



# UFBA

UNIVERSIDADE FEDERAL DA BAHIA  
ESCOLA POLITÉCNICA  
PROGRAMA DE PÓS GRADUAÇÃO EM  
ENGENHARIA INDUSTRIAL - PEI

MESTRADO EM ENGENHARIA INDUSTRIAL

GESNER ANDRADE NERY JÚNIOR

SINTONIA ÓTIMA DE CONTROLADORES MPC  
CONSIDERANDO INCERTEZAS DE MODELAGEM



SALVADOR  
2015



SINTONIA ÓTIMA DE CONTROLADORES MPC CONSIDERANDO  
INCERTEZAS DE MODELAGEM

Gesner Andrade Nery Júnior

Dissertação de Mestrado Profissional  
apresentada ao Programa de Pós-graduação  
em Engenharia Industrial, da Universidade  
Federal da Bahia, como parte dos requisitos  
necessários à obtenção do título de Mestre  
em Engenharia Industrial.

Orientadores: Ricardo de Araújo Kalid  
Márcio A. F. Martins

Salvador

Janeiro de 2015

---

N455 Nery Júnior, Gesner Andrade.  
Sintonia ótima de controladores MPC considerando incertezas de modelagem/ Gesner Andrade Nery Júnior. – Salvador, 2015.

92 f.: il. color.

Orientador: Prof. Ricardo de Araújo Kalid.

Co-orientador: Prof. Márcio A. F. Martins.

Dissertação (Mestrado) – Universidade Federal da Bahia.  
Escola Politécnica, 2015.

1. Otimização matemática. 2. Controle preditivo. 3. Modelos matemáticos. I. Kalid, Ricardo de Araújo. II. Martins, Márcio A. F. III. Universidade Federal da Bahia. IV. Título.

CDD: 629.8

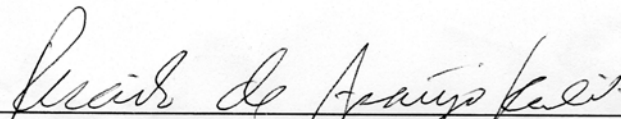
---

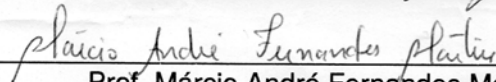
**SINTONIA ÓTIMA DE CONTROLADORES MPC CONSIDERANDO  
INCERTEZAS DE MODELAGEM**

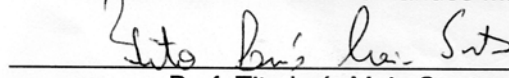
**Gesner Andrade Nery Júnior**

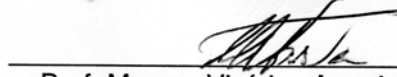
DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA INDUSTRIAL DA UNIVERSIDADE FEDERAL DA BAHIA COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA INDUSTRIAL.

Aprovada<sup>1</sup> por:

  
Prof. Ricardo de Araújo Kalid, D. Sc.

  
Prof. Márcio André Fernandes Martins, D. Sc.

  
Prof. Tito Luís Maia Santos, D. Sc.

  
Prof. Marcus Vinícius Americano da Costa Filho, D. Sc.

Salvador, BA - BRASIL  
Janeiro/2015

<sup>1</sup> Conforme resolução do Programa, o conjunto de orientadores teve a representação de 1 (um) único voto no parecer final da banca examinadora.

*Dedico à minha mãe  
e ao meu pai, que me educaram.*

*Dedico aos meus familiares  
e amigos, que me suportaram.*

*Dedico à minha amada  
e a todos que me amaram...*

---

# Agradecimentos

---

Agradeço aos meus pais que me criaram com muito amor e me deram educação, para que eu pudesse ser independente e trilhasse meus próprios caminhos. Agradeço ao meu irmão, que sempre me perturbou, mas eu sei que é porque você me ama. Eu amo muito vocês.

A minha amada, Nathália, que me reconfortou nesses momentos em que estou distante de casa, me amando e sendo muito carinhosa. Por tantas vezes estive cansado e você me deu motivação para recomeçar a escrever a dissertação. Te amo, meu amor.

Ao meu primo-irmão Deivide, que me confronta com várias questões, fazendo com que eu repense o meu modo de ver o mundo. Te amo.

Agradeço aos meus amigos do colégio que contribuíram para a construção do meu caráter: Blener, Mauro e Franklin. Obrigado! Parte do que sou, devo muito a vocês!

Quero agradecer aos meus amigos da faculdade: Géba, Giga, Guilherme e Léo pelas brincadeiras, discussões, festas e por me aturarem, eu sei que dei muito trabalho a vocês eheh.

Agradeço enormemente ao suporte e confiança conferidos a mim pelo Professor Kalid, que me acompanha desde os tempos de graduação e na iniciação científica, acreditando que eu poderia terminar o mestrado mesmo a distância. Sem a sua ajuda nada disso seria possível. Estou eternamente em débito, muito obrigado!

Gostaria de agradecer também a Márcio que revisou o artigo, deu sugestões para melhoria do trabalho e tornou-se co-autor. Somente com sua ajuda o artigo pôde ser publicado numa revista internacional. Muito obrigado.

Gostaria de agradecer a todo o pessoal que frequentava os laboratórios do PEI:

Reiner, Marcos, Raony, Chico, Robson, Érbert, Daniel, Isabel e Carol. As tardes nos laboratórios ficaram menos chatas com a presença de vocês, as conversas, pedaladas, churrascos, comidinhas low-carb...

Agradeço também a todos os meus familiares pelo carinho e amor, são tantos primos e primas, tios e tias, etc. que seria injusto citar apenas alguns. Amo todos vocês.

Agradeço a todos meus amigos, sem vocês a vida seria muito entediante!

Resumo da Dissertação apresentada à UFBA como parte dos requisitos necessários para a obtenção do grau de Mestre em Engenharia Industrial

Os métodos heurísticos de sintonia de controladores preditivos se mostram muito específicos, tanto em relação ao tipo de controlador, como ao tipo de sistema a ser controlado e, algumas vezes, não refletem com o desempenho ou a robustez desejados. Além disso, esses métodos não consideram a incerteza das medidas ou dos modelos, sendo que frequentemente a sintonia é feita na base da experiência do engenheiro, ou ainda, na falta desta, por tentativa e erro. Por outro lado, os métodos de sintonia ótima possuem a vantagem de serem bastante flexíveis, podendo ser utilizados para uma ampla gama de tipos de controladores e sistemas, e as sintonias resultantes da aplicação de tais métodos atendem o desempenho demandado pelo usuário, de acordo com o critério de desempenho ou função-objetivo previamente escolhidos, desde que sejam factíveis. O objetivo deste trabalho é desenvolver uma metodologia para a sintonia ótima de controladores preditivos multivariáveis, considerando a incerteza de modelagem nos parâmetros do modelo da planta. Uma vez que a formulação do problema de sintonia ótima resulta em uma programação mista-inteira não-linear, um algoritmo de otimização meta-heurístico, baseado na técnica de otimização por enxame de partículas, é utilizado para solucionar o problema proposto. Como forma de alcançar um controle também robusto às incertezas de modelagem, o método inclui na sua formulação a identificação do cenário do pior caso de controle, determinado no domínio da incerteza dos parâmetros do modelo, baseado no Índice de Resilência de Morari e no Número Condicional. Estudos de caso típicos da indústria de processos são realizados e as funções de densidade de probabilidade das funções-objetivo são analisadas, evidenciando o bom desempenho e robustez das sintonias propostas.

**Palavras Chaves:** Otimização matemática, controle preditivo e modelos matemáticos - processo estocástico.



Abstract of Dissertation presented to PEI/UFBA as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

The tuning guides in literature for model predictive controllers are very specific, both in relation to the type of controller, such as the type of system to be controlled and sometimes they do not lead the controllers to desired performance or robustness. Besides that, those tuning methods do not consider the measurements or models uncertainties and the tuning is frequently done on the basis of the experience of the engineer, or missing that, by trial and error. The optimum tuning methods have the advantage of being very flexible and can be used for a wide range of types of systems and controllers, and the tunings resulting from the application of such methods meet the performance demanded by the user, according to the performance criterion or objective function previously chosen by him, if they are feasible. The objective of this work is to develop a methodology for the optimal tuning of multivariable predictive controllers, considering the uncertainty in the parameters of the plant model. Once the optimal tuning problem formulation results in a nonlinear mixed integer programming, a meta-heuristic optimization algorithm based on the particle swarm optimization technique is used to solve the proposed problem. In order to achieve a robust control in respect to uncertainties of the model parameters, the method includes in its formulation the identification of the worst-case control scenario, determined in the field of uncertainty of the model parameters, based on the Morari Resilience Index and the Condition Number. Case studies are realized and the probability density functions of the objective functions are analyzed, showing good performance and robustness of the tunings parameters achieved.

**Keywords:** Mathematical optimization, model predictive control and mathematical models - stochastic processes.

---

# Sumário

---

<b>Lista de Figuras</b>	<b>xi</b>
<b>Lista de Tabelas</b>	<b>xiii</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Motivação e Justificativa . . . . .	1
1.2 Formulação do MPC e os aspectos da sintonia . . . . .	5
1.3 Objetivos desta dissertação . . . . .	14
1.4 Estrutura da dissertação . . . . .	15
1.5 Publicação . . . . .	16
<b>2 Conceitos e Método Proposto</b>	<b>17</b>
2.1 Conceitos Gerais . . . . .	17
2.1.1 Funções de transferência . . . . .	17
2.1.2 Espaço de estados . . . . .	19
2.1.3 Critérios de desempenho de controle . . . . .	19
2.1.4 Decomposição em valores singulares . . . . .	21
2.1.5 Número condicional . . . . .	22
2.1.6 Índice de Resiliência de Morari . . . . .	24
2.1.7 Otimização por enxame de partículas . . . . .	25
2.2 O método proposto . . . . .	28

2.2.1	O modelo com controle mais difícil . . . . .	28
2.3	Critérios de desempenho escolhidos . . . . .	30
2.3.1	Configuração do algoritmo . . . . .	31
2.4	Conclusões . . . . .	34
<b>3</b>	<b>Estudos de caso</b>	<b>35</b>
3.1	<i>Shell Standard Control Problem - SSCP</i> . . . . .	35
3.2	Separador bifásico . . . . .	46
3.3	Conclusões . . . . .	56
<b>4</b>	<b>Considerações finais</b>	<b>57</b>
4.1	Contribuições . . . . .	57
4.2	Trabalhos futuros . . . . .	58
<b>A</b>	<b>Códigos desenvolvidos</b>	<b>59</b>
A.1	Código da rotina <i>psoconsmix</i> . . . . .	59
A.2	Código da rotina <i>mincons</i> . . . . .	71
	<b>Referências Bibliográficas</b>	<b>87</b>

---

# Lista de Figuras

---

1.1	Um instante de ação do MPC para um caso SISO. . . . .	4
2.1	Movimentação das partículas sobre a região de busca em uma iteração da otimização realizada pelo PSO. . . . .	26
2.2	Diagrama do conjunto de modelos que formam $\mathbf{G}(s)$ , considerando a magnitude das incertezas de modelagem. . . . .	28
2.3	Fluxograma do método proposto. . . . .	32
3.1	Coluna de destilação do SSCP. . . . .	36
3.2	Variação do CN com a frequência para o modelo do SSCP. . . . .	38
3.3	Variação do MRI com a frequência para o modelo do SSCP. . . . .	38
3.4	Comportamento dinâmico de $Y_1$ no Caso 1 do SSCP. . . . .	40
3.5	Comportamento dinâmico de $Y_2$ no Caso 1 do SSCP. . . . .	41
3.6	Comportamento dinâmico de $U_1$ no Caso 1 do SSCP. . . . .	41
3.7	Comportamento dinâmico de $U_2$ no Caso 1 do SSCP. . . . .	42
3.8	Comportamento dinâmico de $y_1$ no Caso 2 do SSCP. . . . .	43
3.9	Comportamento dinâmico de $y_2$ no Caso 2 do SSCP. . . . .	43
3.10	Comportamento dinâmico de $u_1$ no Caso 2 do SSCP. . . . .	44
3.11	Comportamento dinâmico de $u_2$ no Caso 2 do SSCP. . . . .	44
3.12	Histograma dos valores da função-objetivo $\Phi$ do Caso 1 do SSCP. . .	46
3.13	Esquema de um separador bifásico de óleo e gás com controles de pressão e nível. . . . .	47

3.14	Comportamento dinâmico de $y_1$ para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência. . . . .	52
3.15	Comportamento dinâmico de $y_2$ para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência. . . . .	53
3.16	Comportamento dinâmico de $u_1$ para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência. . . . .	53
3.17	Comportamento dinâmico de $u_2$ para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência. . . . .	54
3.18	Comportamento dinâmico das vazões de líquido na entrada e na saída do separador, utilizando a sintonia ótima obtida e a de referência. . .	55
3.19	Histograma representando a PDF aproximada da função-objetivo $\Phi$ do separador de produção usando a sintonia ótima. . . . .	56

---

# Lista de Tabelas

---

3.1	Limites dos valores dos parâmetros . . . . .	37
3.2	Região de busca dos parâmetros sintonizáveis do MPC para o SSCP. .	39
3.3	Restrições rígidas das MVs e PVs para o Caso 1 do SSCP. . . . .	39
3.4	Restrições rígidas das MVs e PVs para o Caso 2 do SSCP. . . . .	40
3.5	Parâmetros ótimos encontrados pelo PSO para os casos do SSCP. . .	40
3.6	Valores da função-objetivo para os cenários de com a planta nominal e com a planta menos controlável, para os Caso 1 e 2. . . . .	45
3.7	Restrições rígidas das MVs e PVs, valores nominais dos distúrbios e SPs. . . . .	47
3.8	Região de busca dos parâmetros sintonizáveis do MPC. . . . .	49
3.9	Limites dos valores dos parâmetros do modelo. . . . .	51
3.10	Parâmetros de sintonia ótimos e de referência. . . . .	52

---

# Capítulo 1

## Introdução

---

*“Deus disse: “Faça-se a luz!”. E a luz foi feita.”*

Gênesis 1,3

### 1.1 Motivação e Justificativa

Ainda há, nos dias de hoje, um predomínio da utilização de controladores monovariáveis, mais especificamente controladores que são alguma combinação dos tipos proporcional, integral e derivativo (os controladores tipo PID - *Proportional, Integral and Derivative*). A implementação desses controladores na indústria é relativamente simples, rápida e barata, contudo, geralmente, não apresentam o melhor desempenho operacional possível. De fato, Åström e Hägglund (1995) afirmam que mais de 95% dos controladores são do tipo PID, e desses, a maioria é na verdade PI. Apesar da data da publicação, seguramente, não houve grandes mudanças nessa estatística. Os processos químicos são inerentemente multivariáveis e para controlá-los com controladores PID, comumente utiliza-se a estratégia conhecida como controle multivariável descentralizado. Nesta estratégia, o processo é tratado como vários sistemas de simples entrada e simples saída, denominados SISO (*Single Input Single Output*), nos quais são criadas várias malhas de controle independentes (do ponto de vista do controle, pois as variáveis muitas vezes são interdependentes). Contudo, devido às interações entre as variáveis do processo, há necessariamente um acoplamento entre as malhas de controle e, como consequência, existe a possibilidade de o controle ter um desempenho e robustez abaixo dos desejados, podendo levar algumas vezes a operação em malha aberta. As técnicas de desacoplamento

existem para solucionar esse tipo de questão, apesar de tornar o sistema de controle ainda mais complexo e trabalhoso, pois acrescenta-se mais uma etapa no projeto do controlador.

Sob um viés teórico diferente, o MPC (Controle Preditivo baseado em Modelo - *Model Predictive Control*) é uma técnica de controle que pode ser utilizada sem maiores complicações em sistemas multivariáveis, também conhecidos como MIMO (*Multiple Input Multiple Output*). Este é um tipo de controlador que tem sido extensivamente estudado por pesquisadores da academia e engenheiros da indústria de processos, e.g. refinarias e petroquímicas - essas caracterizando-se como o campo mais fértil de aplicação desta estratégia avançada de controle. Um dos principais motivos para o sucesso do MPC na indústria é que ele pode lidar naturalmente com restrições de processo. O interesse neste aspecto é que a operação do sistema com *set-points* próximos a restrições pode levar a ganhos econômicos, eficiência, diminuição nos impactos ambientais e aumentar a segurança. Além da incorporação explícita das restrições de processo e de sua lei multivariável, outras razões para o sucesso do MPC se deve pelo fato de: incorporar um modelo dinâmico do processo, que permite considerar o efeito futuro das variáveis manipuladas sob as controladas, mesmo na presença de longos tempos mortos; tratar com flexibilidade diferentes critérios de desempenho e metas econômicas do processo; e a sua ideia básica de funcionamento ser de fácil entendimento (MACIEJOWSKI, 2000). Por fim, ao contrário do que ocorre nos controladores PID, no MPC não existe a necessidade do emparelhamento entre as variáveis de processo (PV - *Process Variable*) e as variáveis manipuladas (MV - *Manipulated Variable*), isto é, não é preciso definir qual MV controlará uma PV específica. Dessa maneira, o MPC dispensa esta etapa no projeto do sistema de controle o que facilita a sua implementação e anula a possibilidade de um emparelhamento ruim.

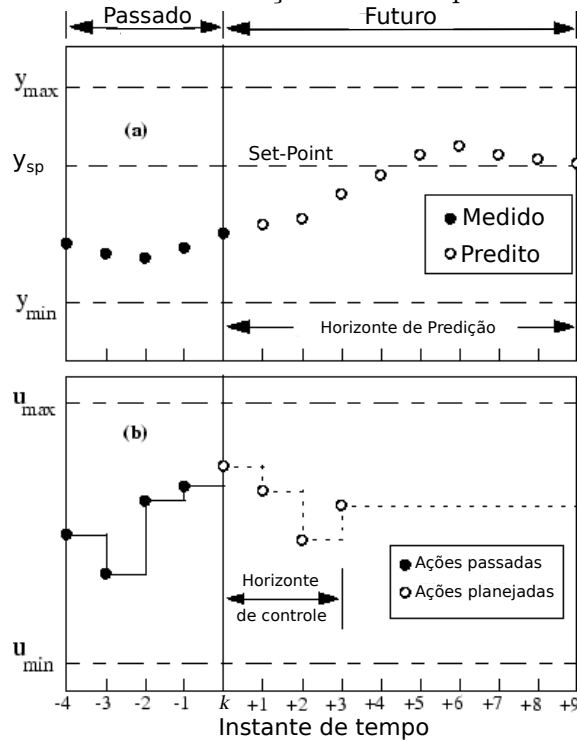
O termo MPC abrange uma família de controladores que utilizam o princípio do horizonte deslizante (ou móvel), que será apresentado com mais detalhes adiante, e um modelo dinâmico do sistema a ser controlado para predições. Existem várias categorias de MPC que se diferem, principalmente, no que se refere ao tipo de modelo usado na predição. Como exemplos de MPC, tem-se o DMC (*Dynamic Matrix Control*) que utiliza uma matriz dinâmica da resposta ao degrau do sistema;



já o GPC (*Generalized Predictive Control*) utiliza um modelo CARIMA (*Controlled Auto-Regressive Integrated Moving-Average*), e assim por diante. Além disso, invariavelmente, todos os controladores MPC calculam os valores das MVs por meio da solução de um problema de otimização (lei de controle) no qual são incorporadas as informações como: os SPs (*Set-Points*), as restrições, o valor das variáveis de saída preditas do sistema, o modelo do processo, entre outros dados.

O MPC é uma técnica de controle que combina a retroalimentação com o comportamento previsto das PVs, por um modelo dinâmico do processo, frente às perturbações ocasionadas pelas MVs e por distúrbios medidos. Em cada instante, são realizadas as medições das PVs e o MPC resolve um problema de otimização para obter o conjunto de ações de cada MV, desde o instante atual até alguns instantes no futuro (o horizonte de predição). No cálculo dos movimentos das MVs há graus de liberdade que permitem construir um problema de otimização, cuja resolução resultam nas ações a serem realizadas nas MVs. A quantidade de ações a serem realizadas por uma MV é igual ao configurado pelo parâmetro de sintonia denominado horizonte de controle. A partir do modelo do processo, o MPC prevê como as PVs se movimentam no futuro até o horizonte de predição, atendendo as restrições impostas, ao mesmo tempo que minimizam o valor da função-objetivo (lei de controle). O conjunto de ações que minimiza uma função-objetivo arbitrária é a escolhida. Contudo, o MPC somente envia para o processo o conjunto de ações calculados para o primeiro instante de tempo, descartando as outras ações, sendo que no momento seguinte todo o procedimento é realizado novamente, com um horizonte de predição que se move a cada instante (*receding horizon*). A Figura 1.1 mostra um instante de ação do MPC para um caso SISO, no qual podem ser observadas as faixas de tempo correspondentes aos horizontes de predição e de controle, as ações passadas e planejadas do MPC para a MV e as medições passadas da PV, bem como as suas predições.

Figura 1.1: Um instante de ação do MPC para um caso SISO.



Fonte: Adaptado do *Toolbox* de MPC do MATLAB<sup>®</sup>.

A ideia de utilizar um modelo do sistema para prever as suas condições futuras e controlá-lo, levando-se em consideração essas previsões, não é recente. Esse conceito, por exemplo, é utilizado no preditor de Smith (SMITH, 1957) para controlar plantas com tempos mortos elevados. Contudo, o preditor de Smith é um compensador de tempo morto e, ao contrário do MPC, não faz otimização dentro de um horizonte de predição móvel. Os modernos controladores MPC tiveram sua origem na indústria, principalmente, a partir do desenvolvimento da tecnologia por dois grupos independentes nos anos 70. Richalet et al. (1978) desenvolveram para a empresa ADERSA o MPHC (*Model Predictive Heuristic Control*), comercializado com o nome de IDCOM, que utiliza um modelo de resposta ao impulso da planta. Por sua vez, Cutler e Ramaker (1979) desenvolveram o DMC, com modelos de resposta ao degrau, para a companhia petrolífera Shell, a qual utilizou a tecnologia com êxito em suas refinarias. Esse fato incentivou bastante a pesquisa e o desenvolvimento do MPC em todo o mundo. Posteriormente, Clarke, Mohtadi e Tuffs (1987) apresentaram o GPC que, à época, possuía o diferencial de ser capaz de controlar, em um único algoritmo de MPC, processos com parâmetros variáveis, tempos-mortos variá-

veis, de fase não mínima, etc. A utilização de modelos em espaço de estados com o MPC foi realizada por Marquis e Broustail (1988), sendo adequada para representação de vários tipos de processos como: os multivariáveis, instáveis, integradores, de fase não mínima, etc. No presente trabalho, apesar de os modelos de processos serem apresentados em termos de funções de transferência contínuas, dada a relativa facilidade de compreensão da relação dos parâmetros dos modelos com a realidade física, o MPC utilizado aqui usa a representação convertida em espaço de estados discreta.

Apesar do sucesso do MPC, os controladores preditivos possuem parâmetros que precisam ser sintonizados para que o controle tenha um desempenho satisfatório, haja vista que uma sintonia ruim pode ocasionar até a instabilidade do sistema. Mesmo com os avanços alcançados pela técnica do MPC, desde a sua introdução nos anos 70, a sua sintonia ainda é uma tarefa que exige a experiência do engenheiro de controle e muitas vezes se torna um procedimento de tentativa e erro. Para entender melhor o problema de sintonia de MPC, a sua formulação e os aspectos concernentes a isso são explanados detalhadamente na próxima seção.

## 1.2 Formulação do MPC e os aspectos da sintonia

O MPC é um problema de otimização definido pela função-objetivo que atende às necessidades do processo, podendo ter restrições rígidas nos limites das variáveis de saída, nas variáveis manipuladas e nas suas taxas de variação e, ainda, em variáveis auxiliares, de modo que o desempenho seja adequado ao processo. Para um problema com  $n_u$  variáveis manipuladas e  $n_y$  variáveis controladas, o problema de otimização apresentado pelas Equações 1.1 até 1.7 é conhecida como a lei de controle do MPC:

$$\begin{aligned} \arg \min_{\substack{\Delta \hat{\mathbf{u}}(k|k), \dots, \\ \Delta \hat{\mathbf{u}}(k+H_c-1|k)}} J = & \sum_{j=1}^{H_p} \|\hat{\mathbf{y}}(k+j|k) - \mathbf{y}_{\text{sp},k}\|_{\mathbf{Q}}^2 \\ & + \sum_{j=0}^{H_c-1} \|\Delta \hat{\mathbf{u}}(k+j|k)\|_{\mathbf{R}}^2 \end{aligned} \quad (1.1)$$

$$+ \sum_{j=0}^{H_c-1} \|\hat{\mathbf{u}}(k+j|k) - \mathbf{u}_{\text{trg},k}\|_{\mathbf{S}}^2,$$

sujeito a:

$$\hat{\mathbf{x}}(k+1) = f(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k), \mathbf{w}(k)), \quad (1.2)$$

$$\hat{\mathbf{y}}(k) = g(\hat{\mathbf{x}}(k), \hat{\mathbf{u}}(k)), \quad (1.3)$$

$$\Delta \hat{\mathbf{u}}(k+j|k) \in \mathbf{L}, j = 0, 1, \dots, H_c - 1 \quad (1.4)$$

$$\mathbf{L} = \begin{cases} -\Delta \mathbf{u}_{\max}(k) \leq \Delta \hat{\mathbf{u}}(k+j|k) \leq \Delta \mathbf{u}_{\max}(k), \\ \mathbf{u}_{\min}(k) \leq \mathbf{u}(k-1) + \sum_{i=0}^j \Delta \hat{\mathbf{u}}(k+i|k) \leq \mathbf{u}_{\max}(k), \end{cases} \quad (1.5)$$

$$\Delta \hat{\mathbf{u}}(k+j|k) = 0, H_c \leq j \leq H_p - 1 \quad (1.6)$$

$$\mathbf{y}_{\min}(k) \leq \hat{\mathbf{y}}(k+j|k) \leq \mathbf{y}_{\max}(k), j = 1, 2, \dots, H_p, \quad (1.7)$$

em que  $k$  é um instante de tempo discreto; as Equações 1.2 e 1.3 representam um modelo genérico discreto do processo em que  $\hat{\mathbf{y}}$ ,  $\hat{\mathbf{u}}$ ,  $\hat{\mathbf{x}}$  e  $\mathbf{w}$  são respectivamente as variáveis de saída, de entrada, de estado do sistema e os distúrbios do sistema;  $H_p$  e  $H_c$  são os horizonte de predição e horizonte de controle, respectivamente;  $\Delta \hat{\mathbf{u}}(k+j|k) \in \mathbb{R}^{n_u}$  são as ações de controle calculadas no instante atual  $k$ , mas que serão realizadas somente no instante futuro  $k+j$ ;  $\hat{\mathbf{y}}(k+j|k) \in \mathbb{R}^{n_y}$  são as predições das variáveis de saídas calculadas no instante atual  $k$ , obtidas a partir do modelo do processo, para instantes de tempo no futuro  $k+j$ , em que também se considera os efeitos das ações futuras que seriam realizadas pelo MPC;  $\mathbf{y}_{\text{sp},k} \in \mathbb{R}^{n_y}$  são os SPs desejados para as PVs, no instante  $k$ ;  $\mathbf{u}_{\text{trg},k} \in \mathbb{R}^{n_u}$  são os *targets* das MVs, no instante  $k$ , representando, por exemplo, valores a serem perseguidos pelas MVs devido a restrição de operacionais, de segurança ou ambientais;  $\mathbf{Q}$ ,  $\mathbf{R}$  e  $\mathbf{S}$  são matrizes diagonais, com dimensões apropriadas, e denotam os pesos dados na função-objetivo para as PVs, variações das MVs e as próprias MVs e cujos elementos são denotados por  $q_i$ ,  $r_i$  e  $s_i$ , respectivamente. As inequações 1.5 e 1.7 correspondem

às restrições rígidas desejadas para as MVs, variações das MVs e PVs. Apesar da lei de controle apresentada ser bastante generalizada, dada a grande liberdade de formulação de controladores MPC, existem controladores que seguem regras de controle diferentes das mostradas acima como, por exemplo, o IHMPC (*Infinite Horizon Model Predictive Control*) (RAWLINGS; MUSKE, 1993), o controle por faixas (FERRAMOSCA et al., 2010), entre outros.

