



**Sintonia de Controle Preditivo pela  
Técnica de Enxame de Partículas:  
Aplicação a Uma Coluna  
Desmetanizadora**

João Corrêa Lima Neto

**Projeto de Final de Curso**

Orientador

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

**Junho de 2011**

# SINTONIA DE CONTROLE PREDITIVO PELA TÉCNICA DE ENXAME DE PARTÍCULAS: APLICAÇÃO A UMA COLUNA DESMETANIZADORA

*João Corrêa Lima Neto*

Projeto Final de Curso submetido ao Corpo Docente da Escola de Química, como parte dos requisitos necessários à obtenção do grau de Engenheiro Químico.

Aprovado por:

---

Andrea Valdman, M.Sc.

---

Rossana Odette Mattos Folly, D.Sc.

---

Lívia Pereira Lemos Costa, M.Sc.

Orientado por:

---

Maurício Bezerra de Souza Júnior, D.Sc.

Rio de Janeiro, RJ – Brasil  
Junho de 2011

Neto, João Corrêa Lima.

Sintonia de Controle Preditivo Pela Técnica de Enxame de Partículas: Aplicação a Uma Coluna Desmetanizadora / João Corrêa Lima Neto. Rio de Janeiro: UFRJ/EQ, 2011.

vii, 58 p.; il.

(Monografia) – Universidade Federal do Rio de Janeiro, Escola de Química, 2011.

Orientador: Maurício Bezerra de Souza Júnior.

1. Controle de Processos. 2. Otimização. 3. Gás Natural. 4. Monografia. (Graduação – UFRJ/EQ). 5. Maurício Bezerra de Souza Júnior, D.Sc.

*A meus pais, por todo o apoio e compreensão.*

## **AGRADECIMENTOS**

Agradeço aos meus pais, Lilian e Onofre, e aos meus irmãos, Aline e Marcos, por todo o suporte que me foi dado desde que saí de minha cidade natal, Manaus. São muitos anos longe de minha família, com uma compreensão magnífica por saberem de meus sonhos e desejos, me darem total apoio e me confortarem nos momentos difíceis e comemorarem as minhas vitórias junto comigo.

Um agradecimento mais do que especial ao meu amigo Pedro Santiago e a sua mãe, Cláudia Maria Santiago, que, mais do que um lugar pra ficar, me deram carinho e suporte, me fizeram sentir como se estivesse em casa e me trataram como se eu fizesse parte da família.

Agradeço aos amigos do quarteto fantástico (Alan Barbagelata El-Assad, Letícia Protector e Pedro Santiago) que, muito mais do que colegas de faculdade, tornaram-se irmãos e grandes amigos para a vida toda

Agradeço ao meu orientador, Maurício Bezerra de Souza Júnior, por todo o apoio dado, pelos conhecimentos compartilhados e conselhos que contribuíram para o desenvolvimento desse trabalho.

Agradeço também a todos os amigos que fiz no estágio, que contribuíram para o meu desenvolvimento profissional, me ensinaram muitas coisas e compartilharam diversos momentos felizes.

Resumo do Projeto Final apresentado à Escola de Química como parte dos requisitos necessários para obtenção do grau de Engenheiro Químico.

## **SINTONIA DE CONTROLE PREDITIVO PELA TÉCNICA DE ENXAME DE PARTÍCULAS: APLICAÇÃO A UMA COLUNA DESMETANIZADORA**

João Corrêa Lima Neto

Junho, 2011

Orientadores: Maurício Bezerra de Souza Júnior

Os controladores preditivos (MPC ou *model predictive controllers*) aparecem para a indústria em meados da década de 1970 e, desde então, só vem se desenvolvendo e ganhando mais aplicações. Podem ser citadas diversas vantagens de seu uso quando comparados com os controladores tradicionais PID, como incorporação de restrições ao modelo e controle de plantas multivariáveis, não quadradas, entre outros. Mesmo com o progresso alcançado nas últimas décadas, com estudos enfocando sua formulação matemática, técnicas para a sintonia não estão bem estabelecidas. Apesar de existirem diversos trabalhos, a maioria delas é baseada em tentativa-e-erro e exige conhecimento especializado da planta e de técnicas de controle. Dessa maneira, o presente trabalho propõe uma metodologia de sintonia de controladores preditivos baseada em otimização de função de desempenho, considerando como parâmetro de decisão todos os parâmetros típicos de sintonia MPC (horizontes e penalidades). Para tal, utilizou-se a ferramenta de projeto de controladores preditivos do MATLAB® e uma técnica meta-heurística de otimização por enxame de partículas. A abordagem foi aplicada para um modelo de uma coluna de desmetanização de um processo de produção de gás natural baseado na tecnologia de turbo-expansão. Os resultados obtidos demonstraram ser superiores àqueles obtidos por tentativa-e-erro. Adicionalmente, a sintonia proposta pode ser facilmente expandida a outras aplicações.

## Sumário

<b>1. INTRODUÇÃO</b> .....	1
<b>1.1. Objetivos</b> .....	3
<b>1.2. Estrutura</b> .....	3
<b>2. REVISÃO BIBLIOGRÁFICA</b> .....	5
<b>2.1. Controle Preditivo</b> .....	5
2.1.1 <i>Histórico</i> .....	5
2.2.2 <i>Características</i> .....	11
2.2.3 <i>Algoritmo</i> .....	13
2.2.4 <i>Sintonia de Controladores MPC</i> .....	16
<b>2.3 Otimização por Enxame de Partículas</b> .....	25
<b>3 METODOLOGIA</b> .....	28
<b>3.2 Exemplo de Processo</b> .....	28
<b>3.3 Sintonia</b> .....	35
<b>4 RESULTADOS</b> .....	42
<b>5 CONCLUSÕES E SUGESTÕES</b> .....	54
<b>REFERÊNCIAS BIBLIOGRÁFICAS</b> .....	56
<b>APÊNDICE A1 – Códigos Utilizados</b> .....	59

# 1. INTRODUÇÃO

Controle preditivo baseado em modelo (MPC – *Model Predictive Control*) é uma técnica importante de controle avançado para problemas multivariáveis. Trata-se de uma classe de algoritmos computacionais de controle que utilizam explicitamente um modelo de processo para prever o comportamento futuro da planta e para otimizar o seu comportamento (Qin & Badgwell, 2003).

O principal objetivo no projeto de um controlador preditivo é calcular a trajetória futura das variáveis manipuladas de modo que, a cada intervalo de controle, apenas o primeiro movimento seja mandado para o atuador, visando otimizar o comportamento futuro de uma unidade de processo. A otimização é feita dentro de uma janela limitada de tempo utilizando informações disponíveis no início desse intervalo (Wang, 2009).

Originalmente desenvolvidos para atender a necessidades especiais de controle em plantas de geração de energia e em refinarias, os controladores preditivos possuem agora uma extensa faixa de aplicação, que vão desde a indústria química, de processamento de alimentos e automotiva à indústria aeroespacial (Qin & Badgwell, 2003). Nos últimos anos, o MPC estabeleceu-se como uma técnica padrão de controle multivariável que realiza a operação econômica de grandes unidades de processo.

A grande aceitação do MPC como estratégia de controle avançado na indústria é devida, principalmente, à sua habilidade em lidar com restrições em uma unidade de processo (sejam elas físicas, de segurança ou de questões ambientais), tempo morto e sistemas multivariáveis em uma estrutura unificada de projeto.

Os objetivos globais de um controlador MPC foram resumidos por Qin e Badgwell (2003):

- Prevenir que as restrições nas variáveis de entrada e de saída sejam violadas;
- Conduzir algumas variáveis de saída aos seus pontos ótimos, enquanto mantém as outras dentro de um intervalo estabelecido;



- Prevenir ação excessiva das variáveis manipuladas;
- Controlar o máximo possível de variáveis de processo na ausência de um sensor ou atuador.

Devido à ampla faixa de aplicações, a estrutura dos controladores MPC, que incorpora o modelo de processo, pode variar drasticamente de aplicação para aplicação. De forma que há diversos trabalhos dedicados a sintonia MPC, mas para casos específicos, o que torna os métodos de sintonia não gerais e, portanto, com uma abordagem de tentativa e erro sendo utilizada, o que dispende tempo e pode resultar numa sintonia não ótima (Lee, Svrcek & Young, 2008).

A grande dificuldade no desenvolvimento de condições teóricas que garantam a estabilidade e o bom desempenho do controlador, especialmente na presença de distúrbios, incertezas do modelo e restrições, conduz a uma estratégia de sintonia dos parâmetros baseada na experiência e nos ajustes manuais dos parâmetros de sintonia. Muitas das vezes, a relação entre os valores dos parâmetros na sintonia e o significado físico dos mesmos não estão claros. Assim, apesar de muitas técnicas de sintonia de controladores MPC estarem disponíveis na literatura, nenhum método geral parece ter sido proposto, para conhecimento do autor deste texto.

Na formulação do problema de sintonia de parâmetros ótimos, a função objetivo, muitas das vezes, é uma função multidimensional e difícil de ser analisada. Quando se tem este tipo de problema para otimização, alguns dos candidatos para método de busca do ponto ótimo são baseados em gradiente e na hessiana. Todos eles, porém, dependem da existência e do conhecimento de limites e derivadas (Medeiros, 2005). Dessa maneira, métodos de otimização estocásticos aparecem como bons candidatos para a solução deste problema. De fato, esses métodos não são dependentes da estrutura da função objetivo e de suas restrições, podem escapar de mínimo local e não requerem o cálculo de derivadas (Magni *et al*, 2009). Isto motiva o emprego do método de otimização por enxame de partículas para a resolução do problema de sintonia de MPC.

