text{ K}$$

$$Ie_{T_s}^{22} = 0$$

$$Ie_{surge}^{22} = 0$$

$$m_{rc}^{22} = 0$$

$$Ie_{surge}^{11} = 0$$

Estágio 23

$$P_s^{23} = 6,40 \text{ MPa}$$

$$T_s^{23} = 307,46 \text{ K}$$

$$Ie_{T_s}^{23} = 0$$

$$m_{rc}^{23} = 0$$

$$Ie_{surge}^{23} = 0$$

Linha de saída trocador de calor

$$T^{s1} = 340 \text{ K}$$

$$Ie_{T^{s1}} = 0$$

$$T^{s2} = 340 \text{ K}$$

$$Ie_{T^{s2}} = 0$$

$$P_H = 17,6 \text{ MPa}$$

$$Ie_{PH} = 0$$

Linha de exportação

$$m_{exp}^s = 17,09 \text{ m}^3/\text{s}$$

$$P_{exp}^{ac} = 0$$



Linhas de injeção

$$m_{inj}^{s1} = 2,0 m^3/s$$

$$I_{inj1} = 0$$

$$m_{inj}^{s2} = 2,5 m^3/s$$

$$I_{inj2} = 0$$

$$P_{L1}^{ac} = 0$$

$$P_{L2}^{ac} = 0$$



## APÊNDICE C - EQUAÇÕES DIFERENCIAIS E ALGÉBRICAS

### Equações Diferenciais

$$\frac{dP_E}{dt} = \frac{RZT_E}{M_W V_E} \cdot (m_E - m_{flare} - m_E^{11} - m_E^{21})$$

$$\frac{dm_{flare}}{dt} = \frac{m_{flare}^N - m_{flare}}{\tau_{flare}}$$

$$\frac{dw_1}{dt} = \frac{1}{J_1} \left( \frac{u_P^1 P_N^1}{w_1} - (m_{CP}^{11} + m_{CP}^{12} + m_{CP}^{13}) \mu^1 w^1 R_1^2 \right)$$

$$\frac{dw_2}{dt} = \frac{1}{J_2} \left( \frac{u_P^2 P_N^2}{w_2} - (m_{CP}^{21} + m_{CP}^{22} + m_{CP}^{23}) \mu^2 w^2 R_2^2 \right)$$

$$\frac{dP_s^1}{dt} = \frac{RZT_s^{11}}{M_W V_{11}} \cdot (m_{TC}^{11} - m_{CP}^{11})$$

$$\frac{dIe_{P_s}^1}{dt} = e_{P_s}^1$$

$$\frac{dT_s^{11}}{dt} = \frac{T_{ss}^{11} - T_s^{11}}{\tau_{T_s^{11}}}$$

$$\frac{dIe_{T_s}^{11}}{dt} = e_{T_s}^{11}$$

$$\frac{dm_{rc}^{11}}{dt} = \frac{m_{rc}^{11} - m_r^{11}}{\tau_{m_r^{11}}}$$

$$\frac{dIe_{surge}^{11}}{dt} = e_{surge}^{11} + d_{AW}^{11}$$

$$\frac{dP_s^{12}}{dt} = \frac{RZT_s^{12}}{M_W V_{12}} \cdot (m_{TC}^{12} - m_{CP}^{12})$$

$$\frac{dT_s^{12}}{dt} = \frac{T_{ss}^{12} - T_s^{12}}{\tau_{T_s^{12}}}$$

$$\frac{dIe_{T_s}^{12}}{dt} = e_{T_s}^{12}$$

$$\frac{dm_{rc}^{12}}{dt} = \frac{m_{rc}^{12} - m_r^{12}}{\tau_{m_r^{12}}}$$

$$\frac{dIe_{surge}^{12}}{dt} = e_{surge}^{12} + d_{AW}^{12}$$

$$\frac{dP_s^{13}}{dt} = \frac{RZT_s^{13}}{M_W V_{13}} \cdot (m_{TC}^{13} - m_{CP}^{13})$$