Na lei de controle apresentada aparecem parâmetros que determinam o desempenho e robustez do controlador, os quais podem ser ajustados pelo engenheiro de controle. Esses termos são  $H_p$ ,  $H_c$ ,  $\mathbf{Q}$ ,  $\mathbf{R}$  e  $\mathbf{S}$  e são conhecidos como os parâmetros de sintonia do controlador. Apesar de ter sido dito que as matrizes  $\mathbf{Q}$ ,  $\mathbf{R}$  e  $\mathbf{S}$  são diagonais, rigorosamente, nada impede que elas sejam matrizes cheias e, até mesmo, que variem com o tempo. Contudo, isso implica em mais parâmetros a serem sintonizados, tornando o procedimento de sintonia ainda mais complexo, sendo que os ganhos podem não justificar o trabalho adicional. Além desses parâmetros, o DMC possui um parâmetro a mais a ser ajustado que é o horizonte do modelo, que não aparece explicitamente no problema de controle. Para entender melhor o significado desses parâmetros, seguem alguns esclarecimentos sobre como eles interferem no comportamento em malha fechada do sistema de controle:

- O horizonte de predição  $H_p$  é um número inteiro e especifica a quantidade de instantes de tempo que o MPC possui para levar as variáveis controladas até seus valores de referência. Valores grandes de  $H_p$  geralmente melhoram a estabilidade e a robustez, todavia também aumentam o esforço computacional. Wojsznis et al. (2003) recomendam que  $H_p$  seja aumentado até um valor que incrementos adicionais não apresentem ganhos significativos. Já Qi e Fisher (1994) reportam, em seu artigo, casos em que um aumento no  $H_p$  podem piorar a robustez do controlador. No controlador IHMPC o  $H_p$  tende ao infinito, isto é,  $H_p \rightarrow \infty$  e dessa maneira o parâmetro já está pré-definido e não é sintonizável.
- $H_c$  é outro parâmetro inteiro de sintonia que designa a quantidade de instantes de tempo nos quais o MPC pode gerar ações nas MVs, depois disso o MPC não calcula novos movimentos das MVs, permanecendo igual ao último valor

calculado. É sabido que valores altos de  $H_c$  também aumentam o esforço computacional, e melhoram o desempenho do controle em sistemas com tempos mortos desconhecidos (FISHER, 1991).

- O horizonte do modelo ( $H_n$ ), o qual não aparece na lei de controle apresentada acima, é um parâmetro inteiro de sintonia próprio dos controladores DMC e MPHC, que utilizam modelos de resposta ao degrau e ao impulso, respectivamente. Ele corresponde à quantidade de pontos utilizados para representar o modelo do sistema. Valores pequenos podem causar instabilidade, já valores altos aumentam a robustez mas demandam, em contrapartida, um esforço computacional maior.
- $\mathbf{Q}$  é uma matriz cujos elementos são números contínuos os quais determinam a importância das variáveis controladas na função-objetivo apresentada na Equação 1.1. Em geral, usam-se valores mais altos para as variáveis que devem possuir um controle mais próximo dos *set-points* por motivos operacionais, de segurança ou ambientais, por exemplo.
- Os elementos da matriz  $\mathbf{R}$  também são contínuos mas, por sua vez, atuam sobre as variações nas variáveis manipuladas, sendo também conhecidos como fatores de supressão de movimento. O ajuste desses parâmetros definem o comportamento das MVs quanto à agressividade e oscilações. Valores altos, em geral, aumentam a robustez mas, em compensação, podem deixar o controle lento, já que penalizam mais os movimentos nas MVs.
- As penalizações das magnitudes das variáveis manipuladas são reguladas pelos elementos da matriz  $\mathbf{S}$ . Uma maneira de remover as restrições das MVs no problema de controle, de modo a facilitar a sua solução matemática, é sintonizando, adequadamente, esses parâmetros. Garriga e Soroush (2008) mostraram que no limite em que esses parâmetros tendem ao infinito, o MPC se comporta como se estivesse em malha aberta.

Além desses parâmetros próprios do MPC, o tempo de amostragem  $T$  (ou intervalo de controle) também afeta a ação do controlador e já foi alvo de estudos de sintonia ótima de MPC (DAVTYAN; HOFFMANN; SCHEURING, 2011). Em

um processo de sintonia, ao se regular a frequência de amostragem deve-se reajustar também o  $H_p$  e o  $H_c$  (e o horizonte do modelo, no caso do DMC e MPHC), pois são parâmetros especificados em intervalos de tempo e a não observância desse aspecto gera resultados indesejáveis (SHRIDHAR; COOPER, 1998). Como regra, a frequência de amostragem (que é o inverso de  $T$ ) não pode ser baixa demais pois pode não reter as características da dinâmica do processo, tampouco também não pode ser alta demais pois os parâmetros  $H_p$ ,  $H_c$  e o horizonte do modelo também terão valores altos, causando os problemas já discutidos. Os *set-points*  $\mathbf{y}_{sp,k}$  não são entendidos como parâmetros de sintonia do MPC, pois, apesar de serem ajustáveis e afetarem o controle, são definidos no projeto dos processos a partir de estudos de otimização da operação, levando-se em conta também outros aspectos como a segurança. Além disso, os controladores MPC podem estar hierarquicamente abaixo, numa estrutura de controle, de uma otimização em tempo real (RTO - *Real-Time Optimization*), recebendo continuamente novos SPs de forma a otimizar o processo. Em outras formulações de MPC aparecem ainda as variáveis de folga (*slack variables*)(SANTORO; ODLOAK, 2012), que têm a finalidade de permitir que soluções do problema de otimização sejam encontradas mais facilmente pois suavizam as restrições, mas acompanhando essas variáveis são introduzidos os seus fatores de ponderação, que também são parâmetros a serem sintonizados. Segundo Badgwell (apud GARRIGA; SOROUSH, 2010), a primeira ação de "sintonia" de um MPC é a etapa de identificação do modelo do processo, pois caso o modelo seja suficientemente adequado todo o processo de sintonia restante é facilitado. No presente trabalho, essa etapa de modelagem não será abordada e supõe-se que os modelos, incluindo as incertezas de modelagem, representam adequadamente os comportamentos dinâmicos do processo. Em seus artigos, Garriga e Soroush (2010) e Rani e Unbehauen (1997) mostram uma grande variedade de métodos para a sintonia de MPC e comparam alguns deles, fazendo estudos de caso.

Lee e Yu (1994) sintonizaram controladores MPC baseados em modelos em espaços de estados, com filtros nos sinais dos distúrbios, assumindo que o modelo era estável em malha aberta. Eles recomendam que primeiro a sintonia esteja voltada para a estabilidade nominal do sistema e, conseguido isso, pode-se partir para a sintonia robusta. Em seu artigo, a estabilidade nominal é obtida primeiramente

fazendo  $\mathbf{R} = 0$ , em seguida, o  $H_c$  é ajustado para o maior valor possível dentro dos limites computacionais e para os sistemas que possuem zeros estáveis faz-se  $H_p = H_c$ , caso contrário é sugerido que se utilize o IHMPC. Uma vez conseguida a estabilidade nominal, o próximo passo é resintonizar o MPC para robustez, em que os valores de  $H_p$  e  $H_c$  são mantidos e ajustes são feitos em  $\mathbf{R}$  e nos parâmetros dos filtros utilizados em sua abordagem, até que se obtenha parâmetros adequados. Esse método é somente aplicável para os sistemas estáveis em malha aberta e, da mesma forma que o método mencionado anteriormente, para que se obtenha o desempenho de controle desejado os coeficientes de supressão de movimentos são reajustados por tentativa e erro, o que consome tempo e levam a parâmetros subótimos.

Shridhar e Cooper (1998) propuseram sintonizar um controlador DMC aproximando a dinâmica do processo de todos os pares MV e PV em modelos FOPDT (*First Order Plus Dead Time*). Os parâmetros de sintonia são então calculados a partir dos parâmetros desses modelos e, ao final do processo, os autores orientam que um ajuste fino deva ser feito nos coeficientes de supressão de movimento com o propósito de se obter o desempenho de controle requerido, não havendo uma orientação clara de como regulá-los a fim de que isso seja alcançado, o que pode demandar tempo. Além disso, também não existem instruções de como sintonizar o DMC caso existam restrições a serem obedecidas.

Uma abordagem no domínio da frequência para sintonia de GPC foi apresentada por Sarhadi, Salahshoor e Khaki-Sedigh (2012). O método é somente aplicável ao caso SISO sem restrições e pretende conseguir os parâmetros de sintonia que tornam o controle robusto, com a atenuação dos distúrbios numa determinada faixa de frequência, fazendo uma análise da função de sensibilidade do sistema, em que a influência dos parâmetros é estudada separadamente. A partir da observação dos efeitos de cada parâmetro a sintonia é ajustada até que se obtenha a robustez desejada. Contudo, os problemas industriais de controle são multivariáveis e possuem restrições, portanto esse método de sintonia é pouco aplicável na prática.

Reverter, Ibarrola e Cano-Izquierdo (2014) empregaram o método do lugar das raízes para sintonizar um DMC sem restrições para controlar um sistema SISO aproximado por um modelo FOPDT, em que foram investigados os efeitos de  $H_p$ ,



$H_c$ ,  $H_n$ ,  $\mathbf{R}$  e o intervalo de controle nos pólos do sistema. Ao final da pesquisa produziram algumas regras heurísticas para a escolha adequada desses parâmetros, como por exemplo escolher o valor de  $H_p$  igual a constante de tempo do modelo e o intervalo de controle ser um décimo desse valor. Pelos mesmos motivos, as limitações de aplicação deste método se assemelham ao anterior, restringindo-se a poucos casos na indústria.

Como se observa, esses primeiros trabalhos se baseiam em heurísticas e possuem o mérito de serem viabilizadores do funcionamento do MPC, isto é, caso seja necessário o comissionamento do controlador, sem que o seu desempenho seja otimizado, pode-se recorrer a esses métodos para tal finalidade. Além disso, essas regras norteiam os métodos baseados em otimização e são utilizadas muitas vezes como referência. Porém, eles não garantem que a sintonia obtida é a que irá apresentar o desempenho de controle demandado e, para isso, o MPC necessita ser ressintonizado, consumindo tempo.

Um método de sintonia *on-line* de  $\mathbf{Q}$  e  $\mathbf{R}$  de controladores DMC é proposto por Al-Ghazzawi et al. (2001), no qual são obtidas funções linearizadas da relação entre a resposta prevista em malha fechada e os parâmetros de sintonia. Na estratégia,  $H_p$  e  $H_c$  são predeterminados por algum outro método e as funções linearizadas são utilizadas para elaborar expressões analíticas das funções de sensibilidade do sistema em malha fechada. As funções de sensibilidade são incorporadas em um problema de mínimos quadrados com restrições simplificado, para que a sua solução seja rápida, possibilitando a sua implementação em sintonias *on-line*.

Um controlador SISO de temperatura, do tipo GPC, de um bioreator para produção de *S. Cerevisiae* em batelada foi sintonizado por Bursali et al. (2001). Eles empregaram um método de sintonia ótima para obter o valor do intervalo de controle e o coeficiente de supressão de movimento que minimizam o ISE (*Integral Square Error*), enquanto que o restante dos parâmetros foram previamente definidos.

Ali (2003) adotou lógica *fuzzy* com o intuito de sintonizar *on-line* o  $H_p$ ,  $\mathbf{Q}$  e  $\mathbf{R}$  de controladores MPC não lineares, empregando como base para isso as regras heurísticas consolidadas na literatura. Para a obtenção de  $H_p$  ele utilizou arredondamento matemático em seu algoritmo, que é totalmente concebido com variáveis contínuas.

Han, Zhao e Qian (2006) utilizaram o PSO (*Particle Swarm Optimization*) para sintonizar somente controladores DMC sem restrições, numa abordagem que considera a incerteza nos parâmetros do modelo num problema do tipo Min-Max.

Lee, Svrcek e Young (2008) utilizaram algoritmo genético e lógica *fuzzy* para sintonizar controladores DMC sem restrições, no qual um critério multi-objetivo que envolvia ISE, *overshoot* e o tempo de assentamento foi projetado para a escolha de  $\mathbf{Q}$  e  $\mathbf{R}$  ótimos. Os parâmetros de sintonia inteiros como o  $H_p$ ,  $H_c$  e o  $H_n$  não são sintonizados pelo método, que também não considera as restrições das variáveis.

Uma sintonia ótima com a finalidade de aumentar o desempenho econômico de controladores MPC com restrições foi proposto por Lee, Huang e Tamayo (2008). Em seu artigo, a lei de controle do MPC foi ligeiramente modificada, contando com termos a serem sintonizados que refletiam a importância econômica das PVs e MVs. Outros parâmetros que também foram alvo de ajuste foram as variáveis de relaxamento das restrições. Contudo, mais uma vez, os parâmetros de sintonia inteiros não são sintonizáveis pelo método.

Bagheri e Khaki-Sedigh (2014) fixou o valor de  $H_c$  em 1 e, a partir de uma otimização, obtiveram expressões analíticas com o intuito de sintonizar o  $\mathbf{Q}$  e o  $\mathbf{R}$  de controladores MPC baseado em espaço de estados, para um desempenho de controle desejado. Em sua abordagem os sistemas MIMO considerados eram quadrados e os modelos foram aproximados por FOPDT.

Uma proposta de sintonia ótima *on-line* de um MPC SISO não-linear foi feita por Oliveira et al. (2014). O controlador foi denominado pelos autores de SMGPC (*Sliding Mode Generalized Predictive Controller*), pois combina o GPC e o controle por modos deslizantes, e para a sua sintonia o PSO foi aplicado. Contudo, na verdade, o método não trata da sintonia do conjunto padrão de parâmetros do MPC, o qual é ajustado heurísticamente *a priori*; outrossim, o artigo se refere especificamente aos parâmetros de sintonia do controle por modos deslizantes.

Os métodos de sintonia ótima de MPC apresentam a clara vantagem de encontrar os parâmetros de sintonia que melhor se adequam às necessidades do usuário, ou pelo menos chegam próximos a isso. Porém, um problema que surge ao se fazer a sintonia ótima de controladores MPC, além da quantidade de parâmetros a serem

sintonizados, está relacionado à natureza dos parâmetros de sintonia, pois existem parâmetros discretos e contínuos.

Ao contrário do método proposto neste trabalho, observa-se que os outros métodos pesquisados possuem limitações que dizem respeito a um dos itens abaixo:

- O tipo de controlador.
- O tipo de sistema, se SISO ou MIMO.
- Os parâmetros de sintonia que podem ser ajustados.
- Se leva ou não em consideração a incerteza de modelagem.
- Se leva em consideração as restrições do sistema explicitamente.

Como se pode perceber, o MPC possui muitos parâmetros de sintonia, existindo até controvérsias na literatura sobre como eles influenciam no controle. Dessa maneira, às vezes, não se sabe ao certo no caso do aumento ou diminuição de um determinado parâmetro se o controlador atenderá as demandas de desempenho e robustez desejadas pelo usuário. Ainda menos claro é a sinergia entre os parâmetros, isto é, seus efeitos combinados. Diante disto, os métodos de sintonia ótima se mostram atraentes, pois exigem menos conhecimento sobre sintonia e geralmente demandam menos tempo, haja vista que consideram intrinsecamente a sinergia entre os parâmetros, pois eles são todos sintonizados concomitantemente e não há a necessidade de reajustá-los, para uma dada condição de sintonia.

Os métodos de sintonia ótima buscam os parâmetros de sintonia que maximizam ou minimizam um critério de desempenho de controle previamente estabelecido e, dessa maneira, pelo menos a performance do controlador fica assegurada. Matematicamente, os critérios de desempenho representam a função-objetivo do problema de otimização, que pode ser não-linear, com ou sem restrições, sendo na sua forma mais geral do tipo programação não-linear inteira mista (MINLP - *Mixed-Integer Nonlinear Programming*). Esse tipo de problema é de difícil resolução e possui uma quantidade reduzida de algoritmos que podem ser usados para encontrar a sua solução. Além disso, várias dessas técnicas de otimização fazem uma série de hipóteses

quanto a convexidade e a diferenciabilidade das funções-objetivo e das restrições, entre outras suposições, o que limita ainda mais a utilização de tais algoritmos. De maneira a evitar a resolução de MINLPs, alguns trabalhos na literatura sobre sintonia ótima de MPC, como os realizados por Suzuki et al. (2012) e Schwartz e Rivera (2006), especificam *a priori* alguns dos parâmetros de sintonia, como o horizonte de predição e o horizonte de controle, que são inteiros. Esse procedimento facilita o processo de sintonia pois o problema de otimização conta com menos variáveis de decisão e, principalmente, somente com variáveis de decisão contínuas, possibilitando a utilização de algoritmos mais simples, confiáveis e rápidos. Caso o leitor queira aprofundar-se mais nos algoritmos determinísticos de resolução de MINLP, Grossmann (2002) faz uma revisão dessas técnicas, mostrando os conceitos básicos e suas limitações. Por outro lado, neste trabalho emprega-se um algoritmo PSO que resolve problemas do tipo MINLP (DATTA; FIGUEIRA, 2011), como forma a superar as limitações dos métodos de otimização que utilizam gradiente.

O método de sintonia proposto no presente trabalho tenta manter um compromisso entre o desempenho e a robustez do controlador MPC multivariável na presença de incertezas e que seja o mais generalizado possível. Para isso, combina a utilização de parâmetros como o número condicional (CN - *Condition Number*) e o índice de resiliência de Morari (MRI - *Morari Resiliency Index*), e parâmetros que estimam o desempenho do controle como a integral do erro ao quadrado ponderado pelo tempo (ITSE - *Integral Time-weighted Squared Error*). O método de sintonia proposto considera, ainda, a incerteza nos valores dos parâmetros do modelo de predição utilizado pelo MPC. Observando a faixa de valores que os parâmetros do modelo podem assumir, define-se o conjunto de parâmetros que determina o modelo mais difícil de controlar, dado um critério escolhido arbitrariamente, o qual é utilizado como referência na sintonia.

### 1.3 Objetivos desta dissertação

O objetivo geral deste trabalho é desenvolver um método para a sintonia ótima de controladores MPC com restrições, que considere a incerteza nos parâmetros do

modelo do processo. Para tanto, as seguintes metas específicas devem ser cumpridas:

- Propor um critério de sintonia que leve em consideração tanto a robustez quanto o desempenho do controlador.
- Elaborar um método que encontre o modelo mais difícil de controlar dentro do domínio da incerteza de modelagem.
- Desenvolver um algoritmo PSO para otimização de MINLP.
- Aplicar o método de sintonia desenvolvido em diferentes sistemas multivariáveis.

## 1.4 Estrutura da dissertação

Esta dissertação está organizada em quatro capítulos, incluindo este capítulo introdutório, e estruturados da seguinte forma:

O capítulo 2 trata de conceitos fundamentais necessários para a compreensão do trabalho, mostrando os critérios de desempenho de controle, o algoritmo de enxame de partículas, os parâmetros de controlabilidade e os tipos de modelos empregados. Além disso, o algoritmo do método de sintonia proposto é apresentado, exibindo a sua estruturação, e também esclarece detalhadamente os parâmetros utilizados na configuração do enxame de partículas e como a incerteza de modelagem é levada em consideração na sintonia do MPC.

Os resultados desse trabalho estão presentes no capítulo 3, sobre o qual são realizados dois estudos de casos de processos típicos da indústria para testar o método e validá-lo.

No capítulo 4 são feitas as considerações finais do trabalho e as recomendações para trabalhos futuros.

Por fim, no Apêndice A são apresentados os códigos desenvolvidos em linguagem MATLAB<sup>®</sup> para a sintonia ótima dos controladores MPC deste trabalho.

## 1.5 Publicação

A elaboração desta pesquisa originou um artigo, o qual foi publicado em 2014 com o título "*A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty*", na revista ISA Transactions, volume 53, páginas 560 a 567, cujos autores são Gesner Andrade Nery Júnior, Ricardo de Araújo Kalid e Márcio André Fernandes Martins. No artigo, o método de sintonia ótima proposto aqui é apresentado e um estudo de caso foi realizado para comprovar a sua eficácia. Buscou-se na literatura um sistema linear, para os testes iniciais do método, e o *Shell Standard Control Problem* (SSCP) se mostrou uma opção adequada para isso. Foi apresentado o modo de como obter o sistema mais difícil de controlar, considerando a incerteza de modelagem, com métricas (MRI e CN) que indicam a controlabilidade. Os parâmetros escolhidos para a sintonia ótima foram o  $H_p$ ,  $H_c$ ,  $\mathbf{Q}$  e  $\mathbf{R}$ . Os resultados mostraram que o método possuía a capacidade de obter bons parâmetros de sintonia para controladores MPC com restrições.

---

# Capítulo 2

## Conceitos e Método Proposto

---

*“Ignotum per ignotius.”*

Expressão latina

Este capítulo tem a finalidade de revisar os conceitos necessários para o entendimento deste trabalho, de modo a facilitar a compreensão do método proposto, o qual é exposto ao final do capítulo. Caso o leitor deseje entender mais a respeito sobre os conceitos relacionados aqui e outros tópicos relacionados, recomenda-se a leitura dos livros de Ljung (1999), Camacho e Bordons (2004) e o artigo de Poli, Kennedy e Blackwell (2007).

### 2.1 Conceitos Gerais

#### 2.1.1 Funções de transferência

As funções de transferência são modelos matemáticos dinâmicos, que relacionam as entradas e saídas do sistema no domínio de Laplace, cujos parâmetros podem ter um significado físico. Aqui serão apresentados os dois tipos mais usuais e, talvez, os mais importantes, dado que muitos processos podem ser adequadamente representados por essas funções. Neste trabalho as variáveis dos modelos apresentados são variáveis de desvio, as quais estão denotadas por letras maiúsculas.

A FOPDT  $G(s)$  que relaciona a variável de saída  $Y(s)$  com a variável de entrada

$U(s)$  é definida como:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot e^{-\gamma \cdot s}}{\tau \cdot s + 1}, \quad (2.1)$$

em que  $K$  é o ganho estacionário do processo,  $\gamma$  é o tempo morto e  $\tau$  é a constante de tempo. Esses parâmetros têm significado físico: o ganho estacionário representa a razão entre o valor da variação entre dois estados estacionários diferentes da variável de saída  $y(t)$  e da variável de entrada  $u(t)$ , ou seja:

$$K = \frac{\Delta y}{\Delta u}. \quad (2.2)$$

O tempo morto é o tempo que a variável  $y(t)$  demora até começar a responder a uma excitação ocasionada pela entrada  $u(t)$ . A constante de tempo representa a "velocidade" de resposta de  $y(t)$  frente a uma excitação de  $u(t)$ , quanto maior for o  $\tau$  mais lenta é essa resposta.

Por sua vez, a função de transferência contínua de segunda ordem mais tempo morto (SOPDT - *Second Order Plus Dead Time*)  $G(s)$  que relaciona a variável de saída  $Y(s)$  com a variável de entrada  $U(s)$  é definida como:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot e^{-\gamma \cdot s}}{\tau^2 \cdot s^2 + 2 \cdot \tau \cdot \zeta \cdot s + 1}, \quad (2.3)$$

em que  $\zeta$  é o fator de amortecimento. O valor do fator de amortecimento determina se a resposta será oscilatória ou estável.

Essas funções de transferência podem ainda possuir zeros, que são raízes no numerador, e se apresentarem como:

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot (b_1 \cdot s + 1) \cdot e^{-\gamma \cdot s}}{\tau \cdot s + 1}, \quad (2.4)$$

$$G(s) = \frac{Y(s)}{U(s)} = \frac{K \cdot (b_2 \cdot s + 1) \cdot (b_1 \cdot s + 1) e^{-\gamma \cdot s}}{\tau^2 \cdot s^2 + 2 \cdot \tau \cdot \zeta \cdot s + 1}. \quad (2.5)$$

Uma matriz  $m \times n$  de funções de transferência de um sistema MIMO com  $n$  entradas e  $m$  saídas pode então ser definida como:

$$\mathbf{G}(s) = \begin{bmatrix} G(s)_{1,1} & \dots & G(s)_{1,n} \\ \vdots & \ddots & \vdots \\ G(s)_{m,1} & \dots & G(s)_{m,n} \end{bmatrix}. \quad (2.6)$$



A matriz  $\mathbf{G}(s)$  relaciona o vetor de variáveis de saída  $\mathbf{Y}(s)$  e o vetor de variáveis de entrada  $\mathbf{U}(s)$  através da equação:

$$\mathbf{Y}(s) = \mathbf{G}(s) \cdot \mathbf{U}(s) \Rightarrow \begin{bmatrix} Y(s)_1 \\ \vdots \\ Y(s)_m \end{bmatrix}_{1 \times m} = \begin{bmatrix} G(s)_{1,1} & \dots & G(s)_{1,n} \\ \vdots & \ddots & \vdots \\ G(s)_{m,1} & \dots & G(s)_{m,n} \end{bmatrix}_{m \times n} \cdot \begin{bmatrix} U(s)_1 \\ \vdots \\ U(s)_n \end{bmatrix}_{1 \times n} \quad (2.7)$$

Os modelos de processo apresentados anteriormente estão no domínio contínuo, ou seja, são funções no domínio da frequência da variável de Laplace  $s$ , contudo os computadores utilizam versões discretizadas equivalentes desses modelos, obtidas a partir de processos de discretização que não serão descritos aqui, mas que podem ser vistos em Aguirre (2007). A opção por mostrar os modelos dos processos na sua forma contínua foi feita porque esses tipos de modelos foram utilizados na identificação dos sistemas que serão apresentados nos estudos de caso.