No Brasil, existem 25 unidades de processamento de gás natural distribuídas por quatro estados brasileiros, totalizando uma capacidade instalada de cerca de 65 milhões de m<sup>3</sup>/dia, sendo que diversas unidades já vem recebendo ampliações para aumento de capacidade. Já no último ano, a oferta de gás natural no mercado cresceu 12% acima da média dos últimos três anos e as perspectivas para o mercado nacional de gás natural ainda são muito boas, tendo grande importância para este último fato, as recentes descobertas das reservas de pré-sal, que contém grandes quantidades de gás associado ao óleo (Cordeiro, 2011).

Entretanto, o processamento de gás natural é considerado um processo complexo devido a diversos fatores, como alta integração entre os sistemas, por se tratarem de processos criogênicos, e a necessidade de manter as especificações dos produtos, que geralmente são bem restritivas (Cordeiro, 2011). Assim, devem ser despendidos esforços para que uma boa estratégia de controle seja aplicada, de modo a manter o produto dentro dos requisitos de mercado.

### **1.1. Objetivos**

O objetivo deste trabalho foi desenvolver uma metodologia geral para a sintonia dos parâmetros de controladores preditivos baseados em modelo. Propõe-se um método automático integrado ao *toolbox* de controle preditivo do MATLAB<sup>®</sup> para a sintonia, no qual os parâmetros de ajuste são determinados por otimização por enxame de partículas (PSO – *Particle Swarm Optimization*).

Para testar a metodologia, utilizou-se o modelo de processo linearizado identificado por Cordeiro (2011) para uma torre desmetanizadora de uma unidade de produção de gás natural baseada no processo de turbo-expansão.

### **1.2. Estrutura**

O texto está dividido conforme a ordem lógica para a compreensão da metodologia proposta, desde os fundamentos necessários para o entendimento de controladores MPC e da Otimização por Enxame de partículas, que são os pilares desse trabalho, até o exemplo de aplicação para validação da metodologia.

No capítulo 2, é apresentada uma revisão bibliográfica. Nesta parte, é apresentado um histórico do controle preditivo baseado em modelo, mostrando a sua evolução. Suas principais características são citadas e, ainda, o algoritmo de controle utilizado é apresentado. É feita também uma introdução ao PSO, mostrando como se dá a sua formulação matemática. Realiza-se ainda uma breve revisão sobre os métodos de sintonia para MPC disponíveis. No capítulo 3, a metodologia proposta é apresentada. Nesse ponto, é feita uma descrição do processo e o algoritmo de sintonia é apresentado. No capítulo 4, são apresentados os resultados das simulações baseadas em PSO-MPC. Por fim, o capítulo 5 apresenta as conclusões obtidas e inclui propostas para o desenvolvimento de estudos futuros.

## 2. REVISÃO BIBLIOGRÁFICA

### 2.1. Controle Preditivo

#### 2.1.1 Histórico

A Figura 1 mostra a árvore evolutiva dos algoritmos industriais mais significativos para o controle preditivo, mostrando a relação entre eles de maneira concisa. Pode-se encontrar uma descrição completa desses algoritmos no trabalho de Qin e Badgwell (2003), onde é feito um estudo das tecnologias industriais de MPC.

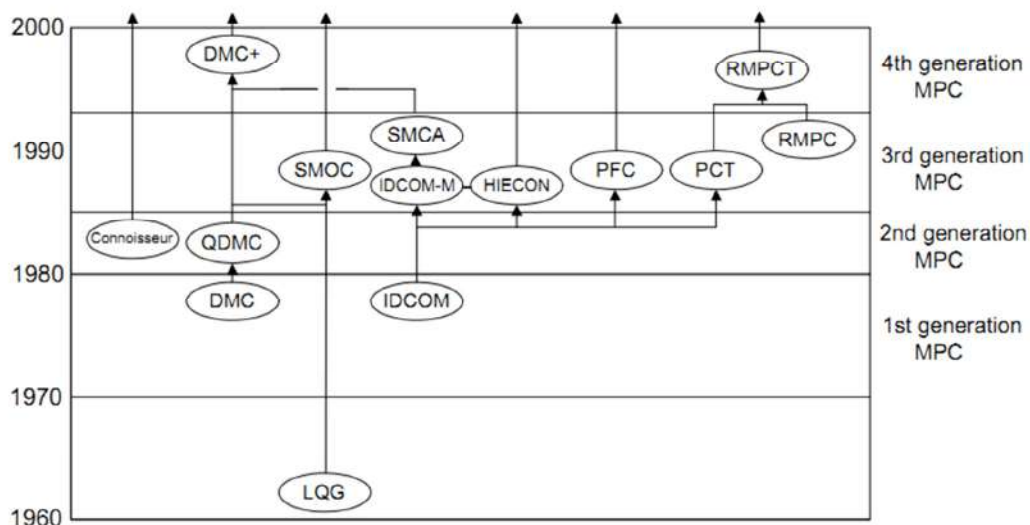


Figura 1 - Algoritmos MPC Industriais

Fonte: Qin & Badgwell (2003)

O problema proposto por Kalman (1960), com uma solução conhecida por controlador gaussiano quadrático linear, é denominado LQG (*Linear Quadratic Gaussian Controller*). Apesar de rapidamente ter se tornado uma abordagem padrão para problemas de controle em diversas aplicações, ele teve pouco impacto nas tecnologias de desenvolvimento de controladores na indústria. Os principais motivos para este fato são citados por Richalet *et al* (1976) e Garcia, Prett e Morari (1989):

- Não lida com restrições nas variáveis de processo;
- Trabalha apenas com modelos lineares;
- Não lida bem com incertezas no modelo, de forma que só apresentou efeito estabilizante em modelos perfeitos;

- Critério único de desempenho;
- Razões culturais: técnicos instrumentistas e engenheiros de controle ou não foram expostos ao LQG ou o trataram como impraticável.

Esse cenário levou ao desenvolvimento, no âmbito industrial, de metodologias de controle baseadas em modelo mais gerais. Assim, em 1978, Richalet *et al* descreveram em um artigo aplicações de sucesso do Controle Heurístico Preditivo Baseado em Modelo (MPHC – *Model Predictive Heuristic Control*), que ficou conhecido como IDCOM (*Identification and Comand*), e que tem uma abordagem com as seguintes características:

- Modelo de resposta ao impulso para a planta, com variáveis de entradas ou internas lineares;
- Função objetivo quadrática em um horizonte de predição finito;
- Comportamento futuro das variáveis de saída especificado por uma trajetória de referência;
- Restrições para as variáveis de entrada e de saída incluídas na sua formulação;
- Valores ótimos de saída calculados usando um algoritmo heurístico iterativo.

Logo em seguida, em 1979, Cutler e Ramaker, então engenheiros da Shell, apresentaram detalhes de um algoritmo de controle multivariável sem restrições, o qual foi denominado DMC (*Dynamic Matrix Control*), e mostraram resultados da aplicação no controle de temperatura de um forno para demonstrar a melhora na qualidade do controle. E, em uma parceria, Cutler e Ramaker (1980) descreveram uma aplicação de DMC em uma unidade de FCC, na qual o algoritmo foi modificado para lidar com não-linearidades e restrições.

Características chave do DMC são listadas abaixo:

- Modelo linear de resposta ao degrau para a planta;
- Função objetivo quadrática em um horizonte de predição finito;
- Comportamento futuro das variáveis de saída especificado pela tentativa de seguir o setpoint o mais próximo possível;

- Valores ótimos para as variáveis de entrada calculados como a solução de um problema de mínimos quadrados.

Conforme apresentado, o IDCOM e o DMC representaram a primeira geração da tecnologia MPC, tendo um grande impacto na indústria de controladores de processo e servindo para definir o paradigma do MPC na indústria. Contudo, apesar de proverem um controle excelente para processos multivariáveis sem restrições, estas ainda eram algo um tanto *ad hoc*. Assim, novamente engenheiros da Shell propuseram um algoritmo DMC formulado como um problema de programação quadrática, em que restrições nas variáveis de saída e de entrada apareciam explicitamente. A primeira publicação do QDMC (*quadratic dynamic matrix control*) foi feita por Cutler, Morshedi e Haydel (1983) e, em 1986, Garcia e Morshedi publicaram um descrição mais compreensiva do algoritmo, onde apresentaram resultados da aplicação em um forno de pirólise.

Entre as principais características do QDMC, estão:

- Modelo de resposta ao degrau linear para a planta;
- Função objetivo quadrática em um horizonte finito de predição;
- Comportamento futuro das variáveis de saída especificado pela tentativa de seguir o setpoint o mais próximo possível sujeito a um termo de restrições nas variações da ação do controlador entre os intervalos de controle;
- Valores ótimos para as variáveis de entrada calculados como a solução de uma programação quadrática.

Assim, o QDMC pode ser tratado como o representante da segunda geração da tecnologia MPC, composto de algoritmos os quais provêm uma forma sistemática de implementar restrições nas variáveis de entrada e de saída, que só se tornou possível formulando o MPC como um problema de programação quadrática.