$$\frac{dT_s^{13}}{dt} = \frac{T_{ss}^{13} - T_s^{13}}{\tau_{T_s^{13}}}$$

$$\frac{dIe_{T_s}^{13}}{dt} = e_{T_s}^{13}$$

$$\frac{dm_{rc}^{13}}{dt} = \frac{m_{rc}^{13} - m_r^{13}}{\tau_{m_r^{13}}}$$

$$\frac{dIe_{surge}^{13}}{dt} = e_{surge}^{13} + d_{AW}^{13}$$

$$\frac{dP_s^2}{dt} = \frac{RZT_s^{21}}{M_W V_{21}} \cdot (m_{TC}^{21} - m_{CP}^{21})$$

$$\frac{dIe_{P_s}^2}{dt} = e_{P_s}^2$$

$$\frac{dT_s^{21}}{dt} = \frac{T_{ss}^{21} - T_s^{21}}{\tau_{T_s^{21}}}$$

$$\frac{dIe_{T_s}^{21}}{dt} = e_{T_s}^{21}$$

$$\frac{dm_{rc}^{21}}{dt} = \frac{m_{rc}^{21} - m_r^{21}}{\tau_{m_r^{21}}}$$

$$\frac{dIe_{surge}^{21}}{dt} = e_{surge}^{21} + d_{AW}^{21}$$

$$\frac{dP_s^{22}}{dt} = \frac{RZT_s^{22}}{M_W V_{22}} \cdot (m_{TC}^{22} - m_{CP}^{22})$$

$$\frac{dT_s^{22}}{dt} = \frac{T_{ss}^{22} - T_s^{22}}{\tau_{T_s^{22}}}$$

$$\frac{dIe_{T_s}^{22}}{dt} = e_{T_s}^{22}$$

$$\frac{dm_{rc}^{22}}{dt} = \frac{m_{rc}^{22} - m_r^{22}}{\tau_{m_r^{22}}}$$

$$\frac{dIe_{surge}^{22}}{dt} = e_{surge}^{22} + d_{AW}^{22}$$

$$\frac{dP_s^{23}}{dt} = \frac{RZT_s^{23}}{M_W V_{23}} \cdot (m_{TC}^{23} - m_{CP}^{23})$$

$$\frac{dT_s^{23}}{dt} = \frac{T_{ss}^{23} - T_s^{23}}{\tau_{T_s^{23}}}$$

$$\frac{dIe_{T_s}^{23}}{dt} = e_{T_s}^{23}$$

$$\frac{dm_{rc}^{23}}{dt} = \frac{m_{rc}^{23} - m_r^{23}}{\tau_{m_r^{23}}}$$

$$\frac{dIe_{surge}^{23}}{dt} = e_{surge}^{23} + d_{AW}^{23}$$

$$\frac{dT^{s1}}{dt} = \frac{T_{ss}^{s1} - T^{s1}}{\tau_{T^{s1}}}$$

$$\frac{dIe_{T^{s1}}}{dt} = e_{T^{s1}}$$

$$\frac{dT^{s2}}{dt} = \frac{T_{ss}^{s2} - T^{s2}}{\tau_{T^{s2}}}$$

$$\frac{dIe_{T^{s2}}}{dt} = e_{T^{s2}}$$

$$\frac{dP_H}{dt} = \frac{RZT_{exp}}{M_W V} \cdot (m_{TC}^{s1} + m_{TC}^{s2} - m_{exp} - m_{inj}^1 - m_{inj}^2)$$