### 2.1.2 Espaço de estados

O MPC programado neste trabalho utiliza o modelo de processo em espaço de estados, na sua forma discreta, linear e invariante no tempo. Esse tipo de representação de modelo é na seguinte forma matricial:

$$\begin{cases} \mathbf{x}(k+1) = \mathbf{A} \cdot \mathbf{x}(k) + \mathbf{B} \cdot \mathbf{u}(k) \\ \mathbf{y}(k) = \mathbf{C} \cdot \mathbf{x}(k) + \mathbf{D} \cdot \mathbf{u}(k) \end{cases}, \quad (2.8)$$

em que as matrizes  $\mathbf{A}$ ,  $\mathbf{B}$ ,  $\mathbf{C}$  e  $\mathbf{D}$  são invariantes no tempo,  $\mathbf{x}(k)$  é o vetor de estados,  $\mathbf{y}(k)$  é o vetor de variáveis de saída e  $\mathbf{u}(k)$  é o vetor das variáveis de entrada.

### 2.1.3 Critérios de desempenho de controle

Para comparar o desempenho de diferentes sintonias de um controlador, cujo sistema será submetido a um mesmo distúrbio ou mudanças de *set-point*, é possível idealizar uma função arbitrária que quantitativamente avalia o quão bom é aquela sintonia. Dessa forma, é possível saber se uma sintonia é melhor que a outra com uma simples comparação entre números. Essas funções são conhecidas como critérios de

desempenho de controle e devem ser elaboradas de forma a refletir o comportamento dinâmico desejado em malha fechada. Por exemplo, caso se deseje avaliar sintonias de acordo com o erro integral das PVs, o critério de ITSE (*Integral Time-weighted Squared Error*) poderia ser usado. Esses critérios são úteis também em métodos de sintonia ótima, pois nesse caso eles tomam o papel das funções-objetivo do problema de otimização e caso uma solução ótima seja encontrada, a sintonia deverá atender da melhor forma possível os requerimentos de desempenho do usuário.

Uma infinidade de critérios de desempenho podem ser concebidos de forma a comparar as diferentes propriedades da resposta de um sistema dinâmico, como o *overshoot*, o tempo de subida, tempo de acomodação, a variância de uma variável, etc. Por vezes, existem múltiplos objetivos que a sintonia deve alcançar e, assim, critérios de desempenho adequados para isso devem ser utilizados. Os critérios baseados na integral do erro são bastante difundidos, como o IAE (*Integral Absolute Error*), o ISE (*Integral Squared Error*), o ITAE (*Integral Time Absolute Error*) e o ITSE. Esses critérios consistem na diferença entre o valor da PV e seu SP (o erro  $e(t)$ ), neste trabalho é empregado a forma discretizada do ITSE:

$$\text{ITSE} = \sum_{k=0}^{\infty} k \cdot T \cdot e^2(k) \quad (2.9)$$

Existe também a possibilidade de escrever critérios de desempenho que medem o esforço do controlador para levar as PVs de volta para os seus SPs, o presente trabalho utiliza um critério análogo ao ITSE para isso:

$$\text{ITSVMV} = \sum_{k=0}^{\infty} k \cdot T \cdot (u(k+1) - u(k))^2 \quad (2.10)$$

Os critérios baseados na integral do erro e da variação das MVs podem ainda ser combinados em um somatório, gerando um critério de desempenho multi-objetivo, o qual é utilizado em um dos estudos de caso deste trabalho.

Como já foi dito, existe uma grande liberdade na escolha dos critérios. Por exemplo, Arruda et al. (2008) criaram um critério multi-objetivo que mensura o desempenho da sintonia de controladores PI/PID em uma estrutura de controle descentralizado para processos multivariáveis, o critério desenvolvido utiliza o ITSE e a variância nas PVs e MVs e usou um algoritmo genético para encontrar a sintonia

ótima, o critério está apresentado como segue:

$$\arg \min_{\mathbf{Kc}, \tau_i, \tau_D} J = \sqrt{\sum a_{j,0} \cdot f_j^2}, \quad (2.11)$$

em que,

$$f_j = a_{j,1} \cdot \text{ITSE}_j + a_{j,2} \cdot \sigma^2(y_j(t)) + a_{j,3} \cdot \sigma^2(u_j(t)), \quad (2.12)$$

$f_j$  é o critério de desempenho para uma malha de controle  $j$  em particular, os termos  $a_{j,0}$ ,  $a_{j,1}$ ,  $a_{j,2}$  e  $a_{j,3}$  são apenas constantes escolhidas arbitrariamente que ponderam e escalonam os termos do critério.  $\mathbf{Kc}$  é o conjunto dos ganhos dos controladores,  $\tau_i$  é o vetor de todos os tempos integrais e  $\tau_D$  dos tempos derivativos.  $\sigma^2(y_j(t))$  e  $\sigma^2(u_j(t))$  são as variâncias da PV e da MV. Schwartz e Rivera (2006) sintonizaram os fatores de supressão de movimento e os SPs de um MPC, maximizando uma função lucro na cadeia de suprimentos de uma empresa e usou um algoritmo de aproximação estocástica com perturbação simultânea para encontrar os parâmetros ótimos. Gaing (2004) criou uma função-objetivo para sintonizar PIDs que depende do *overshoot* ( $M_p$ ), do tempo de subida ( $t_r$ ), do tempo de acomodação ( $t_s$ ), do erro de estado estacionário (*off-set*, ( $E_{ss}$ )) e de um parâmetro  $\beta$  que pode variar entre 0,8 e 1,5, usando o PSO:

$$\arg \min_{\mathbf{Kc}, \tau_i, \tau_D} J = [(1 - e^{-\beta}) \cdot (M_p + E_{ss}) + e^{-\beta} \cdot (t_s - t_r)]^{-1}, \quad (2.13)$$

#### 2.1.4 Decomposição em valores singulares

Dada uma matriz real  $\mathbf{M}$  de dimensões  $m \times n$ , a decomposição em valores singulares é uma fatoração da forma:

$$\mathbf{M} = \mathbf{\Theta} \cdot \mathbf{K} \cdot \mathbf{\Psi}^T, \quad (2.14)$$

em que  $\mathbf{\Theta}$  é uma matriz  $m \times m$ ,  $\mathbf{K}$  é uma matriz  $m \times n$  e  $\mathbf{\Psi}$  é uma matriz  $n \times n$ . As matrizes  $\mathbf{\Theta}$  e  $\mathbf{\Psi}$  são matrizes unitárias, ou seja:

$$\mathbf{\Theta}^T \cdot \mathbf{\Theta} = \mathbf{I}, \quad (2.15)$$

$$\mathbf{\Psi}^T \cdot \mathbf{\Psi} = \mathbf{I}. \quad (2.16)$$

A matriz  $\mathbf{K}$  é uma matriz diagonal retangular com elementos reais não-negativos  $\delta_i$  na diagonal principal; esses elementos são conhecidos como os valores singulares da matriz  $\mathbf{M}$  e seu conjunto é o vetor  $\boldsymbol{\delta}(\mathbf{M})$ , normalmente ordenado de forma decrescente. No caso de  $\mathbf{M}$  ser um modelo dinâmico de um sistema, as colunas do vetor  $\boldsymbol{\Theta}$  são conhecidas como os vetores singulares de saída, representados por  $\boldsymbol{\theta}_i$ , e as colunas da matriz  $\boldsymbol{\Psi}$  são os vetores singulares de entrada ou  $\boldsymbol{\psi}_i$ . Os vetores  $\boldsymbol{\theta}_i$  e  $\boldsymbol{\psi}_i$  formam uma base ortonormal e representam as direções principais variáveis de saída e de entrada, assim como  $\boldsymbol{\delta}(\mathbf{M})$  corresponde aos ganhos principais do sistema.

Esses conceitos serão utilizados na definição das métricas de controlabilidade CN e MRI, apresentadas nos próximos parágrafos.

### 2.1.5 Número condicional

O número condicional (CN) é uma medida associada à precisão numérica da solução de sistemas lineares multivariáveis. Matematicamente, ele é definido como a razão entre o maior e o menor valor singular de uma matriz, mas considerando que é possível escrever modelos de sistemas MIMO em forma matricial, então pode-se calcular o grau de direção de um modelo dinâmico de uma planta  $\mathbf{G}(s)$ , para uma frequência  $\omega$  fixa, ou seja, calcular o CN de  $\mathbf{G}(i \cdot \omega)$  :

$$\text{CN}(\mathbf{G}(i \cdot \omega)) = \frac{\max \boldsymbol{\delta}(\mathbf{G}(i \cdot \omega))}{\min \boldsymbol{\delta}(\mathbf{G}(i \cdot \omega))}. \quad (2.17)$$

Em última análise o CN de um sistema MIMO pode ser entendido como uma medida da disparidade entre a magnitude dos ganhos do sistema  $\mathbf{G}(s)$ , dado que  $\max \boldsymbol{\delta}(\mathbf{G}(i \cdot \omega))$  corresponde ao maior ganho possível para um vetor unitário e  $\min \boldsymbol{\delta}(\mathbf{G}(i \cdot \omega))$  ao menor ganho (excetuando-se os ganhos nulos dos casos de sistemas que possuem mais variáveis de entrada que variáveis de saída). Para o maior ganho, existem as direções correspondentes de entrada  $\boldsymbol{\psi}_i$  e de saída  $\boldsymbol{\theta}_i$ , que podem ser entendidas como as direções mais importantes, da mesma forma existem as direções correspondentes ao menor ganho (SKOGESTAD; POSTLETHWAITE, 2007). Dessa maneira, um sistema que apresenta um CN elevado possui uma direção em

que o ganho é muito maior que o ganho em outra direção, em outras palavras, isso quer dizer que o sistema é muito mais sensível em uma direção do que em outra.

Um exemplo prático para visualizar a importância do grau de direção de um sistema na sua controlabilidade intrínseca é apresentado por Skogestad e Postlethwaite (2007). Considerando que o sistema a ser controlado seja um carrinho de supermercado que pode ser movimentar em 3 direções: para frente (direção  $x$ ), para os lados (direção  $y$ ) e para cima (levantar o carrinho, o qual corresponde a direção  $z$ ). Nesse sistema, a direção que apresenta o maior ganho é para frente, seguida de movimentar o carrinho para os lados e, por fim, a direção com menor ganho é para cima. Imaginando-se que se deseja movimentar o carrinho para os lados (ganho intermediário), com a finalidade de desviar de uma pessoa, por exemplo, é necessário exercer uma força (variável de entrada) sobre o carrinho de modo que ele se mova na direção requerida. Entretanto, caso exista uma incerteza sobre a direção ao qual o carrinho está apontando, ele pode inadvertidamente mover-se para frente numa alta aceleração (maior ganho), machucando a pessoa a ser contornada. Isso poderia ser interpretado como uma desobediência a uma restrição do processo, ocasionada pelas incertezas do modelo. Assim, fica evidenciado como um sistema com elevado CN e sob a presença de incertezas tem o seu controle dificultado.

Ainda sobre o exemplo do carrinho, considerando que  $\mathbf{G}(s)$  seja o modelo que explica a aceleração do carrinho  $\mathbf{A}(s)$  de acordo com as forças aplicadas  $\mathbf{F}(s)$ , então suponha-se por hipótese que é possível relacionar essas variáveis por:

$$\begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0,01 \end{bmatrix} \cdot \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix}. \quad (2.18)$$

Aqui fica explicitado matematicamente que  $x$  e  $z$  são as direções que possuem o maior e o menor ganhos, com valores de 3 e 0,01, respectivamente. Portanto, o valor do CN é 300, que é a razão entre os ganhos, e mostra o alto grau de direcionalidade do sistema, isto é, o sistema possui uma direção que é mais sensível que outra. Dada as incertezas do processo e da modelagem, os elementos nulos de  $\mathbf{G}(s)$  poderiam ser na verdade outros valores, o que gerariam movimentos não previstos pelo modelo nominal, acarretando nas consequências já citadas.

O CN é utilizado neste trabalho para identificar o modelo menos controlável, considerando as incertezas de modelagem, conforme será apresentado na Seção 2.2.1.

### 2.1.6 Índice de Resiliência de Morari

Em seu trabalho, Morari (1983) define a resiliência de um sistema como a característica do mesmo em se mover rápido e suavemente de um ponto de operação para outro, incluindo as partidas e paradas, e sua habilidade em lidar efetivamente com distúrbios. O índice de resiliência de Morari ou MRI, o qual mensura essa propriedade do sistema, é matematicamente definido, para um dado valor de  $\omega$ , como o menor valor singular do sistema:

$$\text{MRI} = \min \delta(\mathbf{G}(i \cdot \omega)). \quad (2.19)$$

Assim, fica evidente que o MRI nada mais é que o ganho do sistema na sua direção menos eficiente. Quanto menor o valor do MRI mais difícil é o controle do sistema, no limite em que  $\text{MRI} \rightarrow 0$ , a matriz  $\mathbf{G}(s)$  não é invertível e não é possível controlar o sistema independentemente de qual controlador seja utilizado. O valor do CN e do MRI dependem da escala de  $\mathbf{G}(s)$ , podendo ter valores arbitrariamente altos (já os valores mínimos têm limites: o menor valor possível para CN é 1 e para o MRI é 0) e, portanto, é interessante, para efeitos de análise e comparação, que as variáveis de entrada e de saída estejam normalizadas. A normalização pode ser feita dividindo as variáveis de entrada e saída pelo maior valor permitido para elas como, por exemplo, um limite operacional ou de segurança de modo a se obter um modelo normalizado  $\mathbf{G}^N(s)$ . Essa normalização possui uma vantagem na interpretação física do valor do MRI: por exemplo, no caso em que  $\text{MRI} = 0,1$  e considerando que todas as variáveis estão normalizadas, uma mudança unitária na direção menos eficiente das variáveis de entrada (a qual é máxima permitida) somente causará uma mudança de magnitude igual a 0,1 na pior direção das variáveis de saída. Dessa maneira, conclui-se que se  $\text{MRI}(\mathbf{G}^N(i \cdot \omega)) < 1$ , existe a possibilidade de se ter problemas de saturação no controle, já que não seria possível fazer variar as variáveis de saída num valor entre  $\pm 1$  e controlá-las independentemente, mesmo fazendo as maiores mudanças permitidas nas variáveis de entrada (SKOGESTAD, 2001).

Voltando ao exemplo do carrinho de supermercado, o MRI seria o ganho na direção para cima, isto é, 0,01. Dessa maneira é possível conceber uma situação análoga a saturação do controle, considerando que a pessoa que empurra o carrinho não tem força suficiente para levantá-lo para subir um desnível, por exemplo. A saturação do controle denota uma limitação do controlador em levar as variáveis para os valores requeridos, isto é, é o maior esforço de controle possível.

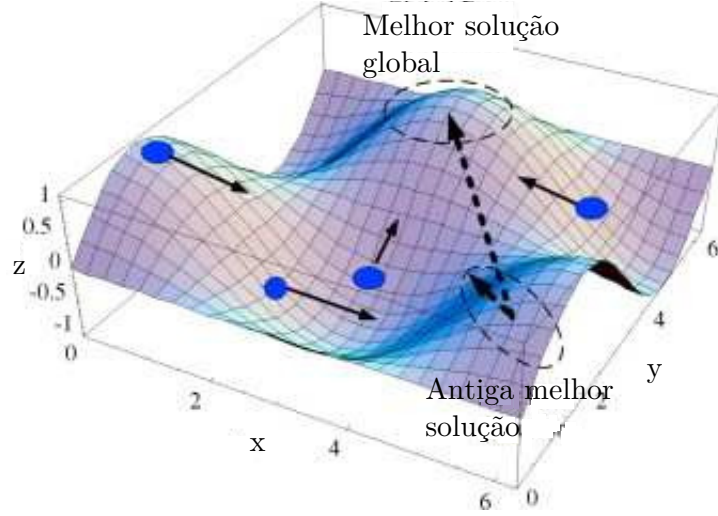
Como pode ser observado, tanto o MRI como o CN não dependem do controlador que será utilizado, sendo funções somente do modelo do sistema. O MRI é utilizado como uma restrição na identificação do modelo menos controlável, o qual é apresentado na Seção 2.2.1.

### 2.1.7 Otimização por enxame de partículas

A técnica de otimização por enxame de partículas ou PSO é um algoritmo meta-heurístico de otimização estocástico inspirado no comportamento dos animais que se movimentam em conjunto, como os peixes em cardumes, os pássaros em bandos, as abelhas nos enxames, etc. A técnica foi introduzida por Kennedy e Eberhart (1995) e, originalmente, tratava-se de um algoritmo para resolução de problemas de otimização somente com variáveis contínuas sem restrições.

No PSO, cada partícula está localizada em um ponto na região de busca de um problema de otimização e possui um valor da função objetivo associado, correspondente àquele ponto. Em uma iteração do algoritmo, cada partícula determina o seu movimento, sobre a região de busca, de acordo com o próprio melhor ponto já encontrado e com o melhor ponto já encontrado por todos ou por uma parte do conjunto, a depender do algoritmo, utilizando uma equação matemática com parâmetros aleatórios. Assim que cada partícula se movimenta, uma nova avaliação da função-objetivo é realizada, reiniciando a iteração. A Figura 2.1 exemplifica a movimentação das partículas (pontos em azul) sobre a região de busca, que nesta ilustração está representada pelo plano formado pelos eixos  $x$  e  $y$ , no intervalo entre 0 e 6, sendo que a função-objetivo é a superfície na qual corresponde aos valores de  $z$ . A forma como cada partícula determina o seu movimento em cada iteração será apresentado nos parágrafos seguintes.

Figura 2.1: Movimentação das partículas sobre a região de busca em uma iteração da otimização realizada pelo PSO.



Fonte: adaptado do sítio "<https://cssanalytics.wordpress.com/2013/09/06/social-learning-algorithms-particle-swarm-optimization-psy/>". Acessado às 17:30 de 21/03/2015.

Considerando que uma partícula  $p$  seja um vetor com  $d$  dimensões, em que  $d$  é o número de variáveis de decisão do problema de otimização, esta partícula é simbolicamente representado por  $\mathbf{z}_h^p$ . Cada elemento do vetor corresponde a uma posição daquela partícula na região de busca e o subscrito  $h$  indica uma dada iteração do algoritmo. Para uma partícula, associa-se um vetor de velocidades  $\mathbf{v}_{h+1}^p \in \mathbb{R}^d$ , que determina para onde a partícula se movimentará na próxima iteração (o que corresponde às setas na figura apresentada anteriormente). Na primeira iteração do algoritmo as posições e as velocidades são, em geral, definidos aleatoriamente e os valores da função-objetivo de cada partícula são então determinados. A partir desses valores iniciais da função-objetivo é possível determinar qual partícula encontrou o melhor resultado e prosseguir com a otimização. Dessa maneira, define-se as equações que regem a posição e a velocidade das partículas do PSO nas iterações seguintes como:

$$\mathbf{v}_{h+1}^p = w_h \mathbf{v}_h^p + c_1 \phi_{1,h}^p (\mathbf{z}_h^p - \mathbf{z}^{p\text{-melhor}}) + c_2 \phi_{2,h}^p (\mathbf{z}_h^p - \mathbf{z}^{g\text{-melhor}}), \quad (2.20)$$

$$\mathbf{z}_{h+1}^p = \mathbf{v}_{h+1}^p + \mathbf{z}_h^p, \quad (2.21)$$

em que  $w_h$  é um parâmetro chamado inércia cujo papel é limitar a velocidade



das partículas ao longo das iterações, o que melhora a estabilidade e convergência do algoritmo (SHI; EBERHART, 1998);  $c_1$  e  $c_2$  são conhecidos como parâmetros cognitivo e social, respectivamente, e determinam se as partículas tenderão a se movimentar como um grupo coeso ou como partículas isoladas, em busca do ponto ótimo;  $\phi_{1,h}^p$  e  $\phi_{2,h}^p$  são números aleatórios uniformemente distribuídos no intervalo entre 0 e 1;  $\mathbf{z}^{p\text{-melhor}}$  é o melhor ponto já encontrado até a iteração  $h$  pela partícula  $p$ , já  $\mathbf{z}^{g\text{-melhor}}$  é o melhor ponto já encontrado pelo grupo de partículas até aquela iteração. Além dos parâmetros já citados ( $w_h$ ,  $c_1$  e  $c_2$ ), outros devem ser definidos pelo usuário como a quantidade de partículas e o número de iterações, os quais também influenciam no sucesso ou fracasso da obtenção da solução ótima.

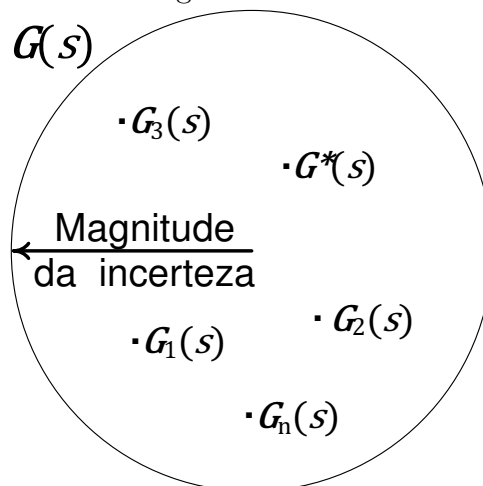
Apesar das limitações do algoritmo original do PSO, ao longo dos anos ele foi sendo estudado e modificações foram introduzidas de forma a melhorar a convergência e aumentar a abrangência dos tipos de problemas que poderiam ser solucionados pela técnica. Hu e Eberhart (2002) desenvolveram um algoritmo PSO que suporta problemas de otimização com restrições, nesse algoritmo todas as partículas são inicializadas somente com soluções exequíveis, isto é, obedecendo a todas as restrições. Laskari, Parsopoulos e Vrahatis (2002) arquitetaram três variantes do PSO para problemas de otimização combinatória (*Integer Programming*) e compararam os resultados obtidos com os da técnica clássica de *Branch and Bound*, mostrando que o PSO foi superior em muitos casos. Lovbjerg e Krink (2002) aperfeiçoaram o PSO, adicionando criticidade auto-organizada (SOC - *self-organized criticality*), de modo a aumentar a diversidade das partículas, acelerar a convergência e melhorar os resultados obtidos. Kitayama e Yasuda (2006) e Datta e Figueira (2011) ampliaram a extensão da técnica para funcionar com variáveis discretas e contínuas, utilizando abordagens diferentes. Os primeiros trataram das variáveis discretas fazendo uso de funções de penalidade, enquanto que os últimos codificaram um PSO em que as variáveis discretas são efetivamente utilizadas, sem o uso de arredondamentos ou decodificações. Dado os bons resultados deste último algoritmo PSO, o qual se mostra adequado para resolver MINLP, ele é usado nesse trabalho para sintonizar os controladores MPC: um problema de otimização não-linear com restrições e variáveis contínuas e inteiras.

## 2.2 O método proposto

### 2.2.1 O modelo com controle mais difícil

O método proposto utiliza o conceito de pior modelo para se controlar. Dentro deste contexto, assume-se que o modelo nominal utilizado no MPC está sujeito a incertezas nos seus parâmetros, de tal modo que o modelo associado ao controle mais difícil está contido no conjunto dos possíveis modelos que podem ser atribuídos ao sistema no domínio da incerteza de modelagem. Assim, considerando que  $\mathbf{G}(s)$  seja o modelo nominal do sistema e que cada um de seus parâmetros possua uma incerteza associada, é válido combinar os possíveis valores desses parâmetros e formar diferentes modelos do sistema que são estatisticamente equivalentes, isto é,  $\mathbf{G}(s)$  combinado com a sua incerteza de modelagem formam, na verdade, um conjunto de modelos; a Figura 2.2 ilustra essa ideia. O modelo de pior controlabilidade pertence a esse conjunto e será denotado por  $\mathbf{G}^*(s)$ . A escolha de  $\mathbf{G}^*(s)$  vai depender da métrica de controlabilidade escolhida. O modelo  $\mathbf{G}^*(s)$  é o que será utilizado na sintonia do MPC como sendo a planta, mas o MPC em si é implementado com o modelo nominal  $\mathbf{G}(s)$  para as previsões.

Figura 2.2: Diagrama do conjunto de modelos que formam  $\mathbf{G}(s)$ , considerando a magnitude das incertezas de modelagem.



Fonte: o próprio autor.

No método proposto, o CN e o MRI são as métricas utilizadas para determinar o modelo  $\mathbf{G}^*(s)$ , a partir do modelo já normalizado  $\mathbf{G}^N$ , através da solução do seguinte

problema de otimização:

### Problema 3.1

$$\mathbf{G}^*(s) = \arg \max_{\mathbf{\Xi}, \omega} \text{CN}(\mathbf{G}^N(i \cdot \omega)), \quad (2.22)$$

sujeito a:

$$\text{MRI}(\mathbf{G}^N(i \cdot \omega)) < 1, \quad (2.23)$$

$$\mathbf{\Xi}_{\min} < \mathbf{\Xi} < \mathbf{\Xi}_{\max}, \quad (2.24)$$

$$\omega_{\min} < \omega < \omega_{\max}, \quad \omega \geq 0, \quad (2.25)$$

em que  $\mathbf{\Xi}$  é o conjunto dos possíveis valores dos parâmetros de  $\mathbf{G}$ , considerando as incertezas de modelagem. Caso não se saiba qual faixa de frequência utilizar, um valor de referência pode ser 0 para a frequência mínima e a máxima 2 décadas acima da frequência de corte do sistema<sup>1</sup>. Para o caso especial em que  $\omega = 0$ , o CN e o MRI calculados se referem ao estado estacionário do sistema. A Equação (2.22) serve para encontrar um modelo com o maior grau de direção possível e mais sensível às incertezas de modelagem. A Equação (2.23) é uma restrição incluída para encontrar um sistema que possa ter problemas com saturação no controle.

Dessa forma, esse procedimento seleciona a pior planta a ser controlada e, ao realizar a sintonia ótima a partir dessa planta, pretende-se encontrar o conjunto de parâmetros do controlador MPC que possui algum grau de robustez. Na estratégia de sintonia proposta, o modelo de simulação é a pior planta e o modelo de predição é o modelo nominal identificado. A ideia é que se o sistema é controlado para o caso mais crítico, existe a possibilidade de ele o ser para as demais plantas, como será mostrado nos resultados do capítulo seguinte. Essa estratégia evita ter que rodar vários problemas de otimização, um para cada combinação de parâmetros do modelo da planta com incerteza.

---

<sup>1</sup>Para um sistema MIMO, a maior frequência de corte do par de variável de entrada e de saída pode ser utilizada

## 2.3 Critérios de desempenho escolhidos

Como será visto no próximo capítulo, neste trabalho é realizada a sintonia do MPC com o método proposto em dois estudos de caso de processos industriais diferentes, em que cada processo tem as suas particularidades e demandam desempenhos de controle peculiares.

O primeiro estudo de caso trata do SSCP e nele optou-se por utilizar a minimização da soma do ITSE discretizado das PVs como critério de desempenho, a fim de assegurar o rastreamento delas em torno dos SPs, e o movimento das MVs também foram levados em consideração, resultando num critério de desempenho multi-objetivo de acordo com a seguinte equação:

### Problema 3.2a

$$\Phi(\mathbf{\Omega}) = \arg \min_{\mathbf{\Omega}} \sum_{k=0}^{tsim} kT \cdot [(\mathbf{y}(k) - \mathbf{y}_{sp,k})^T \mathbf{\Lambda}(\mathbf{y}(k) - \mathbf{y}_{sp,k}) + \Delta \mathbf{u}(k)^T \mathbf{\Upsilon} \Delta \mathbf{u}(k)], \quad (2.26)$$

sujeito a:

$$\mathbf{\Omega}_{\min} < \mathbf{\Omega} < \mathbf{\Omega}_{\max}, \quad (2.27)$$

em que  $\mathbf{\Omega}$  é um vetor que contém todos os parâmetros de sintonia do problema, isto é,  $\mathbf{\Omega} = [H_p, H_c, \mathbf{Q}, \mathbf{R}]$  e a Equação 2.27 define a região de busca para o PSO. As matrizes  $\mathbf{\Lambda} \in \mathbb{R}^{n_y \times n_y}$  e  $\mathbf{\Upsilon} \in \mathbb{R}^{n_u \times n_u}$  são diagonais e usadas para ponderar apropriadamente importância de cada variável, mas neste estudo de caso todas as variáveis foram consideradas igualmente importantes e, portanto, essas matrizes são iguais a matriz identidade.