Com a grande aceitação de controladores MPC, as dificuldades e a complexidades dos problemas impostos ao controle do processo foram aumentando. O algoritmo QDMC, apesar de possibilitar a implementação de

restrições nas variáveis de entrada e de saída, não podia, por exemplo, lidar de maneira clara com as soluções inviáveis geradas pela programação quadrática. Outro problema que passou a ser considerado foi a variação dos graus de liberdade do processo: quando há uma falha de sinal, por exemplo, ou quando há uma falha no atuador de uma válvula. Nessas situações, não é possível atingir todos os objetivos de controle e, portanto, as restrições tem que se tornar flexíveis para tolerar possíveis violações. Outra dificuldade, ainda, é traduzir as necessidades de controle em pesos na função objetivo, principalmente para plantas maiores.

Essas e outras questões levaram engenheiros da Adersa, Setpoint, Inc., e Shell, a desenvolverem novas versões dos algoritmos MPC. Enquanto a Setpoint desenvolveu o denominado IDCOM-M, descrito por Grosdidier, Froisy e Hammann (1988) e, posteriormente por Froisy e Matski (1990) de forma mais detalhada, a Adersa desenvolveu o quase idêntico HIECON (Hierarchical Constraint Control).

As principais características do IDCOM-M são:

- Modelo linear de resposta ao impulso para a planta;
- Supervisão da controlabilidade de subconjuntos mal condicionados da planta;
- Formulação multi-objetivo: função objetivo quadrática para as variáveis de saída seguida por outra função quadrática para as variáveis de entrada;
- Controle de um conjunto de valores futuros para as variáveis de saída, chamados pontos de coincidência, escolhidos da trajetória de referência;
- Um único movimento é calculado para cada variável de entrada;
- Restrições podem ser rígidas ou brandas, com restrições rígidas tratadas como prioridade.

Na continuação do desenvolvimento do IDCOM-M, engenheiros da *Setpoint* incorporaram outros produtos de simulação, identificação, controle etc. e

apresentaram uma solução integrada, denominada SMCA, para arquiteturas de controle multivariável.

E, no final de 1980, engenheiros da Shell, na França, desenvolveram o SMOC (Shell Multivariable Optimizing Controller), descrito por Yousfi e Tounier (1991) e, posteriormente por Marquis e Broustail (1998), o qual foi descrito como a ponte entre modelos de espaço de estado e algoritmos MPC. Eles procuraram combinar as características de tratamento de restrições do MPC com a rica estrutura para controle feedback oferecido pelos métodos de espaço de estado. Suas características são:

- Utilização de um modelo de espaço de estado, de modo de que toda a dinâmica linear pode ser representada (estável, instável e integradora);
- Um modelo explícito para os distúrbios descreve o efeito de distúrbios não medidos; um distúrbio constante na saída é um caso simples;
- Um filtro de Kalman é usado para estimar o estado da planta e distúrbios não medidos das medidas das variáveis de saída;
- Uma distinção é feita entre variáveis controladas na função objetivo e variáveis de realimentação que são usadas para estimação do estado;
- Restrições nas variáveis de entrada e de saída são impostas por programação quadrática.

Assim, os algoritmos IDCOM-M, HIECON, SMCA e SMOC representam a terceira geração da tecnologia MPC onde ainda podem ser citados o algoritmo PCT, da Profimatics, e o RMPC, da Honeywell. Essa geração se distingue pelos diferentes níveis de restrições (rígidas, brandas e ranqueadas), mecanismos para sair de soluções irrealizáveis, mudanças online na estrutura de controle e por permitir uma grande faixa para as dinâmicas de processo (estáveis, integradores e instáveis).

Quando se refere à quarta geração de controladores MPC, o cenário anterior se perde, pois muitas empresas se fundiram. Em 1995, A Honeywell comprou a Profimatics, Inc., formando a Honeywell Hi-spec Solutions, surgindo



então o RMPCT, oriundo do RMPC e do PCT. Em 1996, a Aspen comprou a Setpoint, a DMC Corporation e a Treiber Controls, surgindo, então, o DMC-plus, que, junto com o RMPCT, formam a quarta geração. Suas principais características são:

- Interface gráfica com o usuário baseada em Windows;
- Múltiplos níveis de otimização para a priorização dos objetivos de controle;
- Maior flexibilidade na otimização em estado estacionário, incluindo objetivos econômicos e programação quadrática;
- Consideração direta de incertezas no modelo;
- Melhora na tecnologia de identificação baseada no método de predição do erro e métodos de subespaço.

Em 2003, em um artigo publicado por Qin e Bagdwell, foram apresentadas as principais companhias que representavam o estado da arte em MPC na época, conforme mostrado abaixo:

**Tabela 1 - Companhias e Produtos em tecnologias MPC lineares**

<b>Companhia</b>	<b>Nome de Produto</b>	<b>Descrição</b>
Adersa	HIECON	Controle Hierárquico de restrições
	PFC	Controle Preditivo Funcional
	GLIDE	Pacote de Identificação
Aspen Tech	DMC-plus	Pacote de Controle por Matriz Dinâmica
	DMC-plus model	Pacote de Identificação
Honeywell Hi-Spec	RMPCT	Tecnologia de Controle Preditivo Robusto
Shell Global Solutions	SMOC-II	Controle Shell por Otimização Multivariável
Invensys	Connoisseur	Pacote de Controle e Identificação

Fonte: Qin & Bagdwell (2003)

Já em 2010, em um artigo publicado por Kano e Ogawa, um status mais atual do estado da arte no Japão é apresentado. Eles verificaram que existem

então 329 aplicações, onde se tem um cenário dominado pela DMC-plus e pela tecnologia RMPCT. É citado também que metade das aplicações se encontra em processos com reações e destilação. Abaixo segue um resumo das aplicações e das principais tecnologias em MPC utilizadas no Japão:

**Tabela 2 - Principais aplicações e tecnologias em MPC no Japão (2010)**

<i>In house</i> vs Fornecedores	
Desenvolvimento <i>In house</i>	4%
Pelo Fornecedor	96%
Processos Alvo	
Destilação	40%
Reação	30%
Outros	30%
Produtos	
DMCplus/DMC	46%
RMPCT	34%
Connoisseur	5%
IDCOM	4%
SMOC	3%
MECMACS	3%
Outros	5%

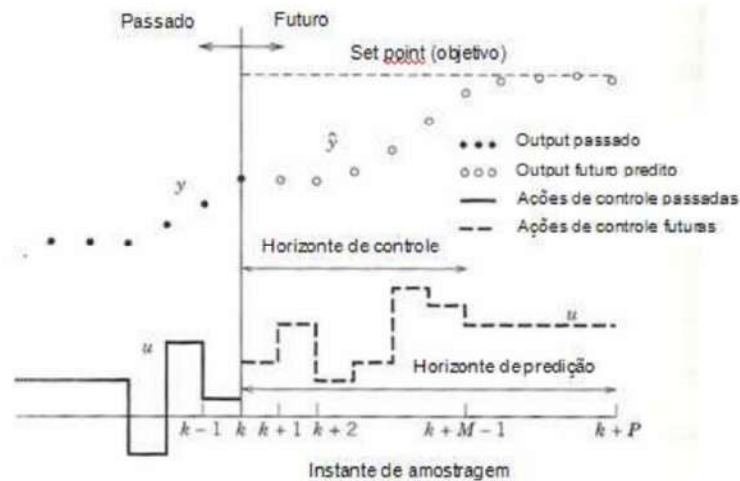
Fonte: Kano & Ogawa (2010)

Verifica-se então que a quarta geração MPC, conforme definida por Qin & Bagdwell (2003), persiste até hoje.

### 2.2.2 Características

Os algoritmos de controle MPC utilizam explicitamente um modelo de processo para prever o comportamento futuro da unidade dentro de uma janela de tempo. Em cada intervalo de controle, o algoritmo tenta otimizar o comportamento da planta calculando uma sequência de ações ótimas de controle em um horizonte determinado. Entretanto, apenas o primeiro movimento de controle é executado e, então, todo o cálculo é repetido para os intervalos de tempo subsequentes.

Os cálculos são baseados em predições de valores futuros das variáveis de controle e dos valores de seus *setpoints*. E têm como objetivo determinar a sequência de movimentos de controle ótimos de modo que as variáveis de controle atinjam os valores desejados de maneira ótima. A Figura 2 ilustra bem os conceitos citados:



**Figura 2 - Conceitos Básicos de Controle Preditivo**

Fonte: Seborg *et al* (2011)

De acordo com Wang (2009), os princípios que permeiam a teoria MPC são conforme se segue:

1. Janela de horizonte móvel (*Moving Horizon Window*): janela de tempo, de tamanho constante, onde a predição é feita e que, a cada intervalo de tempo, translada para a direita sem alteração de tamanho;
2. Horizonte de predição (*Prediction Horizon*): determina o tamanho da janela de tempo na qual as predições são feitas, e é da mesma magnitude da janela de horizonte móvel;
3. Controle por horizonte descendente (*Receding Horizon Control*): Apesar de a sequência ótima para as variáveis de controle ser calculado, apenas o primeiro movimento é mandado para o elemento final de controle;
4. Informações atuais da planta são necessárias para que a predição possa ser feita;
5. Um modelo que descreva o processo é de extrema importância para o controle preditivo;
6. Para que as melhores decisões possam ser tomadas, um critério que reflita o objetivo é necessário. Assim, as ações de controle podem ser tomadas pela solução desse problema.