$$\frac{dIe_{P_H}}{dt} = e_{P_H}$$

$$\frac{dm_{exp}^s}{dt} = \frac{m_{exp} - m_{exp}^s}{\tau_{exp}}$$

$$\frac{dP_{exp}^{ac}}{dt} = k_{exp}^{ac} (m_{exp} - m_{exp}^s)$$

$$\frac{dm_{inj}^{s1}}{dt} = \frac{m_{inj}^1 - m_{inj}^{s1}}{\tau_{L1}}$$

$$\frac{dIe_{inj1}}{dt} = e_{inj1}$$

$$\frac{dm_{inj}^{s2}}{dt} = \frac{m_{inj}^2 - m_{inj}^{s2}}{\tau_{L1}}$$

$$\frac{dIe_{inj2}}{dt} = e_{inj2}$$

$$\frac{dP_{L1}^{ac}}{dt} = k_{L1}^{ac} (m_{inj}^1 - m_{inj}^{s1})$$

$$\frac{dP_{L2}^{ac}}{dt} = k_{L2}^{ac} (m_{inj}^2 - m_{inj}^{s2})$$

## Equações Algébricas

$$m_E^1 = k_{TC}^1 \sqrt{\frac{M_W}{RZT_E}} \sqrt{P_E(P_E - P_s^1)}$$

$$m_E^2 = k_{TC}^{12} \sqrt{\frac{M_W}{RZT_E}} \sqrt{P_E(P_E - P_s^2)}$$

$$m_{flare}^N = k_{flare} \sqrt{\frac{M_W}{RZT_E}} \sqrt{P_E(P_E - P_E^{max})}$$

$$m_{TC}^{11} = m_E^1 + m_{rc}^{11}$$

$$m_{TC}^{21} = m_E^2 + m_{rc}^{21}$$

$$m_{TC}^{12} = m_{CP}^{11} + m_{rc}^{12} - m_{rc}^{11}$$

$$m_{TC}^{13} = m_{CP}^{12} + m_{rc}^{13} - m_{rc}^{12}$$

$$m_{TC}^{22} = m_{CP}^{21} + m_{rc}^{22} - m_{rc}^{21}$$

$$m_{TC}^{23} = m_{CP}^{22} + m_{rc}^{23} - m_{rc}^{22}$$

$$m_{TC}^{s1} = m_{CP}^{13} - m_{rc}^{13}$$

$$m_{TC}^{s2} = m_{CP}^{23} - m_{rc}^{23}$$

$$r^{11} = a^{11} + b^{11}x_{11} + c^{11}y_1 + d^{11}x_{11}^2 + e^{11}y_1 + f^{11}y_1^2$$

$$r^{12} = a^{12} + b^{12}x_{12} + c^{12}y_1 + d^{12}x_{12}^2 + e^{12}y_1 + f^{12}y_1^2$$

$$r^{13} = a^{13} + b^{13}x_{13} + c^{13}y_1 + d^{13}x_{13}^2 + e^{13}y_1 + f^{13}y_1^2$$

$$r^{21} = a^{21} + b^{21}x_{21} + c^{21}y_2 + d^{21}x_{21}^2 + e^{21}y_2 + f^{21}y_2^2$$

$$r^{22} = a^{22} + b^{22}x_{22} + c^{22}y_2 + d^{22}x_{22}^2 + e^{22}y_2 + f^{22}y_2^2$$

$$r^{23} = a^{23} + b^{23}x_{23} + c^{23}y_2 + d^{23}x_{23}^2 + e^{23}y_2 + f^{23}y_2^2$$

$$P_d^{11} = r^{11}P_s^1$$

$$P_d^{12} = r^{12}P_s^{12}$$

$$P_d^{13} = r^{13}P_s^{13}$$

$$P_d^{21} = r^{21}P_s^2$$

$$P_d^{22} = r^{22}P_s^{22}$$

$$P_d^{23} = r^{23}P_s^{23}$$

$$m_{CP}^{11} = \frac{P_s^1 M_W}{RZT_s^{11}} Q^{11}$$

$$m_{CP}^{12} = \frac{P_s^{12} M_W}{RZT_s^{12}} Q^{12}$$

$$m_{CP}^{13} = \frac{P_s^{13} M_W}{R Z T_s^{13}} Q^{13}$$

$$m_{CP}^{21} = \frac{P_s^{21} M_W}{R Z T_s^{21}} Q^{21}$$

$$m_{CP}^{22} = \frac{P_s^{22} M_W}{R Z T_s^{22}} Q^{22}$$

$$m_{CP}^{23} = \frac{P_s^{23} M_W}{R Z T_s^{23}} Q^{23}$$

$$T_{ss}^{11} = \frac{m_{rc}^{11} c_g T_d^{11} + m_E^1 c_g T_E + \phi_c^{11} m_c^{max11} c_c T_c^{11}}{m_{TC}^{11} c_g + \phi_c^{11} m_c^{max11} c_c}$$

$$T_{ss}^{12} = \frac{m_{rc}^{12} c_g T_d^{12} + m_{CP}^{11} c_g T_d^{11} + \phi_c^{12} m_c^{max12} c_c T_c^{12}}{m_{TC}^{12} c_g + \phi_c^{12} m_c^{max12} c_c}$$

$$T_{ss}^{13} = \frac{m_{rc}^{13} c_g T_d^{13} + m_{CP}^{12} c_g T_d^{12} + \phi_c^{13} m_c^{max13} c_c T_c^{13}}{m_{TC}^{13} c_g + \phi_c^{13} m_c^{max13} c_c}$$