O segundo estudo de caso versa sobre um separador bifásico onde ocorre a separação das fases gasosa e líquida de uma corrente de petróleo. O critério de desempenho escolhido é heterodoxo e combina o ITSE normalizado de uma das PVs com um parâmetro que mensura o amortecimento do distúrbio que entra no processo na outra PV, a razão de amplitude RA, segundo a equação:

### Problema 3.2b

$$\Phi(\Omega) = \arg \min_{\Omega} \frac{\sum_{k=0}^{tsim} k \cdot [y_2(k) - y_{2sp,k}]^2}{\sum_{k=0}^{tsim} 9k} + RA, \quad (2.28)$$

$$(2.29)$$

$$RA = \frac{A_{Lout}}{A_{Lin}}, \quad (2.30)$$

sujeito a:

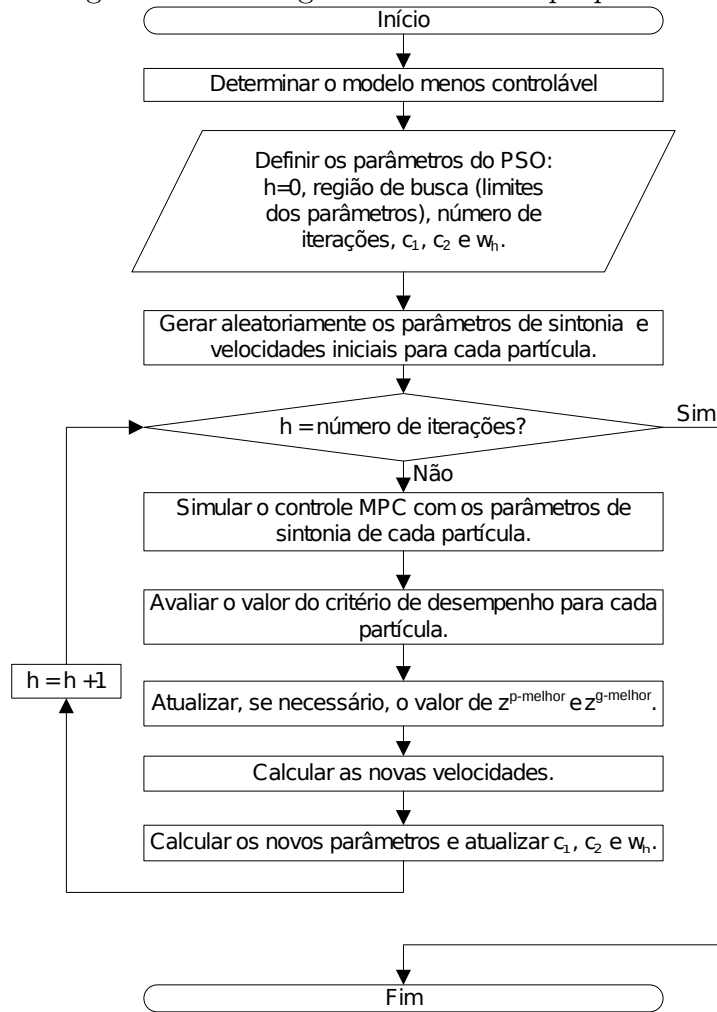
$$\Omega_{\min} < \Omega < \Omega_{\max}, \quad (2.31)$$

em que RA é a razão entre as amplitudes das vazões de líquido na saída e na entrada do sistema no estudo de caso. Na próxima seção é apresentada a estrutura do método proposto, mostrando como o algoritmo PSO é utilizado para a obtenção dos parâmetros de sintonia ótimos.

#### 2.3.1 Configuração do algoritmo

O PSO foi configurado utilizando 40 partículas, seguindo as orientações de Poli, Kennedy e Blackwell (2007) que preconizam um número entre 20 e 50, e limitou-se o número de iterações em 1000, pois se verificou que não há mudanças sensíveis nos resultados obtidos para um número maior de iterações. Por sua vez, o MPC foi configurado especificamente para cada estudo de caso, no que se refere às restrições, *set-points*, etc. As etapas do método de sintonia ótima proposto podem ser visualizadas no fluxograma abaixo, que será descrito detalhadamente como segue:

Figura 2.3: Fluxograma do método proposto.



Fonte: o próprio autor.

A primeira etapa é definir o modelo menos controlável que será a planta para o controlador MPC, o qual foi descrita na Seção 2.2.1. É importante enfatizar que este não é o modelo que o controlador utiliza para realizar as predições, sendo na verdade o modelo do sistema que responde às excitações e distúrbios e do qual são medidas as variáveis de saída, isto é, a planta propriamente dita. O MPC é implementado utilizando o modelo nominal desta planta, isto é, a representação matemática que foi determinada pelos métodos de identificação de sistema ou modelagem fenomenológica e que busca aproximar-se do comportamento dinâmico da planta. Assim, existe uma diferença entre os dois modelos e a sintonia obtida pelo método apresentado aqui deve ser suficientemente robusta para controlar o sistema sob essas condições.

Em seguida, os parâmetros iniciais do PSO devem ser configurados: número de

partículas,  $c_1$ ,  $c_2$ ,  $w_h$  e o número de iterações. Existe uma extensa quantidade de artigos que se debruçam sobre os valores a serem adotados para esses parâmetros do PSO (POLI; KENNEDY; BLACKWELL, 2007). Neste trabalho eles variam a cada iteração, conforme o algoritmo de enxame apresentado por Datta e Figueira (2011).

Para a definição da região de busca dos parâmetros do controlador, utilizou-se os algoritmos heurísticos de sintonia de MPC como referência. Dessa maneira, foi considerado como limite superior de  $H_p$  o valor obtido pela Equação 2.32, presente no artigo de Shridhar e Cooper (1998):

$$H_p^{\max} = \max \left( \frac{5 \cdot \tau_{i,j} + \gamma_{i,j}}{T} + 1 \right), \quad i = 1, \dots, ny, \quad j = 1, \dots, nu, a \quad (2.32)$$

e o limite inferior é dado por:

$$H_p^{\min} = \max \left( \frac{\gamma_{i,j}}{T} + 1 \right), \quad i = 1, \dots, ny, \quad j = 1, \dots, nu. \quad (2.33)$$

Com relação aos limites de  $H_c$ , o limite inferior configurado foi 1 e o superior 6, para evitar um esforço computacional excessivo, contudo pode-se encontrar na literatura outros valores para se ter como referência ao definir essa faixa. Por exemplo, é possível definir o  $H_c^{\max}$  como sendo um quarto do valor calculado pela Equação 2.32 (TRIERWEILER; FARINA, 2003). Com as variáveis normalizadas, os elementos das matrizes diagonais  $\mathbf{Q}$ ,  $\mathbf{R}$  e  $\mathbf{S}$  podem ser variados dentro de uma faixa entre 0 e 1, pois seus valores absolutos não importam mas, sim, os valores relativos entre si. Essa normalização é importante, porque auxilia na convergência numérica do método e permite que variáveis com dimensões e magnitudes diferentes sejam comparadas numa mesma base.

Como é usual dos algoritmos estocásticos de otimização, o valor inicial das variáveis de decisão (parâmetros de sintonia) é gerado aleatoriamente, obedecendo a região de busca previamente estabelecida. Assim, na primeira iteração o PSO gera aleatoriamente os valores iniciais dos parâmetros de sintonia e as velocidades associadas para cada partícula. Como no algoritmo cada partícula é um conjunto de parâmetros de sintonia do MPC, são configurados 40 (igual ao número de partículas) controladores MPC, seguindo cada um dos conjuntos de parâmetros. Esses controladores são implementados com as informações do modelo nominal para predição, restrições, SPs e todos outros dados necessários para o seu funcionamento.

O próximo passo é simular uma situação em que o sistema é perturbado de alguma forma, seja por mudança de *set-points* ou por distúrbios, e os resultados são gravados. Com esses dados, o PSO computa o critério de desempenho de controle de cada MPC e atualiza os valores de  $\mathbf{z}^{p\text{-melhor}}$  e  $\mathbf{z}^{g\text{-melhor}}$ , caso seja encontrado um conjunto de parâmetros com desempenho superior ao anteriormente obtido. Os parâmetros de sintonia são modificados de acordo com as velocidades, como mostrado pelas Equações 2.20 e 2.21, e os 40 controladores são reconfigurados com esses novos parâmetros. Novas simulações são feitas com esses parâmetros, reiniciando o ciclo que se repete até o número máximo de iterações definido. No caso de algoritmos de otimização meta-heurísticos, como o PSO, não é comum o término da rotina em caso da não evolução da função-objetivo a cada iteração. Isso quer dizer que mesmo que a função-objetivo se mantenha no mesmo valor por algumas iterações, a otimização continuará até que se chegue no limite de iterações definido. Neste método de sintonia esta prática foi conservada.

Todo o algoritmo do método proposto foi escrito em linguagem MATLAB<sup>®</sup> e as rotinas de cálculos foram executadas em um computador com processador Intel<sup>®</sup> Core<sup>™</sup> 3i de 2,3 GHz e 4 GB de memória RAM.

## 2.4 Conclusões

Este capítulo revisou alguns conceitos e apresentou o método proposto para sintonia ótima de controladores MPC com restrições que, diferentemente dos demais, considera a incerteza nos parâmetros do modelo do sistema a ser controlado. Após determinar o modelo de difícil controle, a otimização via PSO encontra os parâmetros de sintonia inteiros e contínuos.

Apesar do método aplicar o CN e o MRI, o usuário pode decidir por outras métricas para determinação do pior caso de controle do sistema, definir o tipo de controlador MPC, o modelo de planta e suas restrições e, assim, o método proposto poderia ser aplicado para quaisquer casos de sintonias de controladores preditivos.

A validação do método é feita no Capítulo 3, em que são feitos dois estudos de casos com objetivos diferentes cujos resultados evidenciam a sua eficácia.



---

# Capítulo 3

## Estudos de caso

---

*“Many "educated citizens" take it for granted that reality is what scientists say it is and that other opinions may be recorded, but need not to be taken seriously. But science offers not one story, it offers many [...].”*

Paul K. Feyerabend

Neste capítulo o método de sintonia proposto é aplicado a dois exemplos típicos da indústria de processo, documentados na literatura. O primeiro deles foi objeto de estudo do artigo publicado pelo autor deste trabalho (JÚNIOR; MARTINS; KALID, 2014) e se refere ao SSCP e os resultados serão replicados na próxima seção. O outro exemplo é um separador bifásico de óleo e água de alta pressão, muito comum em plataformas de petróleo.

### 3.1 *Shell Standard Control Problem - SSCP*

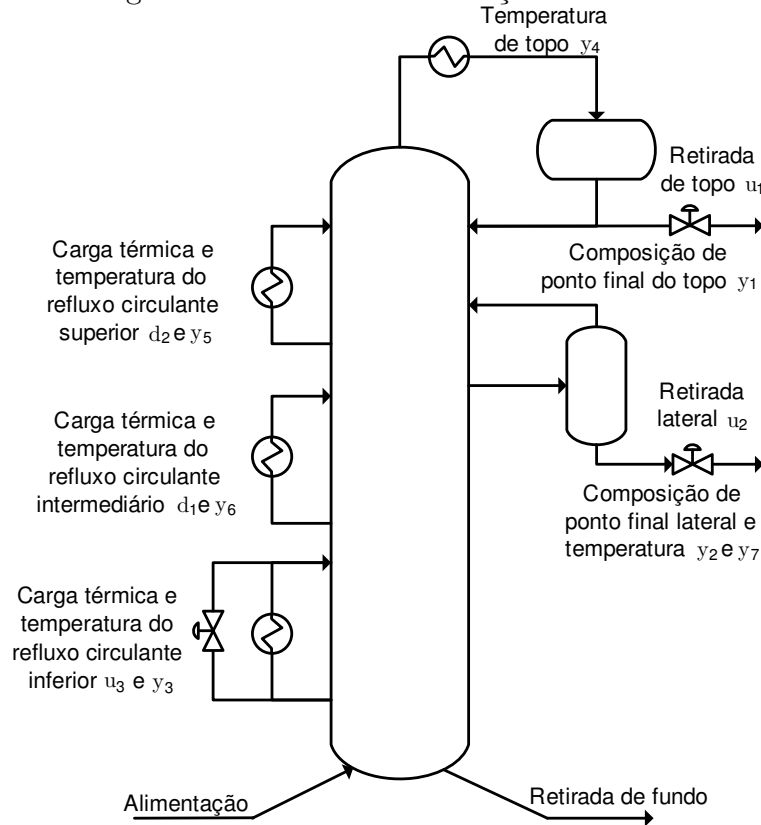
O SSCP (PRETT; MORARI, 1987) consiste numa coluna de destilação (Figura 3.1) em que uma corrente gasosa e quente alimenta o processo pelo fundo e é separada em 3 correntes. Parte da energia térmica é recuperada pelos refluxos circulantes<sup>1</sup> nos trocadores de calor em 3 pontos diferentes ao longo da coluna. Originalmente, o processo é composto por 7 variáveis de saída medidas e 5 variáveis de entrada. Das 5 variáveis de entrada, 3 são MVs: as vazões das retiradas do topo ( $u_1$ ) e lateral ( $u_2$ ) e a carga térmica no refluxo circulante inferior ( $u_3$ ); as outras 2 variáveis são

---

<sup>1</sup>Maiores detalhes sobre os termos e o processo de destilação podem ser encontrados no livro (ARAÚJO; BRASIL; SOUZA, 2011)

distúrbios medidos: cargas térmicas do refluxo circulante intermediário ( $d_1$ ) e de superior ( $d_2$ ). Três variáveis de saída são PVs: composições de ponto final do topo ( $y_1$ ) e lateral ( $y_2$ ) e a temperatura do refluxo circulante de fundo ( $y_3$ ). As 4 variáveis de saída medidas restantes são as temperaturas: da corrente de topo ( $y_4$ ), do refluxo circulante superior ( $y_5$ ) e intermediário ( $y_6$ ) e da retirada lateral ( $y_7$ ).

Figura 3.1: Coluna de destilação do SSCP.



Fonte: o próprio autor.

Contudo, para fins de simplificação será utilizada uma versão menor do sistema (VLACHOS; WILLIAMS; GOMM, 2002), com apenas 2 variáveis de entrada e 2 de saída, que são as composições do ponto final do topo ( $y_1$ ) e lateral ( $y_2$ ) e as vazões das retiradas do topo ( $u_1$ ) e lateral ( $u_2$ ). O modelo nominal do sistema descrito por eles é uma matriz de funções de transferências, escrita com variáveis de desvio, e está equacionada da seguinte maneira:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{4,05 \cdot e^{-27 \cdot s}}{50 \cdot s + 1} & \frac{1,77 \cdot e^{-28 \cdot s}}{60 \cdot s + 1} \\ \frac{5,39 \cdot e^{-18 \cdot s}}{50 \cdot s + 1} & \frac{5,72 \cdot e^{-14 \cdot s}}{60 \cdot s + 1} \end{bmatrix} \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}. \quad (3.1)$$

em que os ganhos estão normalizados e os tempos mortos e constantes de tempo estão em minutos. Foi considerado que todos os parâmetros do modelo possuem uma incerteza associada e como forma de levar-se em conta essa informação, atribuiu-se uma distribuição uniforme para cada parâmetro, com limites inferior e superior correspondentes a -10% e +10% do valor nominal dos parâmetros, respectivamente (cf. Tabela 3.1).

Tabela 3.1: Limites dos valores dos parâmetros

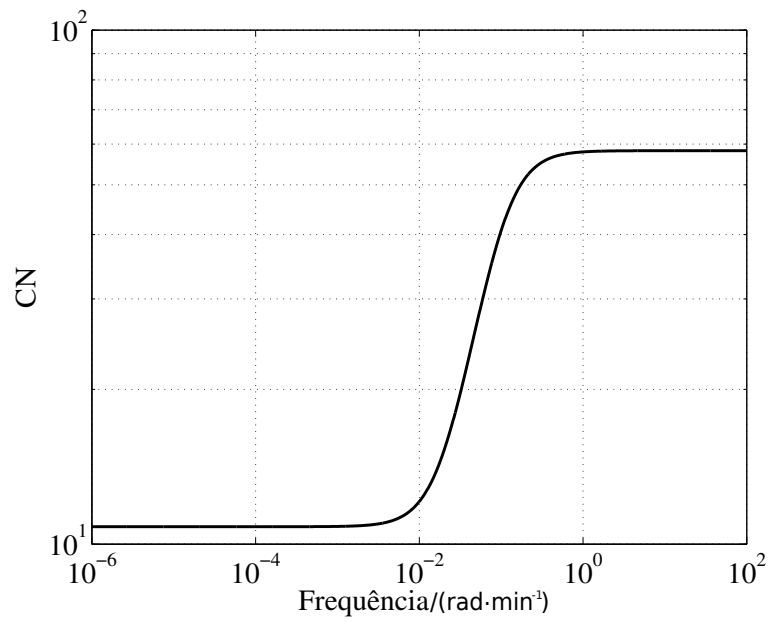
Parâmetro	Limite inferior	Valor nominal	Limite superior
$K_{1,1}$	3,85	4,05	4,25
$K_{1,2}$	1,68	1,77	1,86
$K_{2,1}$	5,66	5,39	5,12
$K_{2,2}$	6,01	5,72	5,43
$t_{1,1}$ /min	25,65	27,00	28,35
$t_{1,2}$ /min	26,60	28,00	29,40
$t_{2,1}$ /min	17,10	18,00	18,90
$t_{2,2}$ /min	13,30	14,00	14,70
$\tau_{1,1}$ /min	47,50	50,00	52,50
$\tau_{1,2}$ /min	57,00	60,00	63,00
$\tau_{2,1}$ /min	47,50	50,00	52,50
$\tau_{2,2}$ /min	57,00	60,00	63,00

Com esses limites, para uma frequência  $\omega$  de 0 a 100 rad/min e com os critérios apresentados pelas Equações 2.22 a 2.25, obteve-se o seguinte modelo  $\mathbf{G}^*(s)$ :

$$\mathbf{G}^*(s) = \begin{bmatrix} \frac{3,645 \cdot e^{-26,8 \cdot s}}{55 \cdot s + 1} & \frac{1,947 \cdot e^{-25,9 \cdot s}}{54 \cdot s + 1} \\ \frac{5,929 \cdot e^{-18,5 \cdot s}}{45 \cdot s + 1} & \frac{5,148 \cdot e^{-13,1 \cdot s}}{66 \cdot s + 1} \end{bmatrix} \quad (3.2)$$

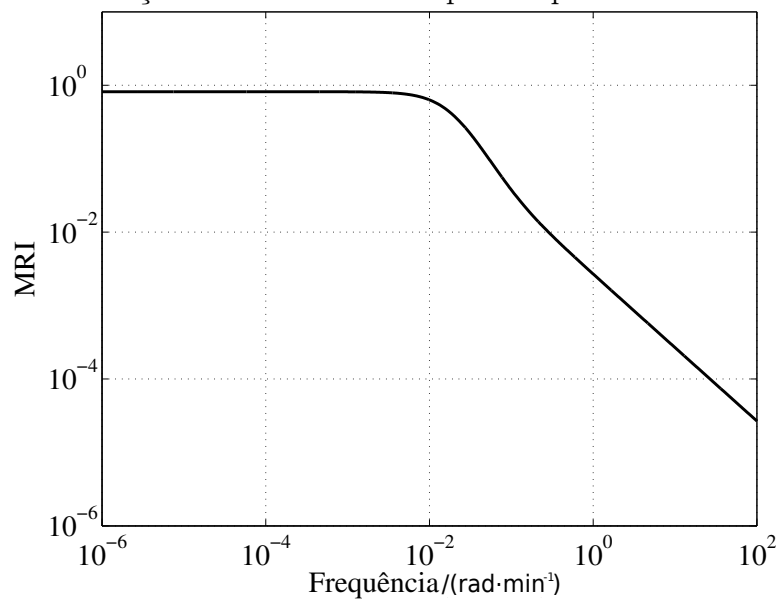
Nas Figuras 3.1 e 3.2 são mostrados os comportamentos do CN e do MRI em função da frequência para o sistema em questão, em que fica evidenciado o aumento da dificuldade em controlá-lo com o crescimento da frequência, pois como pode ser observado o CN chega no patamar mais alto e o MRI diminui monotonicamente.

Figura 3.2: Variação do CN com a frequência para o modelo do SSCP.



Fonte: o próprio autor.

Figura 3.3: Variação do MRI com a frequência para o modelo do SSCP.



Fonte: o próprio autor.

A lei de controle do MPC foi configurada com as Equações 1.1 a 1.7, mas desconsiderando os *targets* das MVs, isto é, todos os elementos de  $\mathbf{S}$  são iguais a zero. A região de busca, isto é, os valores de  $\mathbf{\Omega}_{\min}$  e  $\mathbf{\Omega}_{\max}$  estão especificados na Tabela 3.2.

Tabela 3.2: Região de busca dos parâmetros sintonizáveis do MPC para o SSCP.

Parâmetro ( $\mathbf{\Omega}$ )	Limite inferior ( $\mathbf{\Omega}_{\min}$ )	Limite superior ( $\mathbf{\Omega}_{\max}$ )
$H_p$	8	82
$H_c$	1	6
$q_1$	0	1
$q_2$	0	1
$r_1$	0	1
$r_2$	0	1

Foram simulados dois casos de controle com características distintas, sendo que a primeira diferença entre eles está nos limites das PVs e MVs das Equações 1.5 e 1.7 para o MPC. As Tabelas 3.3 e 3.4 mostram os limites das PVs e das MVs para os dois casos. A simulação do Caso 1 é de 150 instantes de tempo ( $tsim = 150$ ), enquanto que para o Caso 2 a duração é maior, de 250 instantes de tempo ( $tsim = 250$ ). Outra diferença é que no Caso 2 foi introduzido um distúrbio não medido equivalente a um degrau de +0,05 de  $\Delta u_1$  no sistema, isto é, um erro sistemático, no instante de tempo  $k = 150$ . Em ambos os casos, a simulação consiste em uma mudança no SP de  $Y_1$  de 0 para 0,4 e de  $Y_2$  de 0 para 0,2. O PSO resolveu o problema de otimização 3.2a e retornou os valores de sintonia ótimos apresentados na Tabela 3.5.

Tabela 3.3: Restrições rígidas das MVs e PVs para o Caso 1 do SSCP.

Variável	Valor mínimo	Valor máximo	Variação máxima ( $\Delta^{\max}$ )
$U_1$	-0,5	0,5	0,2
$U_2$	-0,5	0,5	0,2
$Y_1$	-0,5	0,5	-
$Y_2$	-0,5	0,5	-

Tabela 3.4: Restrições rígidas das MVs e PVs para o Caso 2 do SSCP.

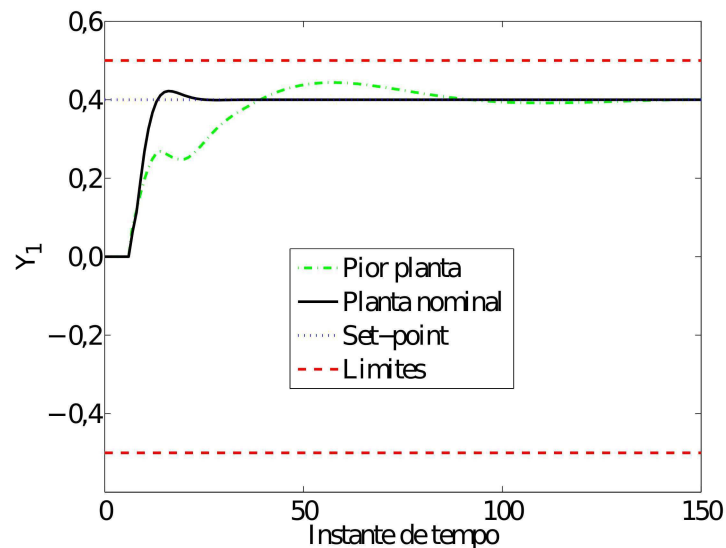
Variável	Valor mínimo	Valor máximo	Variação máxima( $\Delta^{\max}$ )
$U_1$	-0,5	0,5	0,2
$U_2$	-0,4	0,5	0,2
$Y_1$	-0,5	0,5	-
$Y_1$	-0,5	0,3	-

Tabela 3.5: Parâmetros ótimos encontrados pelo PSO para os casos do SSCP.

	$H_p$	$H_c$	$q_1$	$q_2$	$r_1$	$r_2$
Caso 1	82	5	0,0001	0,9918	0,0000	0,0023
Caso 2	14	1	0,8097	0,4626	0,0000	0,0000

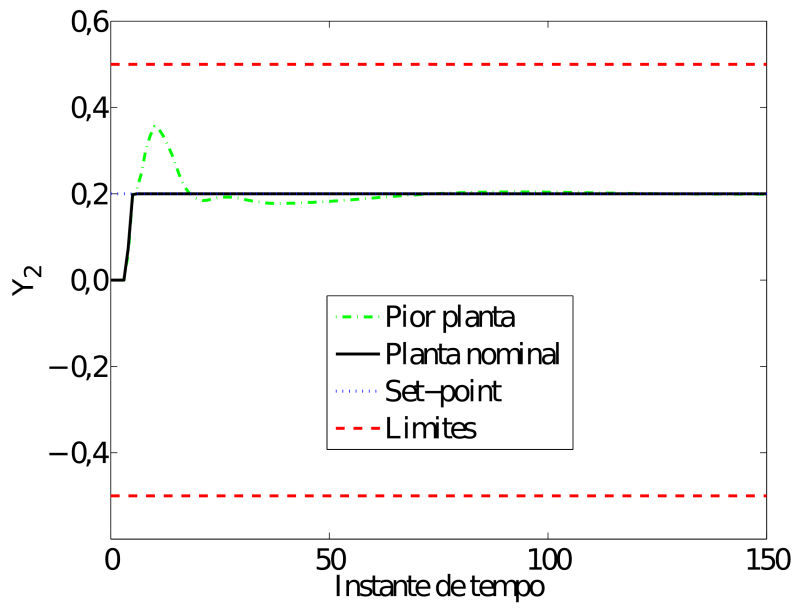
As Figuras 3.4 a 3.7 mostram a resposta em malha fechada das PVs e MVs para o Caso 1, utilizando a sintonia ótima, para a planta nominal e para a planta menos controlável, em que é possível perceber que o controlador obedeceu às restrições impostas a ele, para as duas situações. Como era esperado, para o caso menos controlável a resposta foi mais lenta, com tempos de acomodação e de subida maiores, devido a própria natureza do sistema e da diferença entre o modelo do MPC e da planta.