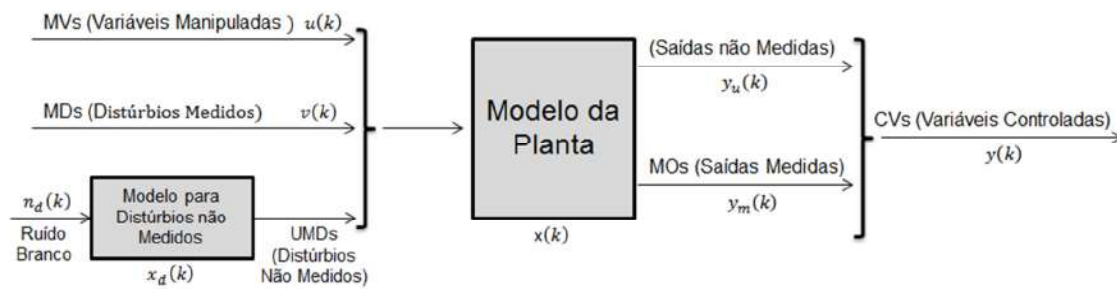
Uma vantagem importante da abordagem de horizonte descendente é a de a informação mais atual da planta ser sempre utilizada, pois não se espera

todo o horizonte de controle para realizar novos cálculos para as ações. Assim, evita-se que as previsões e as ações de controle sejam feitas baseadas em informações obsoletas do processo e, portanto, que sejam afetadas por distúrbios não medidos.

### 2.2.3 Algoritmo

Atualmente, quase todas as formulações MPC encontradas na literatura fazem uma abordagem de espaço de estado. Neste trabalho, todo o desenvolvimento foi feito em *MATLAB*<sup>®</sup>, que disponibiliza um toolbox para projeto de controladores preditivos, e que é baseado em espaço de estado. Portanto, será essa a abordagem aqui feita.

O modelo linear usado para a previsão e otimização pode ser visto a seguir:



**Figura 3 – Modelo Usado para Otimização**

Fonte: Bemporand *et al* (2010)

O modelo para a planta, invariante no tempo, é conforme se segue abaixo:

$$x(k + 1) = Ax(k) + B_u u(k) + B_v v(k) + B_d d(k) \quad (1)$$

$$y_m(k) = C_m x(k) + D_{vm} v(k) + D_{dm} d(k) \quad (2)$$

$$y_u(k) = C_u x(k) + D_{vu} v(k) + D_{du} d(k) \quad (3)$$

Onde:

- $x(k)$  é um vetor de dimensão  $n_x$  das variáveis de estado da planta;
- $u(k)$  é um vetor de dimensão  $n_u$  das variáveis manipuladas;
- $v(k)$  é um vetor de dimensão  $n_v$  dos distúrbios medidos que afetam o processo;

- $y_m(k)$  e  $y_u(k)$  são vetores das saídas medidas e das não medidas, respectivamente.

O distúrbio não medido,  $d(k)$ , é modelado como a saída do sistema linear invariante no tempo representado pelas Eq. (4) e Eq. (5), onde  $n_d(k)$  é um ruído Gaussiano de média zero e com matriz das covariâncias unitária:

$$x_d(k+1) = \bar{A}x_d(k) + \bar{B}n_d(k) \quad (4)$$

$$d(k) = \bar{C}x_d(k) + \bar{D}n_d(k) \quad (5)$$

Então, uma vez que exista um modelo de predição, conforme mostrado acima, cai-se no problema de otimização, onde os movimentos ótimos são calculados de forma a atingir o setpoint desejado para as variáveis controladas com o menor custo possível.

Com os valores de  $x(k)$  e  $x_d(k)$  disponíveis no instante de tempo  $k$ , a ação do controlador é obtida resolvendo-se o seguinte problema de otimização:

$$\begin{aligned} \min_{\Delta u(k|k), \dots, \Delta u(m-1+k|k), \varepsilon} & \left\{ \sum_{i=0}^{p-1} \left[ \sum_{j=1}^{n_y} |w_{i+1,j}^y (y_j(k+i+1|k) - r_j(k+i+1))|^2 \right. \right. \\ & \left. \left. + \sum_{j=1}^{n_u} |w_{i,j}^u \Delta u(k+i|k)|^2 + \sum_{j=1}^{n_u} |w_{i,j}^u (u_j(k+i|k) - u_{target}(k+i))|^2 \right] + \rho_\varepsilon \varepsilon^2 \right\} \end{aligned} \quad (6)$$

Onde o “( )<sub>j</sub>” denota a componente  $j$  do vetor;  $(k+i|k)$  é a predição para o tempo “ $k+i$ ” baseada nas informações disponíveis no instante  $k$ ;  $r(k)$  é a referência. A função objetivo está sujeita as seguintes restrições:

$$u_{j \min}(i) - \varepsilon V_{j \min}^u(i) \leq u_j(k+i|k) \leq u_{j \max}(i) + \varepsilon V_{j \max}^u(i) \quad (7)$$

$$\Delta u_{j \min}(i) - \varepsilon V_{j \min}^{\Delta u}(i) \leq \Delta u_j(k+i|k) \leq \Delta u_{j \max}(i) + \varepsilon V_{j \max}^{\Delta u}(i) \quad (8)$$

$$y_{j \min}(i) - \varepsilon V_{j \min}^y(i) \leq y_j(k+i+1|k) \leq y_{j \max}(i) + \varepsilon V_{j \max}^y(i) \quad (9)$$

$$\Delta u(k+h|k) = 0 \quad (10)$$

$$\varepsilon \geq 0 \quad (11)$$

Onde  $i = 1, \dots, p-1$  e  $h = m, \dots, p-1$

A ação de controle que é mandada para a planta é  $u(k) = u(k - 1) + \Delta u\{k|k\}$ , onde  $\Delta u\{k|k\}$  é o primeiro elemento da sequência ótima calculada pela função objetivo. Tem-se também que, se uma trajetória de referência não for definida, ela torna-se constante e igual ao setpoint para aquela variável de saída, ou seja,  $r(k + i + 1) = r(k)$ .

Tem-se ainda que:

- $w_{i,j}^y$ ,  $w_{i,j}^{\Delta u}$  e  $w_{i,j}^u$  são pesos, valores positivos, para as variáveis correspondentes. Quanto menor  $w$ , menos importante é o comportamento da variável no desempenho global do controlador;
- $u_{j \min}(i)$ ,  $\Delta u_{j \min}(i)$ ,  $y_{j \min}(i)$ ,  $u_{j \max}(i)$ ,  $\Delta u_{j \max}(i)$  e  $y_{j \max}(i)$  são as restrições de mínimo e de máximo para as variáveis correspondentes. As restrições em  $\Delta u_j$ ,  $u_j$  e  $y_j$  podem ser relaxadas pela introdução da variável de relaxamento,  $\varepsilon \geq 0$ . O peso dessa variável,  $\rho_\varepsilon$ , penaliza a violação da restrição. Quanto maior  $\rho_\varepsilon$ , mais a violação da restrição é penalizada;
- Os vetores  $V_{\min}^u$ ,  $V_{\max}^u$ ,  $V_{\min}^{\Delta u}$ ,  $V_{\max}^{\Delta u}$ ,  $V_{\min}^y$  e  $V_{\max}^y$ , denominados vetores ECR (*Equal Concern Relaxation*), são positivos e quantificam o “relaxamento” permitido para determinada variável. Quanto maior  $V$ , mais a restrição se torna violável.  $V = 0$  significa que a restrição é rígida que não pode ser violada. Por *default* da ferramenta utilizada, todas as restrições nas entradas são rígidas ( $V_{\min}^u = V_{\max}^u = V_{\min}^{\Delta u} = V_{\max}^{\Delta u}$ ) e todas as restrições nas variáveis de saída são brandas ( $V_{\min}^y = V_{\max}^y = 1$ );
- Por *default*  $\rho_\varepsilon = 10^5 \max\{w_{i,j}^y, w_{i,j}^{\Delta u}, w_{i,j}^u\}$ ;
- $u_{target}$  é o setpoint para o vetor das variáveis de entrada;
- Conforme mencionado anteriormente, apenas  $\Delta u\{k|k\}$  é usada para o cálculo de  $u(k)$ , sendo todo o resto descartado e a otimização resolvida novamente no instante  $k + 1$  com base em  $y_m(k + 1)$ .

Vale ressaltar que o *MATLAB*<sup>®</sup> usa diferentes algoritmos dependendo ou não da presença de restrições. Se não há restrições, o controlador usa uma rápida solução analítica para o cálculo das ações ótimas de controle em cada

instante de tempo. Caso haja restrições, uma programação quadrática é usada para a solução.