$$T_{ss}^{21} = \frac{m_{rc}^{21} c_g T_d^{21} + m_E^2 c_g T_E + \phi_c^{21} m_c^{max21} c_c T_c^{21}}{m_{TC}^{21} c_g + \phi_c^{21} m_c^{max21} c_c}$$

$$T_{ss}^{22} = \frac{m_{rc}^{22} c_g T_d^{22} + m_{CP}^{21} c_g T_d^{21} + \phi_c^{11} m_c^{max22} c_c T_c^{22}}{m_{TC}^{22} c_g + \phi_c^{22} m_c^{max22} c_c}$$

$$T_{ss}^{23} = \frac{m_{rc}^{23} c_g T_d^{23} + m_{CP}^{22} c_g T_d^{22} + \phi_c^{23} m_c^{max23} c_c T_c^{23}}{m_{TC}^{23} c_g + \phi_c^{23} m_c^{max23} c_c}$$

$$T_{ss}^{s1} = \frac{m_{TC}^{s1} c_g T_d^{13} + \phi_c^{s1} m_c^{maxs1} c_c T_c^{13}}{m_{TC}^{s1} c_g + \phi_c^{s1} m_c^{maxs1} c_c}$$

$$T_{ss}^{s2} = \frac{m_{TC}^{s2} c_g T_d^{23} + \phi_c^{s2} m_c^{maxs2} c_c T_c^{23}}{m_{TC}^{s2} c_g + \phi_c^{s2} m_c^{maxs2} c_c}$$

$$T_H = \frac{m_{TC}^{s1} T^{s1} + m_{TC}^{s2} T^{s2}}{m_{TC}^{s1} + m_{TC}^{s2}}$$

$$T_d^{11} = T_s^{11} (r^{11})^{\sigma^{11}}$$

$$T_d^{12} = T_s^{12} (r^{12})^{\sigma^{12}}$$

$$T_d^{13} = T_s^{13} (r^{13})^{\sigma^{13}}$$

$$T_d^{21} = T_s^{21} (r^{21})^{\sigma^{21}}$$

$$T_d^{22} = T_s^{22} (r^{22})^{\sigma^{22}}$$

$$T_d^{23} = T_s^{23} (r^{23})^{\sigma^{23}}$$

$$P_{exp}^{pc} = k_{exp}^{pc} (m_{exp}^s + m_{Lexp})^2$$

$$\begin{aligned}
P_{exp} &= P_{exp}^{ac} + P_{exp}^{pc} \\
P_{L1} &= P_{L1}^{ac} + k_{L1}(m_{inj}^{s1})^2 + P_{L1}^s \\
P_{L2} &= P_{L2}^{ac} + k_{L2}m_{p2-din}^2 + P_{L2}^s \\
u_P^1 &= K_P^{P^1} e_{P_s}^1 + K_I^{P^1} Ie_{P_s}^1 \\
u_P^2 &= K_P^{P^2} e_{P_s}^2 + K_I^{P^2} Ie_{P_s}^2 \\
\Phi_{exp} &= K_P^{PH} e_{PH} + K_I^{PH} Ie_{PH} \\
\Phi_c^{11} &= K_P^{T^{11}} e_{T_s}^{11} + K_I^{T^{11}} Ie_{T_s}^{11} \\
\Phi_c^{12} &= K_P^{T^{12}} e_{T_s}^{12} + K_I^{T^{12}} Ie_{T_s}^{12} \\
\Phi_c^{13} &= K_P^{T^{13}} e_{T_s}^{13} + K_I^{T^{13}} Ie_{T_s}^{13} \\
\Phi_c^{21} &= K_P^{T^{21}} e_{T_s}^{21} + K_I^{T^{21}} Ie_{T_s}^{21} \\
\Phi_c^{22} &= K_P^{T^{22}} e_{T_s}^{22} + K_I^{T^{22}} Ie_{T_s}^{22} \\
\Phi_c^{23} &= K_P^{T^{23}} e_{T_s}^{23} + K_I^{T^{23}} Ie_{T_s}^{23} \\
\Phi_c^{s1} &= K_P^{T^{s1}} e_{T^{s1}} + K_I^{T^{s1}} Ie_{T^{s1}} \\
\Phi_c^{s2} &= K_P^{T^{s2}} e_{T^{s2}} + K_I^{T^{s2}} Ie_{T^{s2}} \\
\Phi_{inj1} &= K_P^{inj1} e_{inj1} + K_I^{inj1} Ie_{inj1} \\
\Phi_{inj2} &= K_P^{inj2} e_{inj2} + K_I^{inj2} Ie_{inj2} \\
Q_{surge}^{11SP} &= Q_{surge}^{11}(1 + f^{11}) \\
Q_{surge}^{12SP} &= Q_{surge}^{12}(1 + f^{12}) \\
Q_{surge}^{13SP} &= Q_{surge}^{13}(1 + f^{13}) \\
Q_{surge}^{21SP} &= Q_{surge}^{21}(1 + f^{21}) \\
Q_{surge}^{22SP} &= Q_{surge}^{22}(1 + f^{22}) \\
Q_{surge}^{32SP} &= Q_{surge}^{21}(1 + f^{32}) \\
\Phi_r^{11} &= K_P^{s^{11}} e_{surge}^{11} + K_I^{s^{11}} Ie_{surge}^{11} \\
\Phi_r^{12} &= K_P^{s^{11}} e_{surge}^{12} + K_I^{s^{12}} Ie_{surge}^{12} \\
\Phi_r^{13} &= K_P^{s^{13}} e_{surge}^{13} + K_I^{s^{13}} Ie_{surge}^{13} \\
\Phi_r^{21} &= K_P^{s^{21}} e_{surge}^{21} + K_I^{s^{21}} Ie_{surge}^{21} \\
\Phi_r^{22} &= K_P^{s^{22}} e_{surge}^{22} + K_I^{s^{22}} Ie_{surge}^{22} \\
\Phi_r^{23} &= K_P^{s^{23}} e_{surge}^{23} + K_I^{s^{23}} Ie_{surge}^{23}
\end{aligned}$$