Figura 3.4: Comportamento dinâmico de  $Y_1$  no Caso 1 do SSCP.



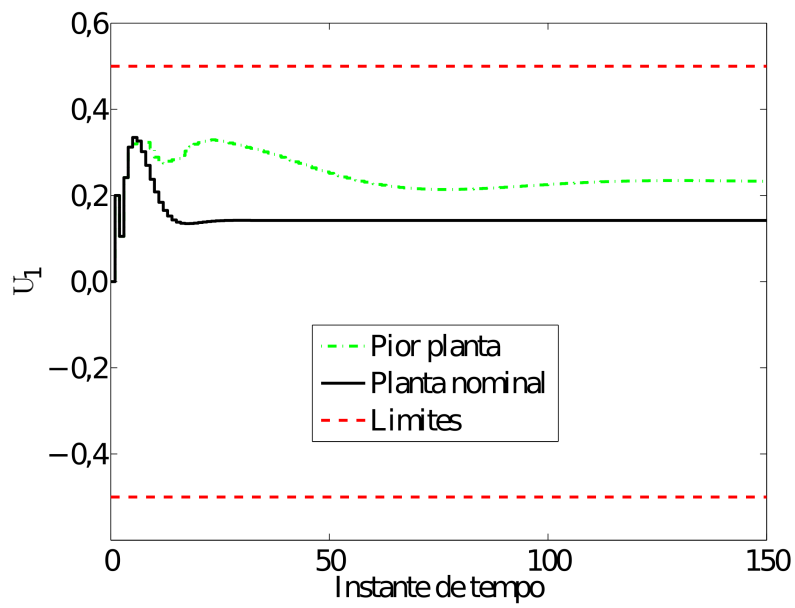
Fonte: o próprio autor.

Figura 3.5: Comportamento dinâmico de  $Y_2$  no Caso 1 do SSCP.



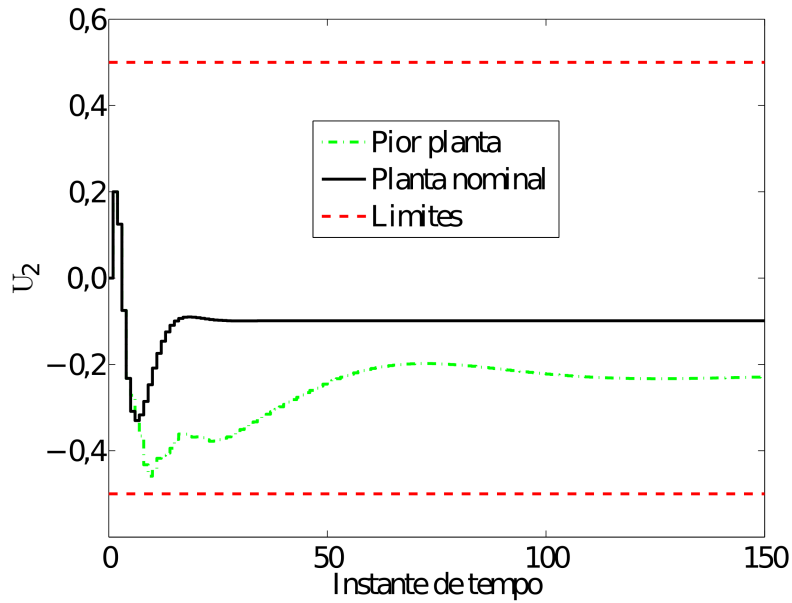
Fonte: o próprio autor.

Figura 3.6: Comportamento dinâmico de  $U_1$  no Caso 1 do SSCP.



Fonte: o próprio autor.

Figura 3.7: Comportamento dinâmico de  $U_2$  no Caso 1 do SSCP.

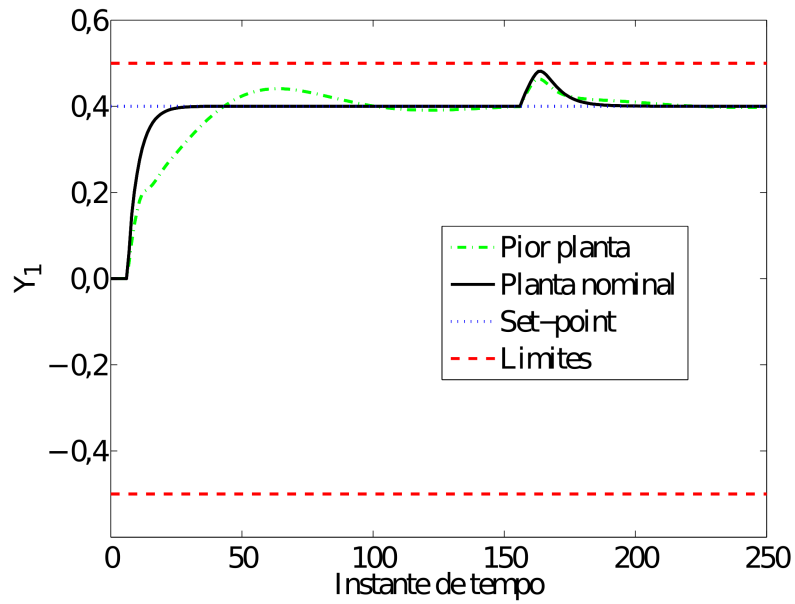


Fonte: o próprio autor.

No Caso 2, mostrado nas Figuras 3.8 a 3.11, da mesma forma que no Caso 1 e pelos mesmos motivos, a resposta foi mais rápida para a planta nominal comparando-a com a planta de difícil controle. Dada as restrições mais "apertadas", obteve-se uma sintonia com um menor *overshoot* em  $Y_2$ , mesmo limitando ainda mais o valor inferior de  $U_2$ . A contrapartida é que a sintonia obtida é um pouco mais lenta que a do Caso 1, como será mostrado numericamente a seguir. A partir dos gráficos, é possível perceber também que o controlador conseguiu aceitar com êxito o distúrbio introduzido no sistema no instante  $k = 150$ . As PVs saíram temporariamente de suas posições (os SPs), mas depois retornaram rapidamente, enquanto que a entrada  $u_1$  convergiu para um novo valor de estado estacionário, em virtude dessa MV ter que compensar os efeitos do distúrbio introduzido.

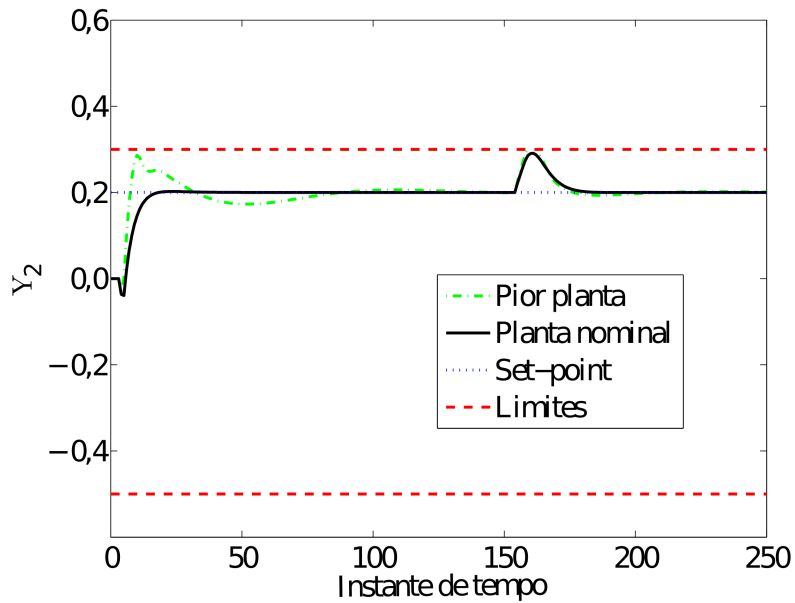


Figura 3.8: Comportamento dinâmico de  $y_1$  no Caso 2 do SSCP.



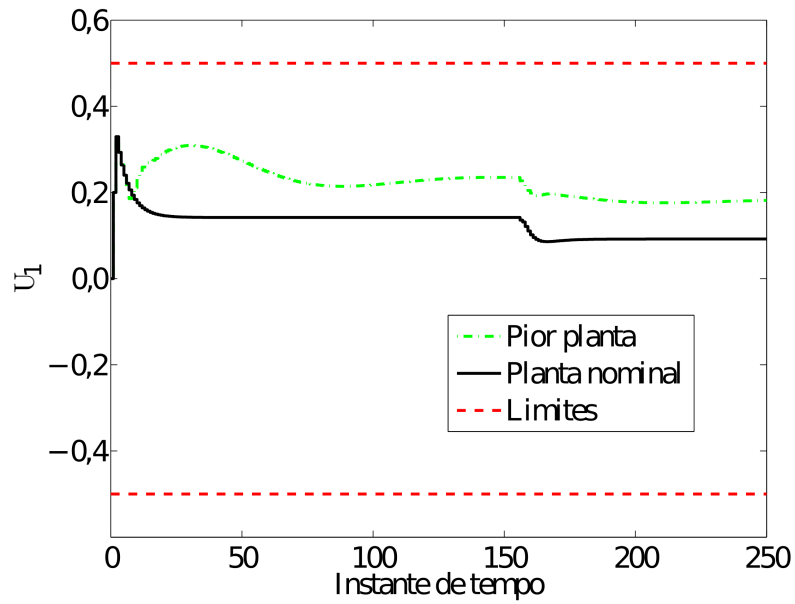
Fonte: o próprio autor.

Figura 3.9: Comportamento dinâmico de  $y_2$  no Caso 2 do SSCP.



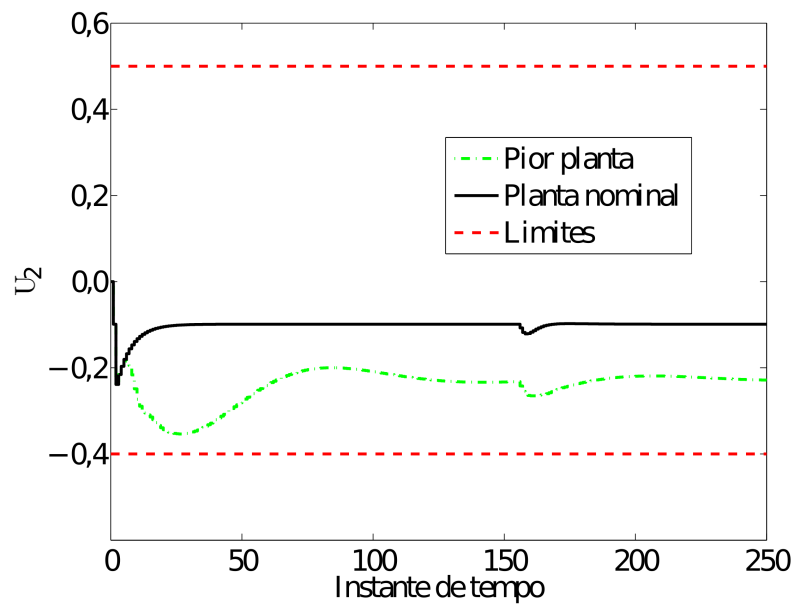
Fonte: o próprio autor.

Figura 3.10: Comportamento dinâmico de  $u_1$  no Caso 2 do SSCP.



Fonte: o próprio autor.

Figura 3.11: Comportamento dinâmico de  $u_2$  no Caso 2 do SSCP.



Fonte: o próprio autor.

A Tabela 3.6 resume em números a comparação do desempenho do controlador, nos primeiros 150 instantes de tempo, para os cenários de simulação com a planta nominal e com a planta menos controlável nos Casos 1 e 2. A terceira linha é uma comparação entre os casos e a terceira coluna é uma comparação entre as plantas.

Tabela 3.6: Valores da função-objetivo para os cenários de com a planta nominal e com a planta menos controlável, para os Caso 1 e 2.

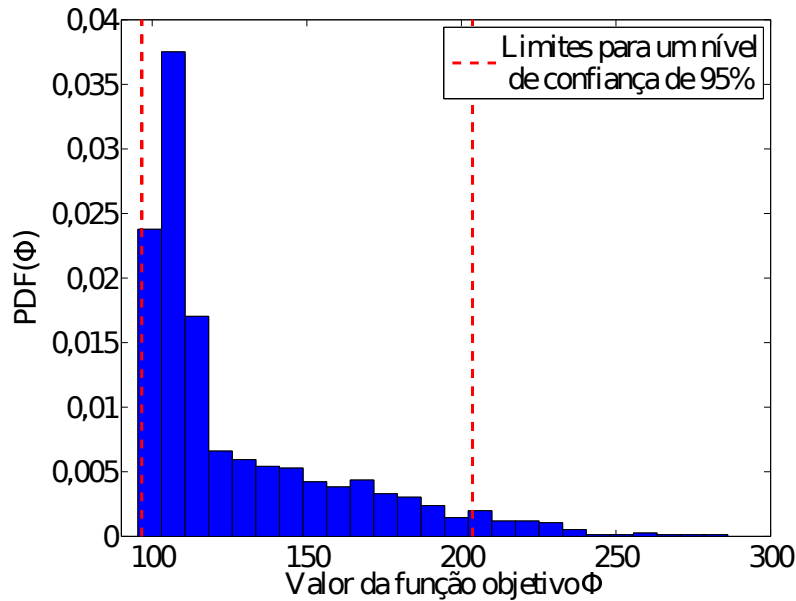
	Planta nominal ( $N$ )	Planta menos controlável ( $W$ )	$(W - N)/N$ /%
Caso ( $C_1$ )	103	285	+177
Caso ( $C_2$ )	116	334	+287
$(C_2 - C_1)/C_1$ /%	+13	+15	-

Com esses dados, é possível afirmar com maior propriedade que a sintonia do Caso 1 é mais rápida que para o Caso 2, haja vista que o valor da função-objetivo do primeiro é 285 contra 334 do segundo, para a planta menos controlável, ao passo que para a planta nominal o valor é de 103 contra 116. Isso pode ser explicado pelas maiores restrições feitas ao controlador para o Caso 2, que limitam as ações possíveis do MPC.

Para mostrar a robustez da sintonia ótima, frente às incertezas de modelagem, foi realizada uma simulação de Monte Carlo variando a planta a ser controlada, conforme os limites dos parâmetros da Tabela 3.1, usando o cenário de controle do Caso 1 e a sintonia ótima obtida para ele, em que foram registrados os valores da função-objetivo de cada simulação, sendo possível a construção do histograma apresentado na Figura 3.12, que é uma aproximação da função densidade de probabilidade (PDF - *Probability Density Function*) da função-objetivo. A partir do histograma é possível inferir que a sintonia tem um bom desempenho para a grande maioria das plantas, haja vista que o valor da função-objetivo na maior parte dos casos é menor que o da pior planta (285). No histograma também aparecem os limites do valor de  $\Phi$  para um nível de confiança de 95%, quais sejam: 98 e 205. Esses limites são estimados seguindo os procedimentos do Suplemento 1 do Guia para Expressão da Incerteza na Medição (GUM-S1) BIPM et al. (2008). O guia orienta que seja calculado o

menor intervalo dentro do qual exista uma probabilidade de 95% de ocorrência da variável. Calculando as estatísticas, a partir dos dados simulados, obteve-se o valor médio de 128 para  $\Phi$  com desvio padrão de 34. Essas estatísticas corroboram com o argumento de que a sintonia é robusta, pois, mesmo com mudanças nos parâmetros da planta, o controlador manteve um ótimo desempenho.

Figura 3.12: Histograma dos valores da função-objetivo  $\Phi$  do Caso 1 do SSCP.



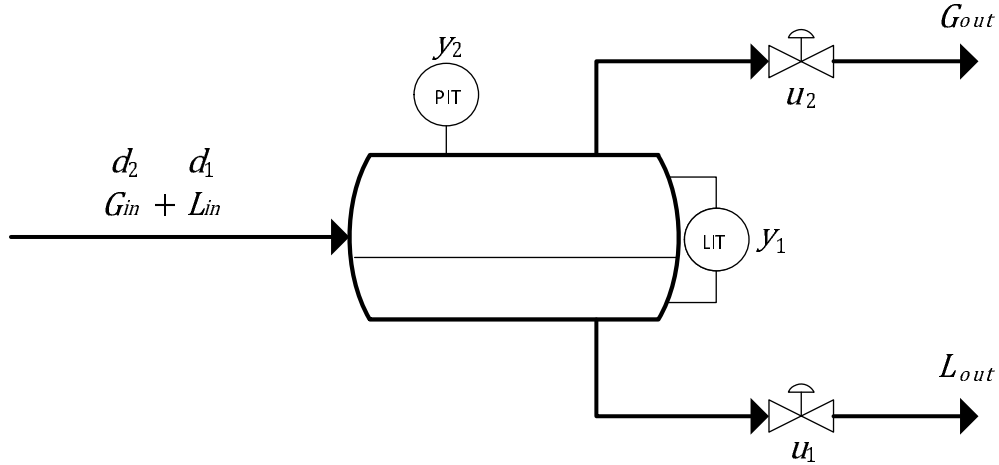
Fonte: o próprio autor.

## 3.2 Separador bifásico

O separador bifásico é um equipamento usado nas unidades de produção de petróleo, o qual recebe o óleo e gás dos poços das áreas de produção e tem a função de separá-los em duas correntes (ver Figura 3.13). Na produção de petróleo em plataformas marítimas existe a produção de poços em regime de golfadas, que para o separador bifásico se caracteriza na oscilação periódica (distúrbios não mensuráveis) nas vazões de alimentação de óleo ( $d_1$ ) e gás ( $d_2$ ), o que implica em perturbações no nível de líquido ( $y_1$ ) e na pressão ( $y_2$ ) do vaso, que são as PVs do processo. As MVs são as aberturas das válvulas nas correntes de descarga de líquido ( $u_1$ ) e de gás ( $u_2$ ). Na Tabela 3.7 estão dispostos os valores referentes a um estado estacionário das PVs e MVs, seus limites máximos e mínimos (utilizados na configuração das restrições da

lei de controle do MPC), bem como os valores nominais dos distúrbios e seus valores extremos esperados.

Figura 3.13: Esquema de um separador bifásico de óleo e gás com controles de pressão e nível.



Fonte: o próprio autor.

Tabela 3.7: Restrições rígidas das MVs e PVs, valores nominais dos distúrbios e SPs.

Variável	Valor inferior	SP ou valor nominal	Valor superior
$d_1 / (\text{m}^3 \cdot \text{s}^{-1})$	0,132	0,165	0,198
$d_2 / (\text{m}^3 \cdot \text{s}^{-1})$	0,132	0,165	0,198
$u_1 / \%$	0	50	100
$u_2 / \%$	0	50	100
$\Delta u_1 / \%$	-10	-	+10
$\Delta u_2 / \%$	-10	-	+10
$y_1 / \text{m}$	1,5	2,0	2,5
$y_2 / \text{bar}$	6	8	10

Em geral, não é necessário que o controle de nível do separador bifásico seja justo, ou seja, ele pode oscilar dentro dos limites considerados seguros e ótimos para operação. Na verdade, o separador deve funcionar como um filtro dos distúrbios que vêm dos poços, para que os processos envolvendo a separação de óleo e água, que estão a jusante, sejam estáveis. Isso acarreta, necessariamente, em uma flexibilização do controle do nível em seu SP. Assim, um dos critérios de desempenho deve ser

sintonizar o MPC de modo a minimizar a razão de amplitude (RA) entre a vazão de líquido que sai ( $A_{L_{out}}$ ) do separador e a que entra ( $A_{L_{in}}$  ou  $A_{d1}$ ) calculada por:

$$RA = \frac{A_{L_{out}}}{A_{L_{in}}}. \quad (3.3)$$

O valor de  $A_{L_{in}}$  pode ser previamente estimado por *softwares* que simulam o escoamento dinâmico multifásico nos dutos submarinos como o OLGA<sup>®</sup>. Com a planta existente e em operação é possível avaliar o valor de  $A_{L_{in}}$ , com o cálculo de  $L_{in}$  por meio de um balanço de massa (Equação 3.4) e utilizando uma fórmula que infere o valor de  $L_{out}$  (Equação 3.5), em m<sup>3</sup>/s :

$$A \frac{dy_1(t)}{dt} = L_{in}(t) - L_{out}(t), \quad (3.4)$$

$$L_{out}(t) = 0,246 \cdot u_1(t) \cdot \sqrt{\frac{y_2(t) - 6}{0,85}}, \quad (3.5)$$

com  $y_2(t)$ , em bar, e  $u_1(t)$  entre 0 e 1, sendo  $A$  a área da seção transversal do separador, o qual é função de  $y_1$  mas que pode ser aproximada por uma constante para fins de simplificação sem prejuízos aos resultados.

Por outro lado, a pressão do separador é mantida num valor tal que a produção de óleo seja maximizada e, portanto, o seu controle deve ser justo. Assim, o outro critério de desempenho é evitar que ocorra oscilações de grandes amplitudes nessa variável. Mudanças de *set-point*, apesar de não serem tão corriqueiras, podem acontecer na pressão do separador, por isso, a situação de controle simulada é uma variação no *set-point* da pressão de 8 bar para 7 bar no início da simulação enquanto que o SP do nível é mantido constante no valor de 2,0 m.

Desse modo a função-objetivo foi concebida como o Problema 3.2b. O motivo de normalizar o ITSE nesse critério é porque ele cresce arbitrariamente com  $t_{sim}$ , dado o distúrbio oscilatório contínuo considerado na simulação. O denominador da fração, que normaliza o ITSE, é o maior valor de ITSE possível, considerando que o erro máximo admissível para a pressão, em qualquer instante, seja de 3 bar (diferença entre o valor máximo de pressão e o novo SP). Além disso, a normalização também adimensionaliza o ITSE e o mantém na mesma ordem de grandeza do RA, que é adimensional. No Problema 3.2b optou-se por não incluir o esforço de controle

da válvula de gás  $\Delta u_2$  para que o controle de pressão fosse o mais justo o possível, contudo o seu valor está limitado pelas restrições rígidas, apresentadas na Tabela 3.7. Os limites dos parâmetros de sintonia  $\Omega_{\min}$  e  $\Omega_{\max}$ , que determinam a região de busca do PSO, são mostrados na Tabela 3.8.

Tabela 3.8: Região de busca dos parâmetros sintonizáveis do MPC.

Parâmetro ( $\Omega$ )	Limite inferior ( $\Omega_{\min}$ )	Limite superior ( $\Omega_{\max}$ )
$H_p$	1	500
$H_c$	1	6
$q_1$	0	1
$q_2$	0	1
$r_1$	0	1
$r_2$	0	1

Para efeitos de simulação da planta, as golfadas de líquido e gás são senóides com período de 30 minutos e com amplitude máxima igual a 20% do valor nominal (conforme a Tabela 3.7) e defasadas em fase por  $\pi$  rad/min, o tempo de simulação *tsim* foi de 250 instantes de amostragem, cujo intervalo de tempo é  $T = 1$  min, mesmo valor utilizado no estudo de caso anterior.

O modelo nominal linearizado utilizado neste estudo de caso, em unidades de engenharia, foi obtido a partir de uma dedução fenomenológica e está presente no livro de Nunes, Medeiros e Araújo (2010):

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \mathbf{G}(s) \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \\ D_1(s) \\ D_2(s) \\ Y_1(s) \\ Y_2(s) \end{bmatrix}, \quad (3.6)$$

em que  $\mathbf{G}(s)$  é uma matriz de funções de transferências dada por:

$$\mathbf{G}(s) = \begin{bmatrix} \frac{-103,9}{99,73 \cdot s + 1} & 0 & \frac{264,7}{99,73 \cdot s + 1} & 0 & 0 & \frac{-12,0}{99,73 \cdot s + 1} \\ \frac{-4,7}{0,42 \cdot s + 1} & \frac{-5,6}{0,42 \cdot s + 1} & \frac{12,1}{0,42 \cdot s + 1} & \frac{-0,1}{0,42 \cdot s + 1} & 0 & 0 \end{bmatrix} \quad (3.7)$$

com os parâmetros de tempo em minutos. As constantes de tempo de cada função de transferência de  $Y_1(s)$  foram obtidas de um mesmo conjunto de parâmetros e, portanto, para todos os efeitos elas representam a mesma variável, ou seja,  $\tau_1 = \tau_{1,1} = \tau_{1,3} = \tau_{1,6}$  (o mesmo vale para as constantes de tempo de  $Y_2(s)$ ,  $\tau_2 = \tau_{2,1} = \tau_{2,2} = \tau_{2,3} = \tau_{2,4} = \tau_{2,5}$ ). Sendo assim, para a determinação do modelo menos controlável considerou-se somente a incerteza de uma única constante de tempo, que afeta da mesma forma todas as funções de  $Y_1(s)$  (idem para  $Y_2(s)$ ). Como foi feito no estudo de caso anterior, se impôs uma faixa de  $\pm 10\%$  do valor nominal dos parâmetros do modelo com distribuição uniforme, conforme Tabela 3.9. Com algum algebrismo, é possível reescrever a Equação 3.6 de modo que as PVs apareçam somente no lado esquerdo da equação:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \mathbf{G}'(s) \cdot \begin{bmatrix} U_1(s) \\ U_2(s) \\ D_1(s) \\ D_2(s) \end{bmatrix}, \quad (3.8)$$

com:

$$\mathbf{G}'(s) = \begin{bmatrix} \frac{G_{1,1} + G_{1,6} \cdot G_{2,1}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{1,6} \cdot G_{2,2}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{1,3} + G_{1,6} \cdot G_{2,3}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{1,6} \cdot G_{2,4}}{1 - G_{1,6} \cdot G_{2,5}} \\ \frac{G_{2,1} + G_{2,5} \cdot G_{1,1}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{2,2}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{2,3} + G_{2,5} \cdot G_{1,3}}{1 - G_{1,6} \cdot G_{2,5}} & \frac{G_{2,4}}{1 - G_{1,6} \cdot G_{2,5}} \end{bmatrix}. \quad (3.9)$$

Com a normalização das PVs e MVs na faixa entre -1 e 1 e utilizando os valores dos parâmetros de  $\mathbf{G}^*(s)$  da Tabela 3.9 é possível obter a matriz  $\mathbf{G}^N(s)$ . Ela é apresentada abaixo mas foi dividida em duas partes devido ao seu tamanho:

$$\mathbf{G}^N(s) = \begin{bmatrix} \mathbf{G}_1^N(s) & \mathbf{G}_2^N(s) \end{bmatrix} \quad (3.10)$$

$$\mathbf{G}_1^N(s) = \begin{bmatrix} \frac{-0,85s - 0,62}{s^2 + 2,69s - 0,005} & \frac{1,985}{s^2 + 2,686s - 0,005} \\ \frac{-3,46s + 0,02}{s^2 + 2,7s - 0,005} & \frac{-4,123s - 0,03759}{s^2 + 2,686s - 0,005} \end{bmatrix} \quad (3.11)$$



$$\mathbf{G}_2^N(s) = \begin{bmatrix} \frac{0,19s^3 + 0,7s^2 + 0,55s + 0,004}{s^4 + 5,1s^3 + 6,5s^2 + 0,05s - 10^{-4}} & \frac{-0,1s - 0,3757}{s^3 + 5,1s^2 + 6,5s - 0,01} \\ \frac{0,51s^3 + 1,4s^2 - 0,0009s - 10^{-4}}{s^4 + 5,1s^3 + 6,5s^2 + 0,05s - 10^{-4}} & \frac{0,3s^2 + 0,8s + 0,007}{s^3 + 5,1s^2 + 6,5s - 0,01} \end{bmatrix} \quad (3.12)$$

Tabela 3.9: Limites dos valores dos parâmetros do modelo.