O *MATLAB*<sup>®</sup> dispõe de uma função objetivo alternativa para o cálculo das ações ótimas de controle:

$$J(\Delta u, \varepsilon) = \sum_{i=0}^{p-1} [y(k+i+1|k) - r(k+i+1)]^T Q [y(k+i+1|k) - r(k+i+1)] \quad (12)$$

$$+ \Delta u(k+i|k)^T R_{\Delta u} \Delta u(k+i|k)$$

$$+ [u(k+i|k) - u_{target}(k+i)]^T R_u [u(k+i|k) - u_{target}(k+i)] + \rho_\varepsilon \varepsilon^2$$

Onde  $Q$  é uma matriz  $n_y \times n_y$  e  $R_{\Delta u}$  e  $R_u$  são matrizes com dimensão  $n_u \times n_u$ . Mas que se tornam iguais sob as seguintes condições:

- $w_{i,j}^y$ ,  $w_{i,j}^{\Delta u}$  e  $w_{i,j}^u$  constantes em todo o horizonte de predição;
- Quando as matrizes  $Q$ ,  $R_{\Delta u}$  e  $R_u$  são diagonais com os pesos  $w_{i,j}^y$ ,  $w_{i,j}^{\Delta u}$  e  $w_{i,j}^u$  elevados a segunda potência como elementos da diagonal principal.

#### 2.2.4 Sintonia de Controladores MPC

O projeto de controladores MPC envolve muitos parâmetros, conforme foi mostrado na formulação acima. Todos eles afetam fortemente o desempenho e determinam o quão robusto vai ser o controlador. Assim, podemos classificar como principais parâmetros de sintonia as penalidades das variáveis de entrada ( $w_{i,j}^{\Delta u}$  e  $w_{i,j}^u$ ) e das variáveis de saída ( $w_{i,j}^y$ ), assim como os horizontes de controle,  $M$ , e de predição,  $P$ .

Vários pesquisadores abordaram a sintonia de controladores MPC com restrições e sem restrições para a formulação DMC, provavelmente a mais bem conhecida e popular tecnologia industrial em MPC (Wojsznis et al, 2003).

Em 1998, Shridhar e Cooper desenvolveram uma estratégia de sintonia para o algoritmo de controle DMC sem restrições, de fácil uso e de sintonia confiável e que ainda, segundo os autores, faz com que o controlador tenha um desempenho ótimo e atue com o mínimo esforço. A proposta se baseia na

dualidade de propósito do termo de supressão das ações de controle,  $\lambda_i^2$ , que tem, como papel principal, evitar ações de controle agressivas quando  $M > 1$ . Além disso, ele atua também melhorando o condicionamento da matriz do sistema,  $A^T \Gamma^T \Gamma A$ , tornando-a mais positiva definida. Tendo como premissa o fato de os dois propósitos estarem relacionados, os autores elaboram o procedimento de sintonia mostrado na Tabela 3.

**Tabela 3 - Sintonia de Controladores DMC**

<b>Estratégia de Sintonia para o Algoritmo DMC</b>
<p>1. Aproximar a dinâmica do processo para todos os pares de variáveis de entrada-saída para sistema de primeira ordem com tempo morto:</p> $\frac{y_j(s)}{u_i(s)} = \frac{K_{ij} e^{-\theta_{ij}s}}{\tau_{ij}s + 1}, (i = 1, 2, \dots, S; j = 1, 2, \dots, R)$ <p>2. Selecionar o tempo de amostragem o mais próximo possível de:</p> $T = \text{Min} \left[ \text{Max} \left( 0, 1\tau_{ij}, 0, 5\theta_{ij}; (i = 1, 2, \dots, S; j = 1, 2, \dots, R) \right) \right]$ <p>3. Calcular o horizonte de predição e o horizonte do modelo (arredondar para o inteiro mais próximo):</p> $P = N = \text{Max} \left( \frac{5\tau_{ij}}{T} + k_{ij} \right), \text{ onde } k_{ij} = \left( \frac{\theta_{ij}}{T} + 1 \right), (i = 1, 2, \dots, S; j = 1, 2, \dots, R)$ <p>4. Selecionar o horizonte de controle, M, como um inteiro no intervalo [1,6]</p> <p>5. Selecionar os pesos das variáveis de controle para, <math>\lambda_j^2</math>, para equalizar a magnitude das medidas</p> <p>6. Calcular <math>\lambda_i^2</math>:</p> $\lambda_i^2 = \frac{M}{500} \sum_{j=1}^R \left[ \gamma_j^2 K_{ij}^2 \left( P - k_{ij} - \frac{3\tau_{ij}}{2T} + 2 - \frac{(M-1)}{2} \right) \right], (i = 1, 2, \dots, S; j = 1, 2, \dots, R)$ <p>7. Implementar o algoritmo DMC usando a matriz de resposta ao degrau, o processo real e os parâmetros calculados nos passos anteriores.</p>

Fonte: Shridhar & Cooper (1998)

Entretanto, esta metodologia não inclui restrições e requer a representação do processo por um sistema de primeira ordem com tempo morto, o que não funciona muito bem para sistemas de ordem superior e/ou sistemas não estáveis (Al-Ghazzawi *et al*, 2001).

Já na última década, em 2001, Al-Ghazzawi *et al*, propuseram uma metodologia para sintonia *online* de controladores preditivos, que tem como base uma aproximação linear entre os valores preditos em malha fechada para as variáveis de saída e os parâmetros de sintonia MPC:

$$\hat{y}_j(k+i) = y_j(k+i) + \nabla x_k y_j^T(k+i) \Delta x_k \quad (13)$$



Onde  $\nabla_{x_k y_j^T}$  é o gradiente das respostas preditas em malha fechada conforme representados pelas Eq. (14) e Eq. (15) e  $\hat{y}_j(k+i)$  é o valor esperado para a saída para o novo valor de  $x$ .

Este método está restrito a modelos de resposta ao degrau. Os autores propõem uma estratégia adaptativa, aplicável tanto a sistemas com restrições quanto a sistema sem restrições, e que é obtida pelas equações de sensibilidade aos parâmetros de sintonia em malha fechada:

$$\frac{\partial Y(k+1)}{\partial \lambda_1} = \phi_H \frac{\partial Y(k)}{\partial \lambda_1} + 2\lambda_1 (G_{\lambda_1} g + T_{\lambda_1} \hat{b}) \quad (14)$$

$$\frac{\partial Y(k+1)}{\partial \gamma_1} = \phi_H \frac{\partial Y(k)}{\partial \gamma_1} + 2\gamma_1 (\theta_{H_1} (Hg + T\hat{b}) - \vartheta_{H_1} e_P(k+1)) \quad (15)$$

Onde :

$$\theta_{H_1} = Se^T G^{-1} H S_p^{mT} I_{\gamma_1} S_p^m G^{-1} \quad (16)$$

$$\vartheta_{H_1} = Se^T G^{-1} H S_p^{mT} I_{\gamma_1} \quad (17)$$

Apenas os pesos nas variáveis de saída,  $\Gamma$ , e nas variáveis de entrada,  $\Lambda$ , são considerados para a sintonia. Os horizontes de controle,  $M$ , e de predição,  $P$ , têm seus valores determinados por diretrizes já conhecidas. Os autores desenvolvem então as expressões analíticas, mostradas acima, para avaliar a sensibilidade com relação às penalidades nas variáveis de entrada e de saída. Uma descrição completa do algoritmo, assim como a nomenclatura para as variáveis e parâmetros, pode ser vista no trabalho publicado pelos autores.

Em 2003, Wojsznis *et al* sugeriram um método prático para a implantação e sintonia de controladores MPC que, segundo os autores, caracteriza-se pela forma simples e intuitiva com a qual os parâmetros do projeto do controlador são calculados. Os seguintes passos devem ser seguidos:

- 1) Projeto, revisão e validação do modelo de processo:

Para o modelo interno do controlador, utiliza-se o modelo de resposta ao degrau. A sua identificação é feita usando duas técnicas de modelagem: FIR (*Finite Impulse Response*) e ARX (*Autoregressive with External Inputs*):

$$\Delta y_k = \sum_{i=1}^P h_i \Delta u_{k-i} \quad (18)$$

$$y_k = \sum_{i=1}^V a_i y_{k-1} + \sum_{i=1}^A b_i u_{k-d-i} \quad (19)$$

Onde A é a ordem autoregressiva, V é a ordem média de movimento e d é o tempo morto.

Isso se deve ao fato de o número de parâmetros para identificação do modelo de resposta ao degrau no horizonte de predição ser muito grande, de modo que se utiliza a modelagem FIR, com menos parâmetros, para estimar a parte inicial da resposta ao degrau e definir o tempo morto. Este último é então utilizado na modelagem ARX, que contém consideravelmente menos parâmetros que a modelagem FIR.

Após a identificação, segue a validação e revisão do modelo, através de simulações para cálculo do erro em relação ao processo real, alteração/verificação da resposta ao degrau e validação estatística do modelo.

## 2) Geração dos parâmetros do controlador:

É sabido que o desempenho do controlador é modificado pelo ajuste dos seguintes parâmetros: P, M,  $\Gamma^u$  e  $\Gamma^y$ . Entretanto, os autores afirmam que usar P e M como parâmetros de sintonia não é usual, de modo que se especifica o horizonte de predição grande o suficiente para que pequenas alterações no seu valor não afetem o desempenho do controlador. Da mesma maneira, considera-se que assumir  $\Gamma^y = 1$  faz parte de boas práticas. Assim, resta apenas  $\Gamma^u$  como parâmetro para sintonia do controlador. Os autores chegam então ao seguinte resultado experimental:

$$PM_i = 3 \left( 1 + \frac{6DT_i}{p} + \frac{3G_iDT_i}{p} \right) \quad (20)$$

Onde  $DT_i$  é o tempo morto para um par “i-j” de variável manipulada-controlada,  $G_i$  é o ganho para o mesmo par e p é o horizonte de

predição. Aplicando-se então a Eq. (20) acima na implementação do MPC,  $\Gamma^u$  é calculado automaticamente, sem necessitar de intervenção.