$$m_r^{11} = K_{ss}^{11} \phi_r^{11} CV_N P_d^{11} \sqrt{\frac{M_W x_P}{ZRT_d^{11}}}$$

$$m_r^{12} = K_{ss}^{12} \phi_r^{12} CV_N P_d^{12} \sqrt{\frac{M_W x_P}{ZRT_d^{12}}}$$

$$m_r^{13} = K_{ss}^{13} \phi_r^{13} CV_N P_d^{13} \sqrt{\frac{M_W x_P}{ZRT_d^{13}}}$$

$$m_r^{21} = K_{ss}^{21} \phi_r^{21} CV_N P_d^{11} \sqrt{\frac{M_W x_P}{ZRT_d^{21}}}$$

$$m_r^{22} = K_{ss}^{22} \phi_r^{22} CV_N P_d^{22} \sqrt{\frac{M_W x_P}{ZRT_d^{22}}}$$

$$m_r^{23} = K_{ss}^{23} \phi_r^{23} CV_N P_d^{23} \sqrt{\frac{M_W x_P}{ZRT_d^{23}}}$$

$$d_{AW}^{11} = \begin{cases} (1 - \Phi_r^{11}), & se \ \Phi_r^{11} > 1 \\ -\Phi_r^{11}, & se \ \Phi_r^{11} < 0 \end{cases}$$

$$d_{AW}^{12} = \begin{cases} (1 - \Phi_r^{12}), & se \ \Phi_r^{12} > 1 \\ -\Phi_r^{12}, & se \ \Phi_r^{12} < 0 \end{cases}$$

$$d_{AW}^{13} = \begin{cases} (1 - \Phi_r^{13}), & se \ \Phi_r^{13} > 1 \\ -\Phi_r^{13}, & se \ \Phi_r^{13} < 0 \end{cases}$$

$$d_{AW}^{21} = \begin{cases} (1 - \Phi_r^{21}), & se \ \Phi_r^{21} > 1 \\ -\Phi_r^{21}, & se \ \Phi_r^{21} < 0 \end{cases}$$

$$d_{AW}^{22} = \begin{cases} (1 - \Phi_r^{22}), & se \ \Phi_r^{22} > 1 \\ -\Phi_r^{22}, & se \ \Phi_r^{22} < 0 \end{cases}$$

$$d_{AW}^{23} = \begin{cases} (1 - \Phi_r^{23}), & se \ \Phi_r^{23} > 1 \\ -\Phi_r^{23}, & se \ \Phi_r^{23} < 0 \end{cases}$$