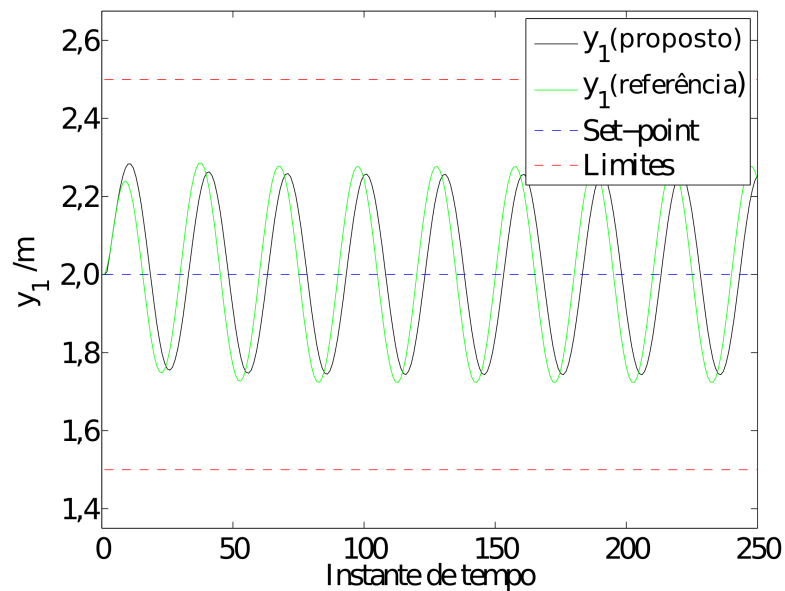
Parâmetro	Limite	Valor	Limite	Parâmetro
	superior	nominal	inferior	de $\mathbf{G}^*(s)$
$K_{1,1}$	-93,51	-103,9	-114,29	-93,51
$K_{1,3}$	291,17	264,7	238,23	264,7
$K_{1,6}$	-10,8	-12,0	-13,2	-13,2
$K_{2,1}$	-5,17	-4,7	-4,23	-5,17
$K_{2,2}$	-6,16	-5,6	-5,04	-6,16
$K_{2,4}$	13,31	12,1	10,89	12,1
$K_{2,3}$	13,31	12,1	10,89	12,1
$K_{2,5}$	-0,09	-0,1	-0,11	-0,09
$\tau_1$ /min	109,73	99,73	89,76	109,7
$\tau_2$ /min	0,462	0,42	0,378	0,378

As Figuras 3.14 a 3.17 mostram o comportamento dinâmico das PVs e MVs do modelo de difícil controle diante do caso de controle proposto, utilizando a sintonia encontrada e compara com o de uma sintonia de referência (SHRIDHAR; COOPER, 1998), em que a planta é o sistema menos controlável determinado e com o MPC configurado com o modelo nominal. Os valores dos parâmetros de sintonia calculados pelo método proposto neste trabalho e a sintonia de referência estão presentes na Tabela 3.10.

Tabela 3.10: Parâmetros de sintonia ótimos e de referência.

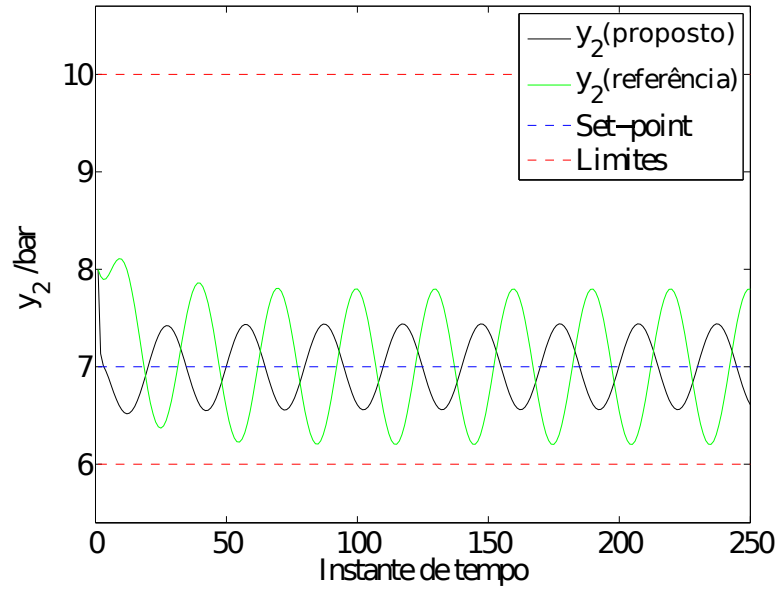
Parâmetro	Método ótimo proposto	Método de referência
$H_p$	44	500
$H_c$	1	100
$q_1$	0,7334	1
$q_2$	0,0946	1
$r_1$	0,1697	727,92
$r_2$	0,9991	45,57

Figura 3.14: Comportamento dinâmico de  $y_1$  para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência.



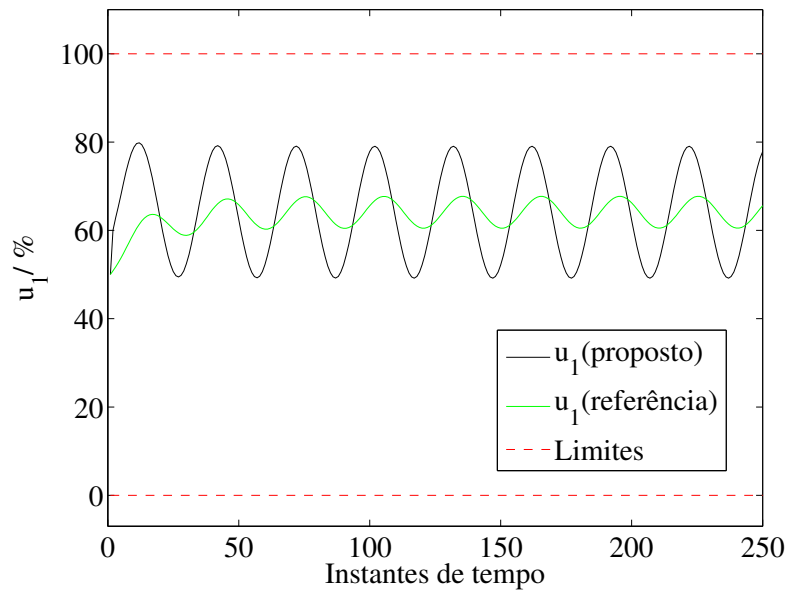
Fonte: o próprio autor.

Figura 3.15: Comportamento dinâmico de  $y_2$  para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência.



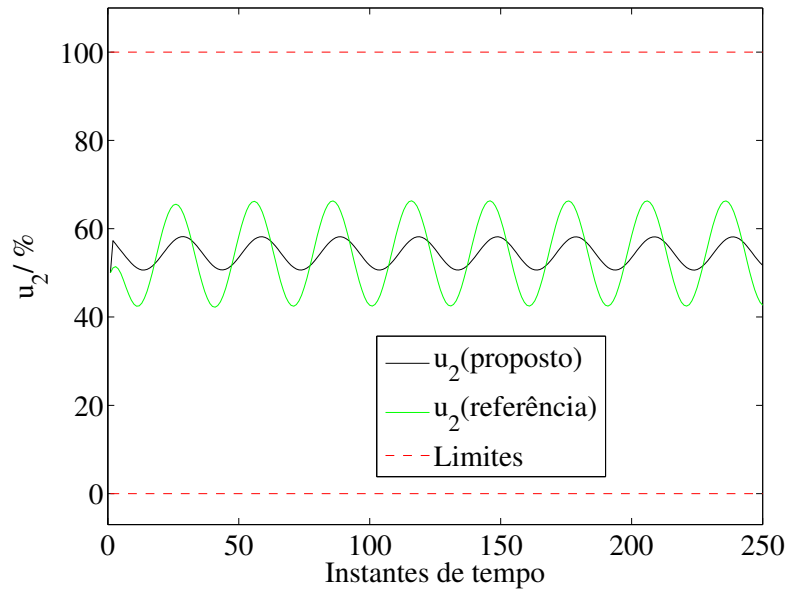
Fonte: o próprio autor.

Figura 3.16: Comportamento dinâmico de  $u_1$  para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência.



Fonte: o próprio autor.

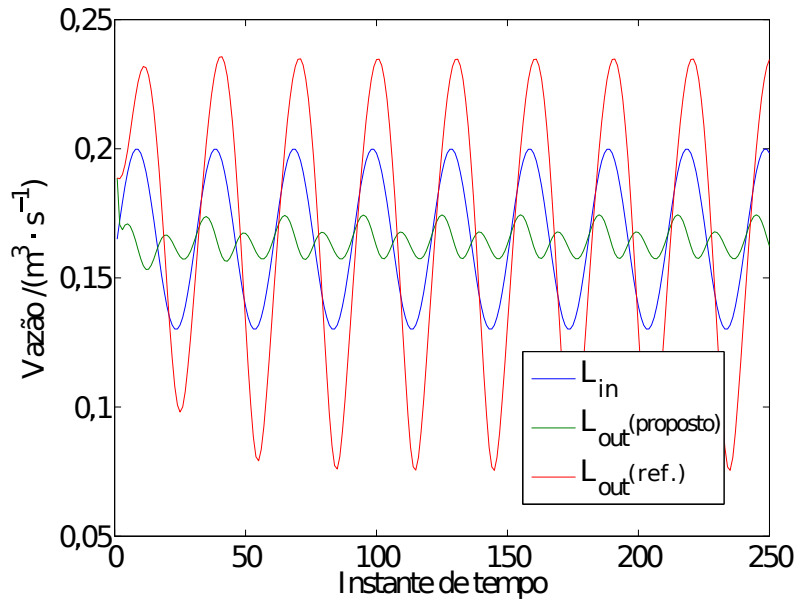
Figura 3.17: Comportamento dinâmico de  $u_2$  para a planta de difícil controle, utilizando a sintonia ótima obtida e a de referência.



Fonte: o próprio autor.

As oscilações observadas nos gráficos são consequência direta dos distúrbios senoidais que entram no sistema e nota-se que, em ambos os casos, o MPC sintonizado conseguiu rastrear a mudança de SP na pressão, contudo verifica-se uma variação maior nessa variável quando é utilizada a sintonia de referência. Com relação ao nível, não houve diferenças sensíveis na sua variação para as duas sintonias. Por sua vez, analisando os esforços de controle, tem-se uma equivalência entre as sintonias, pois a sintonia ótima apresentou uma variação maior em  $u_1$ , comparando-a com a de referência, e menor em  $u_2$ . Porém, a constatação mais importante está apresentada na Figura 3.18, a qual mostra que a sintonia ótima obtida possibilitou que o separador se comportasse como um filtro dos distúrbios gerados por  $d_1$ , pois ocorreu uma diminuição da amplitude na vazão de saída ( $L_{out}$ ), sendo o valor de  $RA = 0,2449$ . Por outro lado, a sintonia obtida pelo método de referência amplificou esses distúrbios, sendo  $RA = 2,28$ , neste caso.

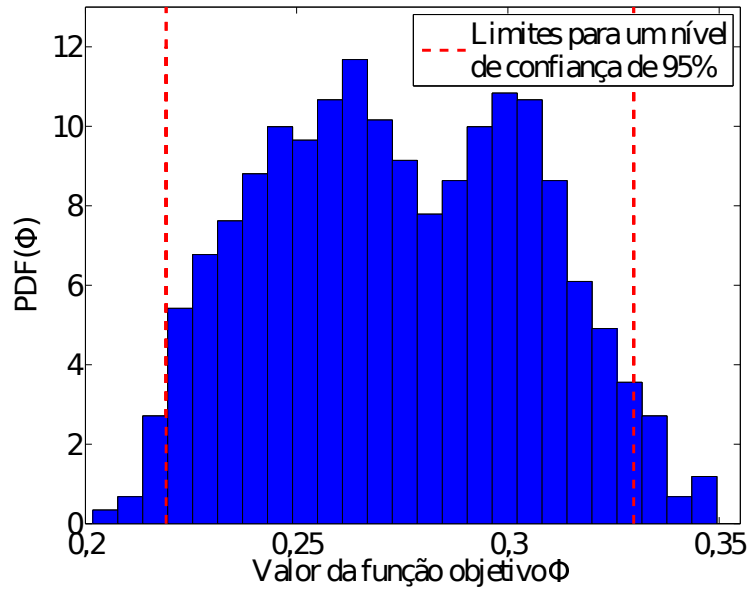
Figura 3.18: Comportamento dinâmico das vazões de líquido na entrada e na saída do separador, utilizando a sintonia ótima obtida e a de referência.



Fonte: o próprio autor.

A Figura 3.19 mostra a PDF aproximada do critério de desempenho, representado pela Equação 2.3, construída pela simulação de Monte Carlo, com os parâmetros de sintonia ótimos. Pelo histograma, é possível perceber que, mais uma vez, a sintonia se mostrou robusta diante da incertezas de modelagem, pois a dispersão do valor de  $\Phi$  é pequena, variando-se os parâmetros do modelo dentro das faixas consideradas na Tabela 3.9, e o controlador conseguiu o objetivo de filtrar os distúrbios ao mesmo tempo que manteve a pressão num valor justo. O valor da média de  $\Phi$  calculado foi de 0,274 com desvio padrão de 0,032, sendo que os limites para um nível de confiança de 95 % foram de 0,219 e 0,330, mostrados na figura como linhas verticais tracejadas.

Figura 3.19: Histograma representando a PDF aproximada da função-objetivo  $\Phi$  do separador de produção usando a sintonia ótima.



Fonte: o próprio autor.

### 3.3 Conclusões

Neste capítulo foram apresentados dois estudos de caso com o intuito de mostrar a resposta do método de sintonia ótima proposto. Os dois sistemas testados eram diferentes e com objetivos de controle diversos, de modo que pudesse ser verificada a flexibilidade do método. O SSCP apresenta um sistema com tempos mortos em que o MPC deve rastrear os SPs das PVs, da maneira mais rápida possível, com limitações mais estritas nas variações das MVs, simulando uma situação em que há uma mudança operacional da coluna de destilação. O exemplo do separador bifásico simula uma planta com distúrbios persistentes em que ocorre uma mudança no SP de uma PV e o MPC deve rastrear essa mudança ao mesmo tempo que aceita os distúrbios externos.

Os resultados mostram que o método proposto encontrou sintonias aderentes aos objetivos de ambos os casos, tanto em desempenho como em robustez, como pode ser observado pelo valor e a dispersão nos valores da função-objetivo nos estudos de caso realizados e a comparação com uma sintonia de referência.

---

# Capítulo 4

## Considerações finais

---

*“This is the end \ Beautiful friend \ This is the end \ My only friend, the end  
Of our elaborate plans, the end \ Of everything that stands, the end  
No safety or surprise, the end ”*

The end - The Doors

### 4.1 Contribuições

Este trabalho desenvolveu um método de sintonia ótima para controladores MPC com restrições, considerando as incertezas de modelagem, usando o PSO para solucionar o MINLP, de modo que fosse possível obter todos os parâmetros de sintonia, discretos e contínuos, em um único algoritmo.

Na grande maioria das vezes, os métodos de sintonia existentes são específicos em relação ao tipo de formulação do MPC, possuindo ainda como limitações a não consideração das restrições e não contemplando as incertezas relacionadas ao modelo utilizado pelo controlador. Alguns desses métodos propõem a sintonia de somente alguns dos parâmetros necessários do controlador preditivo, ficando os remanescentes por conta do engenheiro. Além disso, esses métodos são bastante inflexíveis, pois o usuário não tem a garantia de que a sintonia atenderá a uma demanda de controle previamente especificada. Em compensação, o método apresentado neste trabalho consegue flexibilizar a sintonia, ficando a cargo do usuário escolher um critério de desempenho que se encaixe às suas necessidades, o que pode ser comprovado pelas diferentes formas de critério de desempenho utilizadas nos estudos de caso realizados. Outra flexibilização é que o método em sua concepção não possui premissas

ou suposições que possam restringir o seu uso quanto ao tipo de MPC que pode ser sintonizado. O método desenvolvido possibilita ainda encontrar os melhores parâmetros de sintonia que atendem as exigências pretendidas, com algum grau de robustez às incertezas de modelagem, como mostram os resultados dos estudos de caso. Além disso, esses mesmos resultados apontam que o procedimento formulado para encontrar o modelo mais difícil de controlar, dentro do domínio da incerteza de modelagem, e a utilização desse modelo como referência contribuíram para que o método fosse eficaz em obter a sintonia com esse grau de robustez. A elaboração do algoritmo PSO permite que a simulação do controle e sintonia de todos os parâmetros fossem realizadas no mesmo ambiente, acelerando a obtenção dos resultados. Dessa maneira, pode-se concluir que os objetivos foram todos alcançados.

## 4.2 Trabalhos futuros

Com a realização deste trabalho, ficam em aberto algumas frentes de pesquisa, a saber:

- Fazer estudos de caso com outros tipos de controladores MPC, como os não-lineares, o IHMPC, preditivos-adaptativos, dentre outros.
- Pesquisar outros casos típicos na literatura que demandam diferentes critérios de desempenho, para testar o método e, por ventura, aprimorá-lo.
- Utilizar outras formas e métricas para determinar modelos de difícil controle, inclusive pesquisar métricas para sistemas não-lineares.
- Utilizar métodos de otimização híbridos para encontrar os parâmetros de sintonia ótimos. Dessa maneira, o PSO encontraria um conjunto de estimativas iniciais que poderia ser utilizado pelos tradicionais algoritmos de otimização MINLP.
- Aplicar o método desenvolvido em um problema de sintonia real.



---

# Apêndice A

## Códigos desenvolvidos

---

Neste apêndice estão descritos os dois códigos desenvolvidos em linguagem MATLAB<sup>®</sup> para o algoritmo PSO. A rotina *psoconsmix* é a função principal de PSO para resolver MINLP e a rotina *mincons* é uma sub-função que é usada para obter os valores mínimos das restrições, que são utilizados na função *psoconsmix* como fatores normalizadores. O algoritmo baseou-se na versão do PSO apresentada no artigo de Datta e Figueira (2011).

### A.1 Código da rotina *psoconsmix*

```
function [ y, fy, fev, fpar, par, bpar ] = psoconsmix(f,rb,n,...
                                                opcoes,cons,seed)
%PSOUNC Minimiza a funcao F fazendo buscas dentro da regio de busca RB.
%F pode ser funcao de varias variaveis, RB deve ser uma matriz com
%dimensoes Nx2, em que N e a quantidade de variaveis continuas as quais
%F depende. Cada linha da matriz RB e o intervalo de busca de uma
%variavel. F deve ser um arquivo do MATLAB contendo a funcao a ser
%minimizada.
%
%Exemplo:
%Quer-se minimizar a funcao F(x,y) = (x-2)^2 + y^2
%
%Deve-se criar uma funcao no MATLAB:
%
%function f=fun(p)
```

```

%x = p(:,1);
%y = p(:,2);
%f = (x-2)^2 + y^2;
%
%Em seguida cria-se a matriz de regioao de busca e usa-se a
%funcao PSOCONSMIX:
%
%regiao=[0 4;-2 2]
%psoconsmix(@ (x) fun(x),regiao)
%
%Nesse exemplo o valor de X estara dentro do intervalo de 0 a 4 e o
%valor de Y estara no intervalo de -2 a 2, devido a definicao de
%rb = [0 4;-2 2].
%
%O vetor OPCOES se nao for definido fara com a PSOUNC utilize os
%parametros padrao existentes na funcao. Se OPCOES for definida,
%ela deve ser um vetor com 6 posicoes:
%OPCOES =[niter npt wf wi c1 c2], niter e o numero de %iteracoes,
%npt e o numero de particulas, wf e o valor de inercia final,
%wi e o valor de inercia inicial, c1 e o valor da contribuicao
%individual, e c2 e o valor da contribuicao do grupo.
%
%y e a melhor partacula encontrada, fy e o menor valor da funcao
%objetivo encontrado, fev sao os menores valores de fy ao longo de cada
%iteracao, fpar e sao os valores da funcao objetivo de todas as
%particulas a cada iteracao, par sao os valores das particulas a cada
%iteracao e bpar sao os
%melhores valores encontrado por cada partacula a cada iteracao,
%bpar(:, :,end) sao os melhores valores encontrados por cada partacula.
%
%CONS e um m-file com as restricoes do problema, escritas da mesma
%forma que as restricoes das functions de otimizacao do matlab
rng(seed) %fixa a semente para os numeros pseudo aleatorios
if nargin<4

%opcoes padrao do PSO

niter    = 500;           %numero de iteracoes
npt      = 50;           %numero de particulas

```

```

wf      = 0.4;           %fator de inercia final
wi      = 0.9;           %fator de inercia
c1      = 2.0;           %contribuicao individual
c2      = 2.0;           %contribuicao do grupo

else
niter   = opcoes(1);    %numero de iteracoes
npt     = opcoes(2);    %numero de particulas
wf      = opcoes(3);    %fator de inercia final
wi      = opcoes(4);    %fator de inercia inicial
c1      = opcoes(5);    %contribuicao individual
c2      = opcoes(6);    %contribuicao do grupo
end

if isempty(rb)==1
    error(message('A funcao PSO necessita de uma matriz definindo as...
                  regioes de busca dos parametros'));
end

pm      = 0.15;         %probabilidade de mutacao
m       = size(rb,1);   %numero de parametros
c       = m-n;         %numero de variaveis continuas

for j=1:n
bin=dec2bin(rb(j+c,2)); %numero binarios
bits(j)=length(bin);   %contar o numero de bits
end

nd      = sum(bits);    %tamanho da dimensao dos inteiros
vmax(1:c) = 0.5*(rb(1:c,2)-rb(1:c,1));%definicao da velocidade...
%maxima como sendo metade do range de cada variavel
vmax(c+1:c+nd) = 2*(ones(1,nd));%definicao da velocidade dos...
%numeros inteiros como 2(a velocidade dos numeros so pode ser -1, 0, 1
k       = 1;           %iteracao
fotm    = realmax*(ones(1,npt));%valor inicial...
%da funcao objetivo
vmaxr   = repmat(vmax',1,npt);%vetor vmax...
%repetido varias npt vezes

```

```

rb1r          = repmat (rb(:,1),1,npt);%coluna 1 do vetor rb...
%repetida varias vezes somente variaveis continuas
rb2r          = repmat (rb(:,2),1,npt);%coluna 2 do vetor rb...
%repetida varias vezes somente variaveis continuas
mzeros        = zeros (nd,npt);%matriz de zeros para comparacao
mum           = mzeros+1;%matriz de uns para comparacao
dq            = [1 1 1 1];%fator de adaptacao
eta           = unifrnd(25,45);%expoente de adaptacao
paru          = [0.15 wi c1 c2];%valor superior dos parametros
pari          = [0 0 0 0];%valor inferior dos parametros
w             = wi;%valor inicial da inercia
cont          = 0;%contador auxiliar
pio rp        = 0;%valor inicial de f para a pior particula
parada        = 0;%valor inicial do contador para a parada prematura
prematu ra    = round(0.2*niter);%quantidade de vezes para que a...
%objetivo nao deve mudar para o algoritmo parar

if isempty(cons)==0
[normEQ,normINEQ]=mincons (rb,n, constr);
normEQ  =abs (normEQ);%valores para a normalizacao das restricoes...
%de igualdade
normINEQ=abs (normINEQ);%valores para a normalizacao das restricoes...
%de inequacao
end

for j=1:n
    for j1=1:bits(j)
        deco(j1+cont)=2^(bits(j)-j1);%construcao do vetor de...
        %decodificacao dos inteiros
    end
    cont      = j1+cont;%atualizacao do contador
    posii(j) = cont-bits(j)+1;%vetor com as posicoes iniciais de cada...
    %numero inteiro dentro da matriz xbin
    posif(j) = cont;%vetor com as posicoes finais de cada numero...
    %inteiro dentro da matriz xbin
end

deco          = repmat (deco',1,npt);%matriz de decodificacao
decoaux       = deco(posii,:);%matriz com os maiores valores de...