3) Simulação *off-line*:

A estratégia de controle deve ser simulada antes de ser utilizada para reduzir o risco de falhas, de erros de configuração e evitar testes diretamente na planta.

4) Sintonia *online*:

Fazer ajustes *online* para o bom desempenho do controlador como, por exemplo, incluir filtro de *setpoint* ou a trajetória de referência.

Ainda em 2003, Trierweiler e Farina propõem um método de sintonia para controladores DMC baseada em RPN (*Robust Performance Number*), que é uma medida de controlabilidade do sistema. O RPN indica o quão difícil é para o sistema alcançar, de forma robusta, o desempenho desejado.

Define-se então o número de desempenho robusto (RPN,  $\Gamma$ ) de uma planta multivariável com função de transferência  $G(s)$  da seguinte maneira:

$$RPN \triangleq \Gamma_{sup}(G, T, \omega) = \sup_{(\omega \in R)} \{\Gamma(G, T)\} \quad (21)$$

$$\Gamma(G, T) \triangleq \sqrt{\bar{\sigma}([I - T(j\omega)]T(j\omega)) \left( \gamma^*(G(j\omega)) + \frac{1}{\gamma^*(G(j\omega))} \right)} \quad (22)$$

Onde  $\gamma^*(G(j\omega))$  é o número de condição minimizado de  $G(j\omega)$  e  $\bar{\sigma}([I - T]T)$  é o valor singular máximo da matriz  $[I - T]T$ , sendo T a função complementar de sensibilidade desejada para a saída, que é determinada a partir de  $G(s)$ .

Trierweiler e Farina (2003) afirmam que antes de se determinar a controlabilidade do sistema, é de suma importância que a matriz das funções de transferência esteja escalonada. Dessa maneira, propõem o procedimento, mostrado na Tabela 4, baseado em RPN:

**Tabela 4 - Procedimento de Escalonamento RPN**

1. Determinar a frequência  $\omega_{sup}$  na qual  $\Gamma(G, T)$  tem valor máximo;
2. Calcular as matrizes de escalonamento  $L_s$  e  $R_s$  de modo que  $\gamma(L_s G(j\omega_{sup}) R_s)$  alcance o seu valor mínimo  $\gamma^*(G(j\omega_{sup}))$ ;
3. Escalonar o sistema:  $G_S(s) = L_s G(s) R_s$

Fonte: Trierweiler & Farina (2003)

O procedimento de sintonia proposto pelos autores encontra-se resumido na Tabela 5:

**Tabela 5 - Procedimento de sintonia MPC baseado em RPN para sistemas MIMO**

1. Especificar o desempenho desejado  $T_d$ ;
2. Fatoração de  $G(s)$  para determinar um desempenho alcançável;
3. Escalonamento do sistema de acordo com o procedimento da Tabela 4;
4. Determinação do tempo de amostragem:  $0,06\tau_r < t_s < 0,4\tau_r$ ;
5. Determinação do horizonte de predição (duas possibilidades):

$$(A) \mathbf{P} \approx \frac{t_{80\%}}{t_s}, \mathbf{P}_0 \approx \mathbf{0}, \text{ e } \mathbf{r}_s = \mathbf{L}_s \mathbf{T}(s) \mathbf{r}_d(s)$$

$$(B) \mathbf{P} \approx \frac{t_{90\%}}{t_s}, \mathbf{P}_0 \approx \frac{t_{10\%}}{t_s}, \text{ e } \mathbf{r}_s = \mathbf{L}_s \mathbf{r}_d(s)$$

Onde  $\mathbf{r}_d$  é o sinal de referência;

6. Horizonte de Controle:  $\mathbf{M} \approx \mathbf{P}/4$ ;
7. Calcular a ação de controle  $\mathbf{u}_s$ :

$$J_s = \sum_{j=P_0}^P \|\hat{\mathbf{y}}_s(t+j|t) - \mathbf{r}_s(t+j)\|_Q^2 + \sum_{j=1}^M \|\Delta \mathbf{u}_s(t+j-1)\|_w^2$$

8. O passo 7 calcula  $\mathbf{u}_s$  usando:

$$\mathbf{Q} = \frac{\mathbf{1}}{\sqrt{\mathbf{1} + \mathbf{y}_{z,s}}}$$

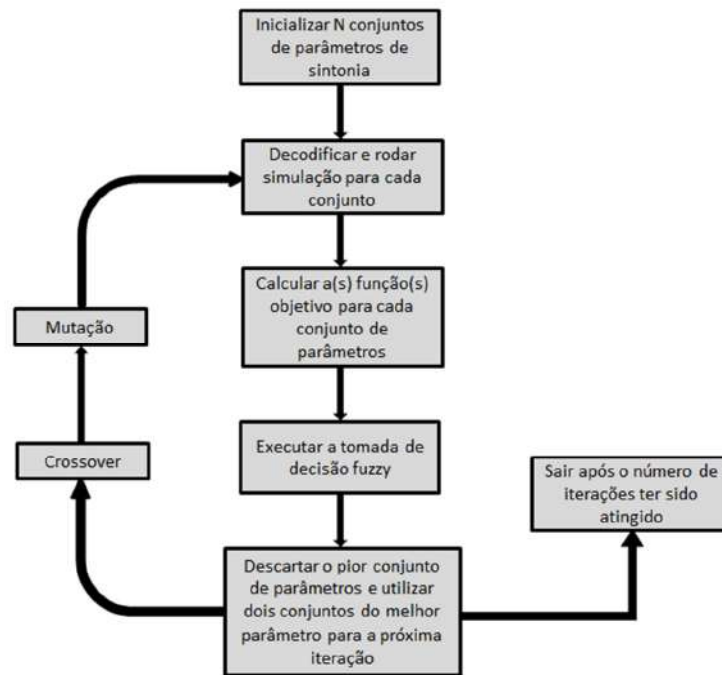
$$\mathbf{w} = \sqrt{(\mathbf{1} + \mathbf{u}_{z,s}) \log_{10}(\text{RPN} + 1) \text{média}(|g_s^{ij}(\omega_{sup})|)}$$

9. Calcular  $\mathbf{u} = \mathbf{R}_s \cdot \mathbf{u}_s$

Fonte: Trierweiler & Farina (2003)

Outra proposta que surgiu para a sintonia de controladores preditivos foi elaborada por Lee, Svrcek e Young (2008). A abordagem tem como base a combinação de algoritmos genéticos com lógica *fuzzy* multiobjetivo para tomada de decisão. Os autores destacam que as vantagens desse tipo de metodologia, é que algoritmos genéticos não são específicos para um tipo de problema, e apenas precisam ser adaptados para ficar de acordo com o número de parâmetros de sintonia e trabalhar dentro do range de cada variável. Assim como a lógica *fuzzy* para tomada de decisão pode lidar com mudanças de ponto ótimo para diferentes sistemas, pode ainda trabalhar com múltiplas entradas para determinar qual conjunto de parâmetros mais se aproxima da solução ótima.

O algoritmo proposto pelos autores pode ser visualizado através da Figura 4:



**Figura 4 - Algoritmo de Sintonia**

Fonte: Lee, Svrcek & Young (2008)

O primeiro passo é a inicialização, em uma população de indivíduos é criada N (conjuntos solução). Na segunda etapa, como cada indivíduo é composto por um grupo de números binários (cada número binário é um cromossomo e cada cromossomo é uma variável da função objetivo), esses números precisam ser traduzidos e uma simulação ser rodada a partir deles. Com os dados de simulação, a função objetivo é avaliada para cada conjunto de parâmetros no terceiro passo. Na quarta etapa, a lógica fuzzy multiobjetivo de tomada de decisão recebe os valores da função objetivo e os ranqueia da melhor para a pior solução. Essa ordem então é informada ao quinto bloco, que elimina o conjunto de parâmetros que gerou a pior solução e duplica o que gerou a melhor. Até atingir o número máximo de iterações, os valores dos parâmetros são atualizados (Crossover e mutação) na direção dos melhores resultados e, assim, a solução converge para um ótimo.

Ainda na linha multiobjetivo, Costa (2009) propõe a sintonia das matrizes dos pesos da variáveis de saída,  $\Gamma$ , e das ações de controle,  $\Lambda$ , através da minimização da seguinte otimização multiobjetivo:

$$\min_{\Gamma, \Lambda} \{f y_1, f y_2, \dots, f y_{ny}, f u_1, f u_2, \dots, f u_3\} \quad (23)$$

Onde:

$$f y_i = \sum_{j=0}^{t_{final}} (y_{i,j}^r - \hat{y}_{i,j})^2, \quad i = 1, 2, \dots, ny \quad (24)$$

$$f u_i = \sum_{j=0}^{t_{final}} (\Delta u_{i,j})^2, \quad i = 1, 2, \dots, nu \quad (25)$$

A autora utiliza duas abordagens para aplicação do enxame de partículas para determinação da frente de Pareto. A primeira abordagem ocorre da seguinte forma: o primeiro objetivo é inicialmente otimizado; o segundo objetivo é otimizado na sequência, tendo como partículas iniciais o conjunto de partículas resultante da ultima iteração realizada para otimizar o primeiro objetivo; o terceiro objetivo tem como ponto inicial o conjunto de partículas da ultima iteração do segundo objetivo e assim por diante.