```

```

%decodificacao

%sorteio dos parametros iniciais
x(1:c, :, k) = rblr(1:c, :) + 2 * rand(c, npt) .* vmaxr(1:c, :);
xbin(1:nd, :, k) = round(rand(nd, npt));

%adicionado somente para que o horizonte de predicacao e o horizonte de
%controle nao tenham valores iguais a 0
for j=1:n
    pos0(j, :) = logical(sum(xbin(posii(j):posif(j), :)));
    xbin(posif(j), ~pos0(j, :)) = 1;
end

for j=1:n
    xint(j, :) = sum(deco(posii(j):posif(j), :) ...
        .* xbin(posii(j):posif(j), :));
end
xint = round(xint);
x(c+1:c+n, :, k) = xint;

%sorteio das velocidades iniciais
v(:, :, k) = rand(c+nd, npt) .* 0; %segundo o artigo todas as...
%velocidades iniciais sao 0

p1=x;%vetor dos melhores pontos auxiliar
f1=fotm;%vetor dos melhores valores de funcao objetivo auxiliar
f2=true(1, npt);%vetor que diz se as solucoes anteriores sao...
%infasiveis 1 e infasivel
absx=abs(x(:, :, k-1));% foi utilizado somente pq o MPC nao aceita...
%valores negativos de pesos
parfor o=1:npt
    [resEQ(1, o), resINEQ(1, o)] = cons(absx(:, o));%valor inicial...
    %das restricoes
end

while k<=niter && parada<=prematura
%atualizacao do contador
k=k+1;

```

```

%Construcao de um variavel de x auxiliar para a comparacao e informacao
%das particulas que estao fora dos limites
xaux=x(:, :, k-1);
posinf=xaux>=rb1r;%diz as posicoes que estao acima de rb1r
possup=xaux<=rb2r;%diz as posicoes que estao abaixo de rb2r
poseq = resEQ==0;%diz as posicoes que estao fora das restricoes...
%de igualdade
posineq=resINEQ<=0;%diz as posicoes que esta fora das restricoes...
%de inequacao

xnorm=(xaux-rb1r)./(rb2r-rb1r);%valores normalizados de x para...
%as penalidades
unipos=~posinf|~possup;%diz as posicoes que estao fora da...
%regiao de busca
pinfa=logical(sum(double(unipos))|~poseq|~posineq;%diz quais...
%particulas que sao infasiveis
penalidadelim=abs(xnorm.*double(~posinf|~possup));%diz a penalidade...
%referente a cada posicao de cada particula em relacao aos limites
penalidadeEQ =abs(resEQ.*double(~poseq))./normEQ;%diz a penalidade...
%referente a cada posicao de cada particula em relacao as restricoes...
%de igualdade
penalidadeINEQ =abs(resINEQ.*double(~poseq))./normINEQ;%diz...
%a penalidade referente a cada posicao de cada particula em relacao...
%as restricoes de igualdade
somapenalidade=sum(penalidadelim)+penalidadeEQ+penalidadeINEQ;%faz...
%a soma das penalidades

%funcao a ser minimizada
absx=abs(x(:, :, k-1)); % foi utilizado somente pq o MPC nao aceita...
%valores negativos de pesos
parfor o=1:npt
    fotm(k,o) = f(absx(:, o));
end
if isempty(f1(~f2))==0 %para evitar que piorp seja matriz vazia ,
caso inicialmente se f1(~f2) for vazia.
piorp=max(f1(~f2));
end

```

```

fud=fotm(k);
fotm(k,pinfa)=piorp+somapenalidade(pinfa);%valor da funcao objetivo ...
%das particulas infasiveis
fotmm=fotm(k,:);
pos=fotm(k,:) < f1;%posicoes onde fotm foi menor que f1
f1(~pinfa & f2)= fotm(k,~pinfa & f2);%Atualizacao de f1 caso a...
%particula encontrou algum ponto fasivel agora e...
%nunca tinha encontrado
f1(pos & ~pinfa & ~f2)=fotm(k,pos & ~pinfa & ~f2);%Atualizacao de f1...
%caso a particula encontrou algum ponto fasivel...
%em algum momento
f1(pos & pinfa & f2) = fotm(k,pos & pinfa & f2);%Atualizacao de f1...
%caso a particula nunca encontrou algum ponto fasivel

z1(:,~pinfa & f2)= x(:,~pinfa & f2,k-1);%Atualizacao de z1 caso a...
%particula encontrou algum ponto fasivel agora e nunca tinha encontrado
z1(:,pos & ~pinfa & ~f2)=x(:,pos & ~pinfa & ~f2,k-1);%Atualizacao...
%de z1 caso a particula encontrou algum ponto fasivel em algum momento
z1(:,pos & pinfa & f2)=x(:,pos & pinfa & f2,k-1);%Atualizacao de...
%z1 caso a particula nunca encontrou algum ponto fasivel

f2=pinfa & f2;
%atualizacao dos pontos que encontraram um ponto fasivel

%escolha da melhor estimativa por particula
p=z1;
pk(:, :, k-1)=z1;
%melhor estimativa encontrada
expospgfa=isempty(min(f1(~f2)));
if expospgfa==1
    pospg=f1==min(f1(f2)); %saber se existe o dado
    pglobalinfa = p(1:size(rb,1),pospg);
    pglobal=pglobalinfa;
    fu(k-1) = min(f1(f2));
    fudd(k-1)= fud(pospg);
    num=1:npt;
    num=num(pospg);
else
    pospg=f1==min(f1(~f2)); %saber se existe o dado

```

```

pglobalfa = p(1:size(rb,1),pospg);
pglobal=pglobalfa;
fu(k-1) = min(f1(~f2));
fudd(k-1)= fud(pospg);
num=1:npt;
num=num(pospg);
end
%Logica para parada prematura do algoritmo
if k==2      %k sempre >=2
    parada=0;
elseif fu(k-1)==fu(k-2) %se o valor da funcao objetivo nao ...
    %atualizar soma-se ao contador
    parada=parada+1;
else
    parada=0;%se o valor da funcao objetivo for...
    %atualizado retoma-se a contagem
end

%manter pglobal com uma estimativa somente, evitando que tenha mais...
%de um valor caso 2 particulas cheguem a mesma resposta
pglobal = pglobal(:,1);
num = num(:,1);
%transformar as variaveis inteiras em binarias
paux = p(c+1:c+n,:);
pgaux = repmat(pglobal(c+1:c+n,:),1,npt);
for j=1:n
    for j1=1:npt
        pint(posii(j):posif(j),j1) = dec2bin(paux(j,j1),bits(j))'*1-48;
        pgint(posii(j):posif(j),j1) = dec2bin(pgaux(j,j1),bits(j))'*1-48;
    end
end
%atualizacao das velocidades
%atualizacao da parte continua
r1=rand(1);
r2=rand(1);

v(1:c,:,k) = w.*v(1:c,:,k-1) + c1.*r1.*(p(1:c,:)-x(1:c,:,k-1))...
+c2.*r2.*(repmat(pglobal(1:c,:),1,npt)-x(1:c,:,k-1));

```



```

%atualizacao da parte inteira
xbiniaux=xbin(:, :, k-1);
r3=rand(1,24);           %probabilidade randomica para os casos
vaux=v(c+1:c+nd, :, k-1); %velocidade auxiliar
p1=w*vaux;
p2=c1*r1*(pint-xbiniaux);
p3=c2*r2*(pgint-xbiniaux);

vaux(xbiniaux==mzeros & p1==mzeros & p2==mzeros &
p3==mzeros & r3(1)<=pm) = mum(xbiniaux==mzeros & p1==mzeros & ...
p2==mzeros & p3==mzeros & r3(1)<=pm);
vaux(xbiniaux==mzeros & p1==mzeros & p2==mzeros & p3>mzeros
& r3(2)>=0.5) = mum(xbiniaux==mzeros & p1==mzeros & p2==mzeros &
p3>mzeros & r3(2)>=0.5);
vaux(xbiniaux==mzeros & p1==mzeros & p2>mzeros &
p3==mzeros & r3(3)>=0.5) = mum(xbiniaux==mzeros & p1==mzeros & p2>mzeros
& p3==mzeros & r3(3)>=0.5);
vaux(xbiniaux==mzeros & p1==mzeros & p2>mzeros & p3>mzeros
& r3(4)>pm) = mum(xbiniaux==mzeros & p1==mzeros & p2>mzeros &
p3>mzeros & r3(4)>pm);
vaux(xbiniaux==mzeros & p1> mzeros & p2==mzeros &
p3==mzeros & r3(5)<0.5) = mzeros(xbiniaux==mzeros & ...
p1> mzeros & p2==mzeros & p3==mzeros & r3(5)<0.5);
vaux(xbiniaux==mzeros & p1> mzeros & p2==mzeros & p3>mzeros
& r3(6)<=pm) = mzeros(xbiniaux==mzeros & p1> mzeros & p2==mzeros &
p3>mzeros & r3(6)<=pm);
vaux(xbiniaux==mzeros & p1> mzeros & p2>mzeros &
p3==mzeros & r3(7)<=pm) = mzeros(xbiniaux==mzeros & p1> mzeros & ...
p2>mzeros & p3==mzeros & r3(7)<=pm);
vaux(xbiniaux==mzeros & p1> mzeros & p2>mzeros & p3>mzeros
& r3(8)<=pm) = mzeros(xbiniaux==mzeros & p1> mzeros & p2>mzeros &
p3>mzeros & r3(8)<=pm);
vaux(xbiniaux==mzeros & p1< mzeros & p2==mzeros &
p3==mzeros) = round(r3(9)) *mum(xbiniaux==mzeros...
& p1< mzeros & p2==mzeros & p3==mzeros);
vaux(xbiniaux==mzeros & p1< mzeros & p2==mzeros &
p3>mzeros) = round(r3(10)) *mum(xbiniaux==mzeros...
& p1< mzeros & p2==mzeros & p3>mzeros);

```

```

vaux(xbinaux==mzeros & p1< mzeros & p2>mzeros &
p3==mzeros) = round(r3(11))*mum(xbinaux==mzeros...
& p1< mzeros & p2>mzeros & p3==mzeros);
vaux(xbinaux==mzeros & p1< mzeros & p2>mzeros &
p3>mzeros) = round(r3(12))*mum(xbinaux==mzeros...
& p1< mzeros & p2>mzeros & p3>mzeros);

vaux(xbinaux==mum & p1==mzeros & p2<mzeros & p3<mzeros
& r3(13)>pm) = -1*mum(xbinaux==mum & p1==mzeros & p2<mzeros &
p3<mzeros & r3(13)>pm);
vaux(xbinaux==mum & p1==mzeros & p2<mzeros &
p3==mzeros & r3(14)>=0.5) = -1*mum(xbinaux==mum & p1==mzeros &...
p2<mzeros & p3==mzeros & r3(14)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros & p3<mzeros
& r3(15)>=0.5) = -1*mum(xbinaux==mum & p1==mzeros & p2==mzeros &
p3<mzeros & r3(15)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros &
p3==mzeros & r3(16)<=pm) = -1*mum(xbinaux==mum & p1==mzeros &...
p2==mzeros & p3==mzeros & r3(16)<=pm);
vaux(xbinaux==mum & p1> mzeros & p2<mzeros & p3<mzeros)
= -1*round(r3(17))*mum(xbinaux==mum & p1>mzeros & p2<mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros & p2<mzeros &
p3==mzeros) = -1*round(r3(18))*mum(xbinaux==mum & p1> mzeros &...
p2<mzeros & p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros & p3<mzeros)
= -1*round(r3(19))*mum(xbinaux==mum & p1> mzeros & p2==mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros) = mzeros(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros & r3(20)<=pm) = -1*mum(xbinaux==mum & p1> mzeros &...
p2==mzeros & p3==mzeros & r3(20)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2<mzeros & p3<mzeros
& r3(21)<=pm) = mzeros(xbinaux==mum & p1< mzeros & p2<mzeros &
p3<mzeros & r3(21)<=pm);

```

```

vaux(xbinaux==mum & p1< mzeros & p2<mzeros    &
p3==mzeros & r3(22)<=pm)    = mzeros(xbinaux==mum & p1< mzeros &...
p2<mzeros    & p3==mzeros & r3(22)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2==mzeros    & p3<mzeros
& r3(23)<=pm)    = mzeros(xbinaux==mum & p1< mzeros & p2==mzeros    &
p3<mzeros & r3(23)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2==mzeros    &
p3==mzeros & r3(24)<0.5)    = mzeros(xbinaux==mum & p1< mzeros &...
p2==mzeros    & p3==mzeros & r3(24)<0.5);

%atualizacao da velocidade
v(c+1:c+nd, :, k)    = vaux;

%Atualizacao dos pontos das particulas
x(1:c, :, k)        = x(1:c, :, k-1) + v(1:c, :, k);
xbin(1:nd, :, k)    = xbin(1:nd, :, k-1) + vaux;

%randomizacao da melhor particula
iaux=x(1:c, :, k);
r5=rand(1, c);

pgrandom=rb(1:c, 1)+2*rand(c, 1).*vmax(1:c)';
iaux(r5<0.05, num) = pgrandom(r5<0.05);
x(1:c, :, k)=iaux;

r6=rand(1, nd);
iaux=xbin(1:nd, :, k);
iaux(r6<0.05, num) = ~iaux(r6<0.05, num);
vaux(r6<0.05, num) = 0;
xbin(1:nd, :, k)=iaux;

%adicionado somente para que o horizonte de predicao e o horizonte de
%controle nao tenham valores iguais a 0
for j=1:n
    pos0(j, :)    = logical(sum(xbin(posii(j):posif(j), :, k)));
    xbin(posif(j), ~pos0(j, :), k)=1;
end
for j=1:n
    xint(j, :)    = sum(deco(posii(j):posif(j), :)).*...

```

```

        xbin(posii(j):posif(j),:,k));
end
x(c+1:c+n, :,k) = xint;

%adaptacao dos parametros
r4=rand(1,4);

par=[pm w c1 c2];

dq(r4<0.5)=(2.*r4(r4<0.5)+(1-2.*r4(r4<0.5)).*...
((paru(r4<0.5)-par(r4<0.5))./(paru(r4<0.5)-pari(r4<0.5))).^...
(eta+1)).^(1/(eta+1))-1;
dq(r4>=0.5)=1-(2*(1-r4(r4>=0.5))+(2*r4(r4>=0.5)-1).*...
((-pari(r4>=0.5)+par(r4>=0.5))./(paru(r4>=0.5)-pari(r4>=0.5))).^...
(eta+1)).^(1/(eta+1)));
par=par+(paru-pari).*dq;
pm=par(1);
w=par(2);
c1=par(3);
c2=par(4);

end
if nargout==1
    y=pglobal;
elseif nargout==2
    y=pglobal;
    fy=min(min(f1(~f2)));
elseif nargout==3
    y=pglobal;
    fy=min(min(f1(~f2)));
    fev=fu;
elseif nargout==4
    y=pglobal;
    fy=min(min(f1(~f2)));
    fev=fu;
    fpar=fotm(2:end,:);
elseif nargout==5
    y=pglobal;
    fy=min(min(f1(~f2)));

```

```

    fev=fu;
    fpar=fotm(2:end,:);
    par=x;
else
    y=pglobal;
    fy=min(min(f1(~f2)));
    fev=fu;
    fpar=fotm(2:end,:);
    par=x;
    bpar=pk;
end

```

## A.2 Código da rotina *mincons*

```

function [normEQ,normINEQ]=mincons(rb,n,constr)
[testeEQ,testeINEQ]=fval(constr,rb(:,1));
numEQ=length(testeEQ);
numINEQ=length(testeINEQ);

niter=100;
npt=5;
wi=1.0;%fator de inercia
c1=1.0;%contribuicao individual
c2=2.0;%contribuicao do grupo
pm=0.15;%probabilidade de mutacao
m=size(rb,1);%numero de parametros
c=m-n;%numero de variaveis contnuas

for j=1:n
bin=dec2bin(rb(j+c,2));%numero binarios
bits(j)=length(bin);%contar o numero de bits
end
nd=sum(bits);%tamanho da dimensao dos inteiros
vmax(1:c)=0.5*(rb(1:c,2)-rb(1:c,1));%definicao da velocidade...
%maxima como sendo metade do range de cada variavel
vmax(c+1:c+nd)=2*(ones(1,nd));%definicao da velocidade dos ...

```

```

%numerosinteiros como 2(a velocidade dos numeoros so pode ser...
%-1, 0, 1
vmaxr          = repmat(vmax',1,npt);%vetor vmax repetido varias...
%npt vezes
rb1r           = repmat(rb(:,1),1,npt);%coluna 1 do vetor rb ...
%repetida varias vezes somente variaveis contnuas
rb2r           = repmat(rb(:,2),1,npt);%coluna 2 do vetor rb ...
%repetida varias vezes somente variaveis contnuas
mzeros        = zeros(nd,npt);%matriz de zeros para comparacao
mum            = mzeros+1;%matriz de uns para comparacao
eta            = unifrnd(25,45);%expoente de adaptacao
paru          = [0.15 wi c1 c2];%valor superior dos parametros
pari           = [0 0 0 0];%valor inferior dos parametros
cont=0;%contador auxiliar

for j=1:n
    for j1=1:bits(j)
        deco(j1+cont)=2^(bits(j)-j1);%construcao do vetor de...
        %decodificacao dos inteiros
    end
    cont      = j1+cont;%atualizacao do contador
    posii(j) = cont-bits(j)+1;%vetor com as posicoes iniciais de...
    %cada numero inteiro dentro da matriz xbin
    posif(j) = cont; %vetor com as posicoes...
    %finais de cada numero inteiro dentro da matriz xbin
end

deco          = repmat(deco',1,npt);%matriz de decodificacao
decoaux       = deco(posii,:);%matriz com os maiores...
%valores de decodificacao

for contEQ=1:numEQ
    k          = 1;%iteracao
    fotm       = realmax*(ones(1,npt));%valor inicial...
    %da funcao objetivo
    dq         = [1 1 1 1];%fator de adaptacao
    w          = wi;%valor inicial da inercia
    clear x xbin

```

```

%sorteio dos parametros iniciais
x(1:c, :, k) = rblr(1:c, :)+2*rand(c, npt).*vmaxr(1:c, :);
xbin(1:nd, :, k) = round(rand(nd, npt));

for j=1:n
    xint(j, :) = sum(deco(posii(j):posif(j), :).*...
    xbin(posii(j):posif(j), :)); %decodificacao dos...
    %numeros binarios em inteiros
end
xint=round(xint);
x(c+1:c+n, :, k) = xint;

%sorteio das velocidades velocidades iniciais
v(:, :, k) = rand(c+nd, npt).*0;%segundo o artigo todas as velocidades...
%iniciais sao 0
p1=x;%vetor dos melhores pontos auxiliar
f1=fotm;%vetor dos melhores valores de funcao objetivo auxiliar
f2=true(1, npt);%vetor que diz se as solucoes anteriores sao ...
%infasiveis, 1 e infasivel

while k<=niter
%atualizacao do contador
k=k+1;

%Construcao de um variavel de x auxiliar para a comparacao e...
%informacao das particulas que estao fora dos limites
xaux=x(:, :, k-1);
posinf=xaux>=rblr;%diz as posicoes que estao acima de rblr
possup=xaux<=rb2r;%diz as posicoes que estao abaixo de rb2r
xnorm=(xaux-rblr)./(rb2r-rblr);%valores normalizados de x...
%para as penalidades
unipos=~posinf|~possup;%diz as posicoes que estao fora da...
%regiao de busca
pinfa=logical(sum(double(unipos)));%diz quais as particulas...
%que sao infasiveis
penalidade=abs(xnorm.*double(unipos));%diz a penalidade...
%referente a cada posicao de cada particula
somapenalidade=sum(penalidade);%faz a soma das penalidades

```

```

%funcao a ser minimizada
absx=x(:, :, k-1); % foi utilizado somente pq o MPC nao aceita valores...
%negativos de pesos
parfor o=1:npt
    [EQ, INEQ] = fval(constr, absx(:, o)); %variavel para avaliar a...
    %quantidade de restricoes
    fotm(k, o) = -abs(EQ);
end
if isempty(f1(~f2))==0 %para evitar que piorp seja matriz vazia...
%, caso inicialmente se f1(~f2) for vazia.
piorp=max(f1(~f2));
end

fotm(k, pinfa)=piorp+somapenalidade(pinfa); %valor da funcao...
%objetivo das partttculas infasiveis
fotmm=fotm(k, :);
pos=fotm(k, :) < f1; %posicoes onde fotm foi menor que f1
f1(~pinfa & f2)=fotm(k, ~pinfa & f2); %Atualizacao de f1 caso a...
%particula encontrou algum ponto fasivel agora e nunca tinha encontrado
f1(pos & ~pinfa & ~f2)=fotm(k, pos & ~pinfa & ~f2); %Atualizacao de f1...
%caso a particula encontrou algum ponto fasivel em algum momento
f1(pos & pinfa & f2) = fotm(k, pos & pinfa & f2); %Atualizacao de f1...
%caso a particula nunca encontrou algum ponto fasivel

z1(:, ~pinfa & f2)= x(:, ~pinfa & f2, k-1); %Atualizacao de z1 caso a...
%particula encontrou algum ponto fasivel agora e nunca tinha encontrado
z1(:, pos & ~pinfa & ~f2)=x(:, pos & ~pinfa & ~f2, k-1); %Atualizacao de...
%z1 caso a particula encontrou algum ponto fasivel em algum momento
z1(:, pos & pinfa & f2)=x(:, pos & pinfa & f2, k-1); %Atualizacao de z1...
%caso a particula nunca encontrou algum ponto fasivel

f2=pinfa & f2; %atualizacao dos pontos que encontraram um ponto fasivel

%escolha da melhor estimativa por partcula
p=z1;
pk(:, :, k-1)=z1;
%melhor estimativa encontrada

```



```

expospgfa=isempty(min(f1(~f2)));
if expospgfa==1
pospg=f1==min(f1(f2));           %saber se existe o dado
pglobalinfa = p(1:size(rb,1),pospg);
pglobal=pglobalinfa;
fu(k-1) = min(f1(f2));
num=1:npt;
num=num(pospg);
else
pospg=f1==min(f1(~f2));           %saber se existe o dado
pglobalfa = p(1:size(rb,1),pospg);
pglobal=pglobalfa;
fu(k-1) = min(f1(~f2));
num=1:npt;
num=num(pospg);
end

%manter pglobal com uma estimativa somente, evitando que tenha
%mais de um valor caso 2 particulas cheguem a mesma resposta
pglobal = pglobal(:,1);
num = num(:,1);
%transformar as variaveis inteiras em binarias
paux = p(c+1:c+n,:);
pgaux = repmat(pglobal(c+1:c+n,:),1,npt);
for j=1:n
    for j1=1:npt
        pint(posii(j):posif(j),j1) = dec2bin(paux(j,j1),bits(j))'*1-48;
        pgint(posii(j):posif(j),j1) = dec2bin(pgaux(j,j1),bits(j))'*1-48;
    end
end
%atualizacao das velocidades
%atualizacao da parte contnua
r1=rand(1);
r2=rand(1);

v(1:c,:,k) = w.*v(1:c,:,k-1) + c1.*r1.*(p(1:c,:)-x(1:c,:,k-1))+...
c2.*r2.*(repmat(pglobal(1:c,:),1,npt)-x(1:c,:,k-1));

%atualizacao da parte inteira

```

```

xбинаux=xbin(:, :, k-1);
r3=rand(1,24);           %probabilidade randomica para os casos
vaux=v(c+1:c+nd, :, k-1); %velocidade auxiliar
p1=w*vaux;
p2=c1*r1*(pint-xбинаux);
p3=c2*r2*(pgint-xбинаux);

vaux(xбинаux==mzeros & p1==mzeros & p2==mzeros &
p3==mzeros & r3(1)<=pm) = mum(xбинаux==mzeros & p1==mzeros &...
p2==mzeros & p3==mzeros & r3(1)<=pm);
vaux(xбинаux==mzeros & p1==mzeros & p2==mzeros & p3>mzeros
& r3(2)>=0.5) = mum(xбинаux==mzeros & p1==mzeros & p2==mzeros &
p3>mzeros & r3(2)>=0.5);
vaux(xбинаux==mzeros & p1==mzeros & p2>mzeros &
p3==mzeros & r3(3)>=0.5) = mum(xбинаux==mzeros & p1==mzeros & p2>mzeros
& p3==mzeros & r3(3)>=0.5);
vaux(xбинаux==mzeros & p1==mzeros & p2>mzeros & p3>mzeros
& r3(4)>pm) = mum(xбинаux==mzeros & p1==mzeros & p2>mzeros &
p3>mzeros & r3(4)>pm);
vaux(xбинаux==mzeros & p1>mzeros & p2==mzeros &
p3==mzeros & r3(5)<0.5) = mzeros(xбинаux==mzeros & p1>mzeros &...
p2==mzeros & p3==mzeros & r3(5)<0.5);
vaux(xбинаux==mzeros & p1>mzeros & p2==mzeros & p3>mzeros
& r3(6)<=pm) = mzeros(xбинаux==mzeros & p1>mzeros &...
p2==mzeros & p3>mzeros & r3(6)<=pm);
vaux(xбинаux==mzeros & p1>mzeros & p2>mzeros &
p3==mzeros & r3(7)<=pm) = mzeros(xбинаux==mzeros & p1>mzeros &...
p2>mzeros & p3==mzeros & r3(7)<=pm);
vaux(xбинаux==mzeros & p1>mzeros & p2>mzeros & p3>mzeros
& r3(8)<=pm) = mzeros(xбинаux==mzeros & p1>mzeros & p2>mzeros &
p3>mzeros & r3(8)<=pm);
vaux(xбинаux==mzeros & p1<mzeros & p2==mzeros &
p3==mzeros) = round(r3(9))*mum(xбинаux==mzeros &...
p1<mzeros & p2==mzeros & p3==mzeros);
vaux(xбинаux==mzeros & p1<mzeros & p2==mzeros &
p3>mzeros) = round(r3(10))*mum(xбинаux==mzeros &...
p1<mzeros & p2==mzeros & p3>mzeros);
vaux(xбинаux==mzeros & p1<mzeros & p2>mzeros &
p3==mzeros) = round(r3(11))*mum(xбинаux==mzeros &...