Entretanto, Costa (2009) observou que quando mais de três objetivos eram otimizados, essa abordagem não permitia a obtenção de toda a frente de Pareto. Uma segunda abordagem foi então proposta, que difere da primeira pelo fato de cada objetivo ser otimizado individualmente, de modo que cada uma das funções tem sua própria população de partículas inicializadas, sem a troca de informações entre os diferentes objetivos durante as iterações.

Costa (2009) propõe então um algoritmo de tomada de decisão, descrito pelas seguintes etapas:

- 1) Geração da curva de Pareto, construída através do julgamento da dominância de pontos resultantes da aplicação do método estocástico de enxame de partículas. Em cada caso, devem ser testados diferentes números de partículas e iterações, até que a região de pareto obtida possa ser considerada bem explorada.

- 2) Os pontos da curva de Pareto são então classificados quanto a robustez, indicando as soluções mais comprometidas com este quesito.

Segundo Costa (2009) a classificação é feita da seguinte maneira: a partir do conjunto ótimo de Pareto, as variáveis de decisão, incluindo variáveis de entrada do modelo e parâmetros, são perturbadas segundo um planejamento experimental e os modelos são recalculados, de modo a obter novos valores das funções objetivo. A ideia é avaliar o quanto estes valores estão distantes dos valores pertencentes ao conjunto ótimo de Pareto por meio de uma métrica estabelecida. Quanto mais distantes, menos robusta é a solução base analisada, devendo ser evitada durante a implementação dos resultados.

Seborg *et al* (2011) sugerem ainda algumas recomendações para estimação de parâmetros de sintonia. Para o tempo de amostragem,  $\Delta t$ , e para o horizonte do modelo,  $N$ , o autor sugere que estes valores sejam escolhidos de modo que  $N\Delta t = t_s$ . A correta escolha desses parâmetros garante que o modelo reflita o efeito completo de uma mudança em uma variável de entrada no tempo necessário para atingir o estado estacionário.

Para os horizontes de predição ( $P$ ) e de controle ( $M$ ), o autor cita algumas regras, levando em consideração, para o valor de  $P$ , que ele seja selecionado de maneira que o efeito completo da última ação de controle seja levado em conta:

$$5 \leq M \leq 20 \quad (26)$$

$$N/3 \leq M \leq n/2 \quad (27)$$

$$P = N + M \quad (28)$$

Quanto aos valores das matrizes pesos, o autor apenas fornece algumas informações conceituais sobre os seus efeitos e sua forma e cita diferentes alternativas para implementá-las no algoritmo de controle, como, por exemplo, variá-la dentro do horizonte de predição

Recentemente, Suzuki *et al* (2007 e 2008), publicaram artigos em que um método de sintonia baseado em otimização por enxame de partículas é utilizado, mas diferente de Costa (2009) não se trabalha com a formulação multiobjetivo. E, em ambos os artigos, os resultados mostrados após

simulações foram satisfatórios. Entretanto, apenas as penalidades da função objetivo foram otimizadas, deixando, mais uma vez, os horizontes como valores fixos.

### 2.3 Otimização por Enxame de Partículas

O conceito de otimização de funções por meio de enxame de partículas foi introduzido por Kennedy e Eberhart (2001). Segundo Das *et al* (2008), a função a qual se deseja achar o ótimo global pode ser representada da seguinte maneira:

$$f(x_1, x_2, x_3, \dots, x_n) = f(\vec{X}) \quad (29)$$

Precisa-se então encontrar o valor de  $\vec{x}$  tal que a função  $f(\vec{x})$  tenha um valor mínimo, ou máximo, de acordo com o que se deseja.

Trata-se de uma técnica de busca paralela multi-agente, onde partículas (ou pássaros), que são entidades conceituais, voam por um espaço de busca multidimensional e que, em qualquer instante de tempo, tem determinadas uma posição e uma velocidade. De forma que o vetor posição da partícula, com relação à origem do espaço de busca, representa uma possível solução.

No início, a população de partículas é inicializada com posições e velocidades aleatórias, denotadas pelos vetores  $\vec{x}_i$  e  $\vec{v}_i$ , respectivamente. Uma relação de vizinhança é então definida no enxame, que determina se duas partículas,  $P_i$  e  $P_j$ , por exemplo, são vizinhas ou não. Entretanto, as versões mais populares de PSO utilizam toda a população como a vizinhança de cada partícula.

Cada partícula  $P$  tem duas variáveis de estado: a posição  $\vec{x}(t)$  e a velocidade  $\vec{v}(t)$  determinados no instante de tempo atual. Também está presente uma pequena memória que consiste na melhor posição da partícula,  $\vec{p}(t)$ , até o momento (o valor da variável que forneceu o maior, ou menor, valor da função) e na melhor posição achada até o momento nas vizinhanças da partícula,  $\vec{g}(t)$ .

Dessa forma, são listados os seguintes parâmetros para o PSO:



- a)  $V_{max}$  que restringe  $\vec{v}_i(t)$  ao intervalo  $[-V_{max}, V_{max}]$ ;
- b) Um fator de peso inercial  $\omega$ ;
- c) Dois números aleatórios uniformemente distribuídos,  $\varphi_1$  e  $\varphi_2$ , que determinam, respectivamente, a influência de  $\vec{p}(t)$  e  $\vec{g}(t)$  nas atualizações da velocidade da partícula;
- d) Duas constantes,  $C_1$  e  $C_2$ , conhecidas como “*self-confidence*” e “*swarm confidence*”, respectivamente.

Inicialmente, os valores de  $\vec{p}(t)$  e  $\vec{g}(t)$  são  $\vec{p}(t) = \vec{g}(t) = \vec{x}(0)$  para todas as partículas. Uma vez que ocorre a inicialização, o processo iterativo começa e as posições e velocidades das partículas são alteradas de acordo com as Eq. (30) e Eq. (31) a seguir, apresentadas para a  $d$ -ésima dimensão da  $i$ -ésima partícula:

$$V_{id}(t + 1) = \omega v_{id}(t) + C_1 \varphi_1 (p_{id}(t) - x_{id}(t)) + C_2 \varphi_2 (g_{id}(t) - x_{id}(t)) \quad (30)$$

$$x_{id}(t + 1) = x_{id}(t) + v_{id}(t + 1) \quad (31)$$

O primeiro termo na fórmula da velocidade representa a velocidade inercial da partícula, sendo  $\omega$  é chamado de fator inercial. Ele é responsável pelo controle de *momentum* da partícula: se  $\omega \ll 1$ , mudanças bruscas de direção podem ocorrer, pois pouca informação do instante de tempo anterior é mantida; se  $\omega = 0$ , o conceito de velocidade é perdido e a partícula se move livremente sem saber da velocidade no instante anterior; se  $\omega > 1$ , as partículas ficam o tempo todo mudando de direção, o que dificulta a convergência, efeito semelhante ao ocorrido quando  $C_1$  e  $C_2$  são muito pequenos.

O parâmetro  $C_1$  é denominado “*self-confidence*”, pois é responsável pela adaptação da velocidade da partícula na direção do melhor valor obtido por ela até o momento. Já  $C_2$  é denominada “*swarm confidence*” pelo fato de regular a adaptação da partícula na direção do melhor valor obtido até o momento por todo o enxame. E, por fim,  $\varphi_1$  e  $\varphi_2$  são parâmetros uniformemente distribuídos no intervalo  $[0,1]$  e tem a função de verificar aleatoriamente o impacto de  $\varphi_1$  e  $\varphi_2$  e, ainda, contribuem para a busca local e global dos valores ótimos.