```

```

p1< mzeros & p2>mzeros & p3==mzeros);
vaux(xbinaux==mzeros & p1< mzeros & p2>mzeros &
p3>mzeros) = round(r3(12))*mum(xbinaux==mzeros &...
p1< mzeros & p2>mzeros & p3>mzeros);

vaux(xbinaux==mum & p1==mzeros & p2<mzeros & p3<mzeros
& r3(13)>pm) = -1*mum(xbinaux==mum & p1==mzeros & p2<mzeros &
p3<mzeros & r3(13)>pm);
vaux(xbinaux==mum & p1==mzeros &...
p2<mzeros & p3==mzeros & r3(14)>=0.5) = -1*mum(xbinaux==mum &...
p1==mzeros & p2<mzeros & p3==mzeros & r3(14)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros & p3<mzeros
& r3(15)>=0.5) = -1*mum(xbinaux==mum & p1==mzeros & p2==mzeros &
p3<mzeros & r3(15)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros &
p3==mzeros & r3(16)<=pm) = -1*mum(xbinaux==mum & p1==mzeros &...
p2==mzeros & p3==mzeros & r3(16)<=pm);
vaux(xbinaux==mum & p1> mzeros & p2<mzeros & p3<mzeros)
= -1*round(r3(17))*mum(xbinaux==mum & p1>mzeros & p2<mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros &...
p2<mzeros & p3==mzeros) = -1*round(r3(18))*mum(xbinaux==mum &...
p1> mzeros & p2<mzeros & p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros & p3<mzeros)
= -1*round(r3(19))*mum(xbinaux==mum & p1> mzeros & p2==mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros) = mzeros(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros & r3(20)<=pm) = -1*mum(xbinaux==mum & p1> mzeros &...
p2==mzeros & p3==mzeros & r3(20)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2<mzeros & p3<mzeros
& r3(21)<=pm) = mzeros(xbinaux==mum & p1< mzeros & p2<mzeros &
p3<mzeros & r3(21)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2<mzeros &
p3==mzeros & r3(22)<=pm) = mzeros(xbinaux==mum & p1< mzeros &...
p2<mzeros & p3==mzeros & r3(22)<=pm);

```

```

vaux(xbinaux==mum & p1< mzeros & p2==mzeros & p3<mzeros
& r3(23)<=pm) = mzeros(xbinaux==mum & p1< mzeros & p2==mzeros &
p3<mzeros & r3(23)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2==mzeros &
p3==mzeros & r3(24)<0.5) = mzeros(xbinaux==mum & p1< mzeros &...
p2==mzeros & p3==mzeros & r3(24)<0.5);

%atualizacao da velocidade
v(c+1:c+nd, :, k) = vaux;

%Atualizacao dos pontos das partculas
x(1:c, :, k) = x(1:c, :, k-1) + v(1:c, :, k);
xbin(1:nd, :, k) = xbin(1:nd, :, k-1) + vaux;

%randomizacao da melhor partcula
xaux=x(1:c, :, k);
r5=rand(1, c);

pgrandom=rb(1:c, 1)+2*rand(c, 1).*vmax(1:c)';
xaux(r5<0.05, num) = pgrandom(r5<0.05);
x(1:c, :, k)=xaux;

r6=rand(1, nd);
xaux=xbin(1:nd, :, k);
xaux(r6<0.05, num) = ~xaux(r6<0.05, num);
vaux(r6<0.05, num) = 0;
xbin(1:nd, :, k)=xaux;

for j=1:n
    xint(j, :) = sum(deco(posii(j):posif(j), :)...
    .*xbin(posii(j):posif(j), :, k));
end
x(c+1:c+n, :, k) = xint;

%adaptacao dos parametros
r4=rand(1, 4);

par=[pm w c1 c2];

```

```

dq(r4<0.5)=(2.*r4(r4<0.5)+(1-2.*r4(r4<0.5)).*...
((paru(r4<0.5)-par(r4<0.5))./(paru(r4<0.5)-pari(r4<0.5))).^...
(eta+1)).^(1/(eta+1))-1;
dq(r4>=0.5)=1-(2*(1-r4(r4>=0.5))+(2*r4(r4>=0.5)-1).*...
((-pari(r4>=0.5)+par(r4>=0.5))./(paru(r4>=0.5)-pari(r4>=0.5))).^...
(eta+1)).^(1/(eta+1));
par=par+(paru-pari).*dq;
pm=par(1);
w=par(2);
c1=par(3);
c2=par(4);
end

fy=min(min(f1(~f2)));
normEQ(contEQ)=fy;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

for contINEQ=1:numINEQ;
k          = 1;%iteracao
fotm       = realmax*(ones(1,npt));%valor inicial da...
%funcao objetivo
dq         = [1 1 1 1];%fator de adaptacao
w          = wi;%valor inicial da inercia
clear x xbin
%sorteio dos parametros iniciais
x(1:c, :, k) = rblr(1:c, :)+2*rand(c, npt).*vmaxr(1:c, :);
xbin(1:nd, :, k) = round(rand(nd, npt));

for j=1:n
    xint(j, :) = sum(deco(posii(j):posif(j), :)).*...
    xbin(posii(j):posif(j), :)); %decodificacao dos numeros...
    %binarios em inteiros
end
xint=round(xint);
x(c+1:c+n, :, k) = xint;

%sorteio das velocidades velocidades iniciais

```

```

v(:, :, k) = rand(c+nd, npt) .* 0; %segundo o artigo todas as...
%velocidades iniciais sao 0
p1=x;%vetor dos melhores pontos auxiliar
f1=fotm;%vetor dos melhores valores de funcao objetivo auxiliar
f2=true(1, npt); %vetor que diz se as solucoes anteriores...
%sao infasiveis, 1 e infasivel

while k<=niter
%atualizacao do contador
k=k+1;

%Construcao de um variavel de x auxiliar para a comparacao e...
%informacao das particulas que estao fora dos limites
xaux=x(:, :, k-1);
posinf=xaux>=rb1r;%diz as posicoes que estao acima de rb1r
possup=xaux<=rb2r;%diz as posicoes que estao abaixo de rb2r
xnorm=(xaux-rb1r)./(rb2r-rb1r); %valores normalizados de x...
%para as penalidades
unipos=~posinf|~possup;%diz as posicoes que estao fora da...
%regiao de busca
pinfa=logical(sum(double(unipos))); %diz quais as particulas...
%que sao infasiveis
penalidade=abs(xnorm.*double(unipos)); %diz a penalidade...
%referente a cada posicao de cada particula
somapenalidade=sum(penalidade); %faz a soma das penalidades

%funcao a ser minimizada
absx=x(:, :, k-1); % foi utilizado somente pq o MPC nao aceita valores...
%negativos de pesos
parfor o=1:npt
    [EQ, INEQ] =fval(constr, absx(:, o)); %variavel para avaliar a...
    %quantidade de restricoes
    fotm(k, o) = -abs(INEQ(cont INEQ));
end
if isempty(f1(~f2))==0 %para evitar que piorp seja matriz vazia ,
caso inicialmente se f1(~f2) for vazia.
piorp=max(f1(~f2));
end

```

```

fotm(k,pinfa)=piorp+somapenalidade(pinfa);%valor da funcao objetivo...
%das partttculas infasiveis
fotmm=fotm(k,:);
pos=fotm(k,:) < f1;%posicoes onde fotm foi menor que f1
f1(~pinfa & f2)= fotm(k,~pinfa & f2);%Atualizacao de f1 caso a...
%particula encontrou algum ponto fasivel agora e nunca tinha encontrado
f1(pos & ~pinfa & ~f2)=fotm(k,pos & ~pinfa & ~f2);%Atualizacao de f1...
%caso a particula encontrou algum ponto fasivel em algum momento
f1(pos & pinfa & f2) = fotm(k,pos & pinfa & f2);%Atualizacao de f1...
%caso a particula nunca encontrou algum ponto fasivel

z1(:,~pinfa & f2)= x(:,~pinfa & f2,k-1);%Atualizacao de z1 caso a...
%particula encontrou algum ponto fasivel agora e nunca tinha encontrado
z1(:,pos & ~pinfa & ~f2)=x(:,pos & ~pinfa & ~f2,k-1);%Atualizacao de...
%z1 caso a particula encontrou algum ponto fasivel em algum momento
z1(:,pos & pinfa & f2)=x(:,pos & pinfa & f2,k-1);%Atualizacao de z1...
%caso a particula nunca encontrou algum ponto fasivel

f2=pinfa & f2;%atualizacao dos pontos que encontraram um ponto fasivel

%escolha da melhor estimativa por partcula
p=z1;
pk(:, :, k-1)=z1;
%melhor estimativa encontrada
expospgfa=isempty(min(f1(~f2)));
if expospgfa==1
pospg=f1==min(f1(f2)); %saber se existe o dado
pglobalinfa = p(1:size(rb,1),pospg);
pglobal=pglobalinfa;
fu(k-1) = min(f1(f2));
num=1:npt;
num=num(pospg);
else
pospg=f1==min(f1(~f2)); %saber se existe o dado
pglobalfa = p(1:size(rb,1),pospg);
pglobal=pglobalfa;
fu(k-1) = min(f1(~f2));
num=1:npt;

```

```

num=num(pospg);
end
%Logica para parada prematura do algoritmo
if k==2          %k sempre >=2
    parada=0;
elseif fu(k-1)==fu(k-2)%se o valor da funcao objetivo nao...
%atualizar soma-se ao contador
    parada=parada+1;
else
    parada=0;          %se o valor da funcao objetivo...
    %for atualizado retoma-se a contagem
end

%manter pglobal com uma estimativa somente, evitando que...
%tenha mais de um valor caso 2 particulas cheguem a mesma resposta
pglobal = pglobal(:,1);
num    = num(:,1);
%transformar as variaveis inteiras em binarias
paux  = p(c+1:c+n,:);
pgaux = repmat(pglobal(c+1:c+n,:),1,npt);
for j=1:n
    for j1=1:npt
        pint(posii(j):posif(j),j1) = dec2bin(paux(j,j1),bits(j))*1-48;
        pgint(posii(j):posif(j),j1) = dec2bin(pgaux(j,j1),bits(j))*1-48;
    end
end
%atualizacao das velocidades
%atualizacao da parte contnua
r1=rand(1);
r2=rand(1);

v(1:c,:,k) = w.*v(1:c,:,k-1) + c1.*r1.*(p(1:c,:)-x(1:c,:,k-1))...
+c2.*r2.*(repmat(pglobal(1:c,:),1,npt)-x(1:c,:,k-1));

%atualizacao da parte inteira
xbinaux=xbin(:,:,k-1);
r3=rand(1,24);          %probabilidade randomica para os casos
vaux=v(c+1:c+nd,:,k-1); %velocidade auxiliar
p1=w*vaux;

```



```

p2=c1*r1*(pint-xbinaux);
p3=c2*r2*(pgint-xbinaux);

vaux(xbinaux==mzeros & p1==mzeros & p2==mzeros &
p3==mzeros & r3(1)<=pm) = mum(xbinaux==mzeros & p1==mzeros &...
p2==mzeros & p3==mzeros & r3(1)<=pm);
vaux(xbinaux==mzeros & p1==mzeros & p2==mzeros & p3>mzeros
& r3(2)>=0.5) = mum(xbinaux==mzeros & p1==mzeros & p2==mzeros &
p3>mzeros & r3(2)>=0.5);
vaux(xbinaux==mzeros & p1==mzeros & p2>mzeros &
p3==mzeros & r3(3)>=0.5) = mum(xbinaux==mzeros & p1==mzeros & p2>mzeros
& p3==mzeros & r3(3)>=0.5);
vaux(xbinaux==mzeros & p1==mzeros & p2>mzeros & p3>mzeros
& r3(4)>pm) = mum(xbinaux==mzeros & p1==mzeros & p2>mzeros &
p3>mzeros & r3(4)>pm);
vaux(xbinaux==mzeros & p1>mzeros & p2==mzeros &
p3==mzeros & r3(5)<0.5) = mzeros(xbinaux==mzeros & p1>mzeros &...
p2==mzeros & p3==mzeros & r3(5)<0.5);
vaux(xbinaux==mzeros & p1>mzeros & p2==mzeros & p3>mzeros
& r3(6)<=pm) = mzeros(xbinaux==mzeros & p1>mzeros & p2==mzeros &
p3>mzeros & r3(6)<=pm);
vaux(xbinaux==mzeros & p1>mzeros & p2>mzeros &
p3==mzeros & r3(7)<=pm) = mzeros(xbinaux==mzeros & p1>mzeros &...
p2>mzeros & p3==mzeros & r3(7)<=pm);
vaux(xbinaux==mzeros & p1>mzeros & p2>mzeros & p3>mzeros
& r3(8)<=pm) = mzeros(xbinaux==mzeros & p1>mzeros & p2>mzeros &
p3>mzeros & r3(8)<=pm);
vaux(xbinaux==mzeros & p1<mzeros & p2==mzeros &
p3==mzeros) = round(r3(9))*mum(xbinaux==mzeros &...
p1<mzeros & p2==mzeros & p3==mzeros);
vaux(xbinaux==mzeros & p1<mzeros & p2==mzeros &
p3>mzeros) = round(r3(10))*mum(xbinaux==mzeros &...
p1<mzeros & p2==mzeros & p3>mzeros);
vaux(xbinaux==mzeros & p1<mzeros & p2>mzeros &
p3==mzeros) = round(r3(11))*mum(xbinaux==mzeros &...
p1<mzeros & p2>mzeros & p3==mzeros);
vaux(xbinaux==mzeros & p1<mzeros & p2>mzeros &
p3>mzeros) = round(r3(12))*mum(xbinaux==mzeros &...
p1<mzeros & p2>mzeros & p3>mzeros);

```

```

vaux(xbinaux==mum & p1==mzeros & p2<mzeros & p3<mzeros
& r3(13)>pm) = -1*mum(xbinaux==mum & p1==mzeros & p2<mzeros &
p3<mzeros & r3(13)>pm);
vaux(xbinaux==mum & p1==mzeros & p2<mzeros &
p3==mzeros & r3(14)>=0.5) = -1*mum(xbinaux==mum & p1==mzeros &...
p2<mzeros & p3==mzeros & r3(14)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros & p3<mzeros
& r3(15)>=0.5) = -1*mum(xbinaux==mum & p1==mzeros & p2==mzeros &
p3<mzeros & r3(15)>=0.5);
vaux(xbinaux==mum & p1==mzeros & p2==mzeros &
p3==mzeros & r3(16)<=pm) = -1*mum(xbinaux==mum & p1==mzeros &...
p2==mzeros & p3==mzeros & r3(16)<=pm);
vaux(xbinaux==mum & p1> mzeros & p2<mzeros & p3<mzeros)
= -1*round(r3(17))*mum(xbinaux==mum & p1>mzeros & p2<mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros & p2<mzeros &
p3==mzeros) = -1*round(r3(18))*mum(xbinaux==mum & p1> mzeros &...
p2<mzeros & p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros & p3<mzeros)
= -1*round(r3(19))*mum(xbinaux==mum & p1> mzeros & p2==mzeros &
p3<mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros) = mzeros(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros);
vaux(xbinaux==mum & p1> mzeros & p2==mzeros &
p3==mzeros & r3(20)<=pm) = -1*mum(xbinaux==mum & p1> mzeros &...
p2==mzeros & p3==mzeros & r3(20)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2<mzeros & p3<mzeros
& r3(21)<=pm) = mzeros(xbinaux==mum & p1< mzeros & p2<mzeros &
p3<mzeros & r3(21)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2<mzeros &
p3==mzeros & r3(22)<=pm) = mzeros(xbinaux==mum & p1< mzeros &...
p2<mzeros & p3==mzeros & r3(22)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2==mzeros & p3<mzeros
& r3(23)<=pm) = mzeros(xbinaux==mum & p1< mzeros & p2==mzeros &
p3<mzeros & r3(23)<=pm);
vaux(xbinaux==mum & p1< mzeros & p2==mzeros &
p3==mzeros & r3(24)<0.5) = mzeros(xbinaux==mum & p1< mzeros &...

```

```

p2==mzeros & p3==mzeros & r3(24)<0.5);

%atualizacao da velocidade
v(c+1:c+nd, :, k) = vaux;

%Atualizacao dos pontos das partculas
x(1:c, :, k) = x(1:c, :, k-1) + v(1:c, :, k);
xbin(1:nd, :, k) = xbin(1:nd, :, k-1) + vaux;

%randomizacao da melhor partcula
iaux=x(1:c, :, k);
r5=rand(1, c);

pgrandom=rb(1:c, 1)+2*rand(c, 1).*vmax(1:c)';
iaux(r5<0.05, num) = pgrandom(r5<0.05);
x(1:c, :, k)=iaux;

r6=rand(1, nd);
iaux=xbin(1:nd, :, k);
iaux(r6<0.05, num) = ~iaux(r6<0.05, num);
vaux(r6<0.05, num) = 0;
xbin(1:nd, :, k)=iaux;

for j=1:n
    xint(j, :) = sum(deco(posii(j):posif(j), :)...
        .*xbin(posii(j):posif(j), :, k)); %decodificacao dos...
        %numeros binarios em inteiros
end
x(c+1:c+n, :, k) = xint;

%adaptacao dos parametros
r4=rand(1, 4);

par=[pm w c1 c2];

dq(r4<0.5)=(2.*r4(r4<0.5)+(1-2.*r4(r4<0.5)).*...
((paru(r4<0.5)-par(r4<0.5))./(paru(r4<0.5)-pari(r4<0.5))).^...
(eta+1)).^(1/(eta+1))-1;
dq(r4>=0.5)=1-(2*(1-r4(r4>=0.5))+(2*r4(r4>=0.5)-1).*...

```

```
((-pari(r4>=0.5)+par(r4>=0.5))./(paru(r4>=0.5)-pari(r4>=0.5))).^...  
(eta+1)).^(1/(eta+1));  
par=par+(paru-pari).*dq;  
pm=par(1);  
w=par(2);  
c1=par(3);  
c2=par(4);  
end  
fy=min(min(f1(~f2)));  
normINEQ(contINEQ)=fy;  
end
```

---

# Referências Bibliográficas

---

AGACHI, P. S.; NAGY, Z. K.; CRISTEA, M. V.; IMRE-LUCACI, Á. *Model Based Control*. 1. ed. [S.l.]: Wiley-VCH, 2006. 290 p.

AGUIRRE, L. A. *Introdução à identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais*. 3. ed. [S.l.]: UFMG, 2007. 730 p.

AL-GHAZZAWI, A; ALI, E.; NOUH A.; ZAFIRIOU E. On-line tuning strategy for model predictive controllers. *Journal of Process Control*, v. 11, n. 3, p. 265 – 284, 2001.

ALI, E. Heuristic on-line tuning for nonlinear model predictive controllers using fuzzy logic. *Journal of Process Control*, v. 13, n. 5, p. 383 – 396, 2003.

ARAÚJO, M. A. S.; BRASIL, N. Índio do; SOUZA, C. M. de. *Processamento de petróleo e gás*. [S.l.]: LTC, 2011.

ARRUDA, L. V.; SWIECH, M.C.; NEVES, F.; DELGADO, M. Um método evolucionário para sintonia de controladores. *Revista Controle & Automação*, v. 19, n. 1, 2008.

ÅSTRÖM, K. J. A.; HÄGGLUND, T. *PID Controllers: Theory, Design, and Tuning*. 2nd. ed. [S.l.]: Instrument Society of America, 1995. 354 p. ISBN 1-55617-516-7.

BADGWELL, T. personal communications, 2008.

BAGHERI, P.; KHAKI-SEDIGH, A. An analytical tuning approach to multivariable model predictive controllers. *Journal of Process Control*, v. 24, n. 12, p. 41 – 54, 2014.

BIPM; IEC; IFCC; ILAC; ISO; IUPAC; IUPAP; OIML. *Evaluation of measurement data — Supplement 1 to the “Guide to the expression of uncertainty in*

measurement” — *Propagation of distributions using a Monte Carlo method*. [S.l.]: Joint Committee for Guides in Metrology, 2008. 90 p.

BURSALI, N.; AKAY B.; ERTUNÇ S.; HAPOGLU H.; ALPBAZ M. New tuning method for generalized predictive control of the production of *S. cerevisiae*. *Food and Bioproducts Processing*, v. 79, n. 1, p. 27 – 34, 2001.

CAMACHO E., F.; BORDONS, C. *Model Predictive Control*. 2. ed. [S.l.]: Springer, 2004. 405 p.

CLARKE, D.; MOHTADI, C.; TUFFS, P. Generalized predictive control - Part I. The basic algorithm. *Automatica*, v. 23, n. 2, p. 137–148, 1987.

CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control – a computer control algorithm. *AIChE National Meeting*, Pergamon Press, Inc., Houston, TX, USA, abr. 1979.

DATTA, D.; FIGUEIRA, J. R. A real-integer-discrete-coded particle swarm optimization for design problems. *Applied Soft Computing*, Elsevier B.V., v. 11, n. 4, p. 3625–3633, jun. 2011.

DAVTYAN, A.; HOFFMANN, S.; SCHEURING, R. Optimization of model predictive control by means of sequential parameter optimization. In: *IEEE Symposium on Computational Intelligence in Control and Automation*. Paris: [s.n.], 2011. p. 11 – 16.

DUAN, Y.; HARLEY, R.; HABETLER, T. Comparison of particle swarm optimization and genetic algorithm in the design of permanent magnet motors. In: *Power Electronics and Motion Control Conference, 2009. IPEMC '09. IEEE 6th International*. [S.l.: s.n.], 2009. p. 822–825.

FERRAMOSCA, A.; LIMON D.; GONZÁLEZ A.H.; ODLOAK D.; CAMACHO E.F. MPC for tracking zone regions. *Journal of Process Control*, v. 20, n. 4, p. 506 – 516, 2010.

FISHER, D. G. Process control: An overview and personal perspective. *The Canadian Journal of Chemical Engineering*, Wiley Subscription Services, Inc., A Wiley Company, v. 69, n. 1, p. 5–26, 1991.

- GAING, Z.-L. A particle swarm optimization approach for optimum design of PID controller in AVR system. *IEEE Transactions on Energy Conversion*, v. 19, n. 2, p. 384–391, jun. 2004.
- GARRIGA, J. L.; SOROUSH, M. Model predictive controller tuning via eigenvalue placement. In: *American Control Conference, 2008*. [S.l.: s.n.], 2008. p. 429–434.
- GARRIGA, J. L.; SOROUSH, M. Model predictive control tuning methods: a review. *Industrial & Engineering Chemistry Research*, v. 49, n. 8, p. 3505–3515, abr. 2010.
- GOUS, G.; VAAL, P. de. Using MV overshoot as a tuning metric in choosing DMC move suppression values. *ISA Transactions*, v. 51, n. 5, p. 657 – 664, 2012.
- GROSSMANN, I. Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, Kluwer Academic Publishers, v. 3, n. 3, p. 227–252, 2002.
- HAN, K.; ZHAO, J.; QIAN, J. A novel robust tuning strategy for model predictive control. *Control and Automation, 2006. WCICA*, p. 6406–6410, 2006.
- HU, X.; EBERHART, R. Solving constrained nonlinear optimization problems with particle swarm optimization. *Biomedical Engineering*, v. 5, n. 11, p. 2–5, 2002.
- KENNEDY, J.; EBERHART, R. Particle swarm optimization. *Proceedings of ICNN'95 - International Conference on Neural Networks*, Ieee, v. 4, p. 1942–1948, 1995.
- KITAYAMA, S.; YASUDA, K. A method for mixed integer programming problems by particle swarm optimization. *Electrical Engineering in Japan*, v. 157, n. 2, p. 40–49, nov. 2006.
- LASKARI, E.; PARSOPOULOS, K.; VRAHATIS, M. Particle swarm optimization for integer programming. *Proceedings of the 2002 Congress on Evolutionary Computation. CEC'02 (Cat. No.02TH8600)*, Ieee, v. 2, p. 1582–1587, 2002.
- LEE, J.; YU, Z. Tuning of model predictive controllers for robust performance. *Computers & Chemical Engineering*, v. 18, n. 1, p. 15 – 37, 1994. An International Journal of Computer Applications in Chemical Engineering.

LEE, J. H. van der; SVRCEK, W. Y.; YOUNG, B. R. A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making. *ISA transactions*, v. 47, n. 1, p. 53–9, jan. 2008.

LEE, K. H.; HUANG, B.; TAMAYO, E. C. Sensitivity analysis for selective constraint and variability tuning in performance assessment of industrial mpc. *Control Engineering Practice*, v. 16, n. 10, p. 1195 – 1215, 2008.

LJUNG, L. *System Identification: Theory for the User*. 2. ed. [S.l.]: Prentice Hall, 1999. 672 p.

LOVBJERG, M.; KRINK, T. Extending particle swarm optimisers with self-organized criticality. *CEC'02. Proceedings of the 2002*, p. 1588–1593, 2002.

MACIEJOWSKI, J. M. *Predictive control: with constraints*. 1. ed. [S.l.]: Prentice Hall, 2000. 352 p.

MARQUIS, P.; BROUSTAIL, J. P. SMOC, a bridge between state space and model predictive controllers: Application to the automation of a hydrotreating unit. *Proceedings of the 1988 IFAC workshop on model based process control*, Oxford: Pergamon Press, p. 37–43, 1988.

MORARI, M. Design of resilient processing plants—III: A general framework for the assessment of dynamic resilience. *Chemical Engineering Science*, v. 38, n. 11, p. 1881–1891, 1983.

NERY JÚNIOR, G. A.; MARTINS, M. A.; KALID, R. A PSO-based optimal tuning strategy for constrained multivariable predictive controllers with model uncertainty. *ISA Transactions*, v. 53, n. 2, p. 560 – 567, 2014.

NUNES, G. C.; MEDEIROS, J. L. de; ARAÚJO, O. de Q. F. *Modelagem e controle na produção de petróleo: Aplicações em MATLAB*. São Paulo, Brasil: Edgard Blücher, 2010.

OLIVEIRA, J.; BOAVENTURA-CUNHA J.; MOURA OLIVEIRA P.B.; FREIRE H. A swarm intelligence-based tuning method for the sliding mode generalized predictive control. *ISA Transactions*, v. 53, n. 5, p. 1501 – 1515, 2014. ICCA 2013.



- POLI, R.; KENNEDY, J.; BLACKWELL, T. Particle swarm optimization. *Swarm Intelligence*, v. 1, n. 1, p. 33–57, ago. 2007.
- PRETT, D. M.; MORARI, M. Introduction to shell control problem. In: *The Shell Process Control Workshop*. [S.l.]: Butterworth-Heinemann, 1987. p. 351 – 362.
- QI, Z.; FISHER, D. Robust stability of model predictive control. *American Control Conference, 1994*, v. 3, p. 3258 – 3262, 1994.
- RANI, K.; UNBEHAUEN, H. Study of predictive controller tuning methods. *Automatica*, v. 33, n. 12, p. 2243 – 2248, 1997.
- RAWLINGS, J.; MUSKE, K. The stability of constrained receding horizon control. *Automatic Control, IEEE Transactions on*, v. 38, n. 10, p. 1512–1516, Oct 1993.
- REGA, R.; PRATIHAR D. K. Particle swarm optimization algorithm vs genetic algorithm to develop integrated scheme for obtaining optimal mechanical structure and adaptive controller of a robot. *Intelligent Control & Automation*, v. 2, n. 4, p. 430, 2011.
- REVERTER, C. M.; IBARROLA, J.; CANO-IZQUIERDO, J. M. Tuning rules for a quick start up in dynamic matrix control. *ISA Transactions*, v. 53, n. 2, p. 612 – 627, 2014.
- RICHALET, J.; RAULT, A.; TESTUD, J. L.; PAPON, J. Model predictive heuristic control: Application to industrial processes. *Automatica*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 14, n. 5, p. 413–428, set. 1978.
- SANTORO, B. F.; ODLOAK, D. Closed-loop stable model predictive control of integrating systems with dead time. *Journal of Process Control*, v. 22, n. 7, p. 1209 – 1218, 2012.
- SARHADI, P.; SALAHSHOOR, K.; KHAKI-SEDIGH, A. Robustness analysis and tuning of generalized predictive control using frequency domain approaches. *Applied Mathematical Modelling*, v. 36, n. 12, p. 6167 – 6185, 2012.
- SCHWARTZ, J.; RIVERA, D. Simulation-based optimal tuning of model predictive control policies for supply chain management using simultaneous perturbation stochastic approximation. *2006 American Control Conference*, Ieee, p. 6 pp., 2006.

SHANNON, C. Communication in the presence of noise. *Proceedings of the IEEE*, v. 86, n. 2, p. 447–457, Feb 1998.

SHI, Y.; EBERHART, R. Parameter selection in particle swarm optimization. In: PORTO, V. et al. (Ed.). *Evolutionary Programming VII*. [S.l.]: Springer Berlin Heidelberg, 1998, (Lecture Notes in Computer Science, v. 1447). p. 591–600.

SHRIDHAR, R.; COOPER, D. J. A tuning strategy for unconstrained multivariable model predictive control. v. 5885, n. 98, p. 4003–4016, 1998.

SKOGESTAD, S. Apresentação, *Flowsheet Controllability assessment tools*. 2001.

SKOGESTAD, S.; POSTLETHWAITE, I. *Multivariable feedback control: analysis and design*. 2. ed. [S.l.]: John Wiley & Sons, 2007.

SMITH, O. Close control loops with dead time. *Chemical Engineering Progress*, v. 53, p. 217–219, 1957.

SUZUKI, R.; KAWAI, F.; NAKAZAWA, C.; MATSUI, T.; AIYOSHI, E. Parameter optimization of model predictive control by PSO. *Electrical Engineering in Japan*, v. 178, n. 1, p. 40–49, jan. 2012.

TRIERWEILER, J.; FARINA, L. RPN tuning strategy for model predictive control. *Journal of Process Control*, v. 13, n. 7, p. 591 – 598, 2003.

VLACHOS, C.; WILLIAMS, D.; GOMM, J. B. Solution to the Shell standard control problem using genetically tuned PID controllers. v. 10, p. 151–163, 2002.

WOJSZNIS, W.; GUDAZ, J.; BLEVINS, T.; MEHTA, A. Practical approach to tuning MPC. *ISA transactions*, v. 42, n. 1, p. 149–62, jan. 2003.

UFBA  
UNIVERSIDADE FEDERAL DA BAHIA  
ESCOLA POLITÉCNICA

PROGRAMA DE PÓS GRADUAÇÃO EM ENGENHARIA INDUSTRIAL - PEI

Rua Aristides Novis, 02, 6º andar, Federação, Salvador BA  
CEP: 40.210-630  
Telefone: (71) 3283-9800  
E-mail: [pei@ufba.br](mailto:pei@ufba.br)  
Home page: <http://www.pei.ufba.br>