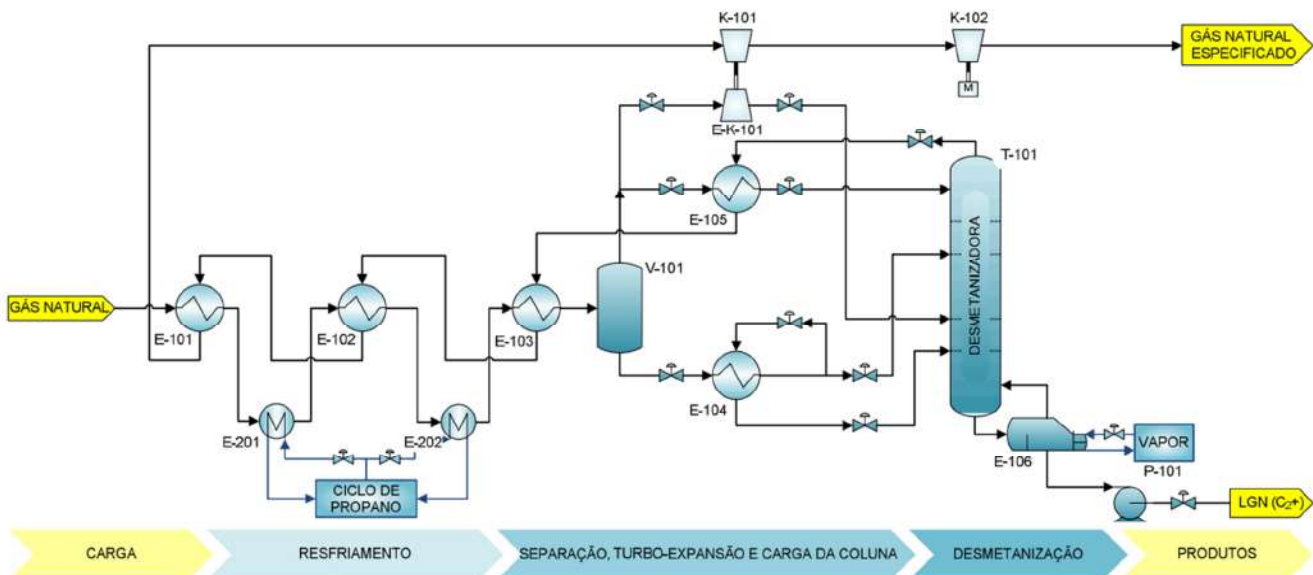
Não há valores ótimos aplicados a problemas gerais para os parâmetros do PSO. Problemas específicos devem ser resolvidos modificando-se ou não tais valores. De modo que os valores ótimos seguem o conjunto de parâmetros que mais vezes gerou um resultado satisfatório para a função objetivo.

### 3 METODOLOGIA

#### 3.2 Exemplo de Processo

A presente seção visa discorrer sobre o processo utilizado para validação do método de sintonia proposto. Serão apresentadas as condições consideradas representativas para a caracterização de uma unidade típica de produção de gás natural, baseada na tecnologia de turbo-expansão.

Essa unidade de processamento de gás natural tem como objetivo especificar o gás conforme os padrões da ANP, definidos na resolução ANP Nº 16, de 17 de junho de 2008, e recuperar o líquido de gás natural para posterior processamento. A Figura 4 mostra, de forma sintetizada, uma unidade de processamento de gás natural considerando a tecnologia aqui adotada:



**Figura 5 - Unidade de Processamento de Gás Natural Baseada em Turbo-Expansão**

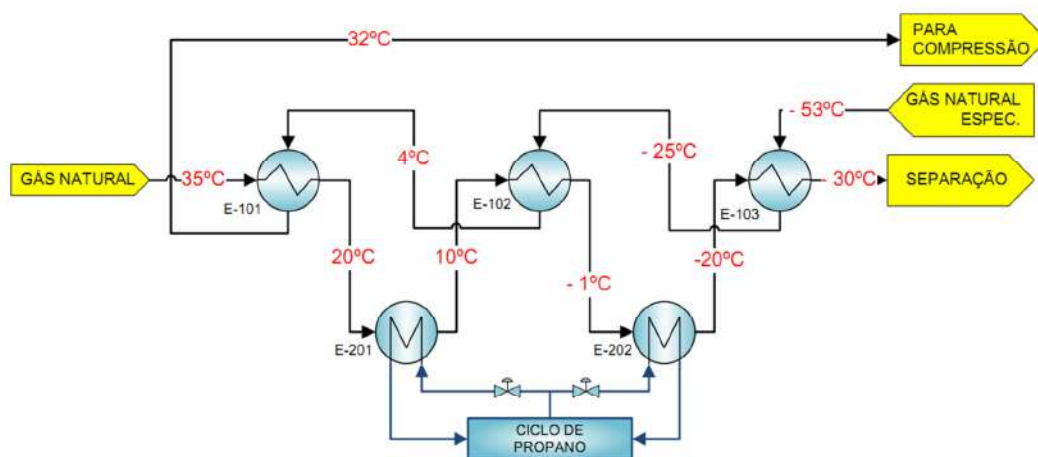
O gás que chega à unidade já foi pré-tratado para retirada de compostos ácidos, como  $H_2S$  e  $CO_2$ , e de água e, portanto, não vão danificar os equipamentos e materiais. Entretanto, ele ainda não está especificado conforme determinado pela ANP, de modo que, pelo seu processamento, pode-se recuperar líquidos para produzir compostos de maior valor agregado, como etano petroquímico, propano e butano ou GLP e nafta leve.

Abaixo será feita uma descrição um pouco mais detalhada das etapas do processo: resfriamento, separação, turbo-expansão e cargas da coluna, e, a destilação na torre e compressão de exportação.

### Etapa de Resfriamento

Esta é a primeira etapa do processo e tem como objetivo basicamente resfriar a carga que chega à unidade. Nela são utilizadas duas fontes de sumidouros de calor: a corrente do topo da desmetanizadora (realizando, dessa maneira, uma integração energética) e propano de refrigeração.

Abaixo segue o esquema de arranjo para os trocadores de calor da etapa de resfriamento, sendo ilustradas as temperaturas de entrada e saída de cada trocador de calor:



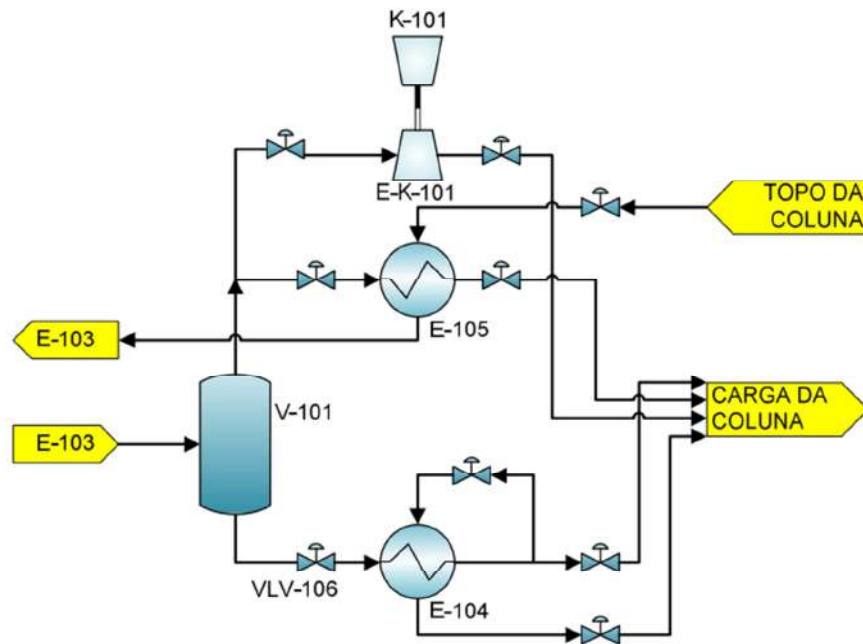
**Figura 6 - Etapa de Resfriamento**

Nesta etapa, a principal variável de controle é a temperatura de saída do ultimo trocador de propano, E-202, e isto pode ser feito pela manipulação da vazão de fluido refrigerante, sendo a temperatura limitada pela capacidade da unidade de refrigeração.

### Etapa de Separação, Turbo-expansão e Carga da Coluna

Durante a etapa de resfriamento, ocorre a condensação dos compostos mais pesados da mistura de hidrocarbonetos. Assim, nessa segunda etapa, a corrente que chega é bifásica sendo necessário então um vaso de separação gás-líquido, denominado Vaso Separador Criogênico, V-101.

A seguir é mostrado um fluxograma que representa esta etapa do processo:



**Figura 7 - Etapa de Separação, Turbo-Expansão e Carga da Coluna**

Durante a operação da unidade, o separador gás líquido deve ter a sua temperatura, que atinge cerca de  $-30^{\circ}\text{C}$ , e o seu nível garantidos para um operação eficiente e segura. O gás que sai do vaso separador, ainda em alta pressão, é dividido em duas correntes: um vai para o turbo-expansor, E-K-101, e a outra vai para E-105, onde é condensada pelo cruzamento com a corrente de topo da coluna desmetanizadora, T-101, a corrente com a menor temperatura do processo. Essa configuração faz com que a carga de topo da coluna desmetanizadora seja líquida.

A corrente líquida oriunda do vaso de separação é então expandida em dois estágios, havendo a presença de um trocador de calor, E-104, entre as duas expansões. A troca térmica que ocorre entre a corrente de baixa pressão e a de pressão média é feita de forma a se aproveitar o calor de expansão.

#### Etapa de Desmetanização e Compressão para Exportação

A desmetanização é a principal etapa na unidade de processamento de gás natural, pois nela é feita a especificação do produto e é onde se determina a

possibilidade dos demais produtos a partir do LGN (líquido de gás natural). A coluna tem a função de separar o metano no topo e etano e mais pesados no fundo.

Devido às temperaturas extremamente baixas no topo da coluna, não há possibilidade de utilização de um condensador, pois seria necessária uma fonte fria externa com temperaturas na ordem de  $-110^{\circ}\text{C}$ . Assim, apenas o refeedor e as condições das cargas causam o perfil de temperatura desejado na coluna.

O gás do topo da coluna troca calor com a carga no trocador E-105 e, subsequentemente, com todos os trocadores da seção de resfriamento. Em seguida, é comprimido no compressor turbo-expansor, K-101, e no compressor adicional, K-102, para que o gás seja exportado dentro das especificações, que contém basicamente metano comprimido. Já a corrente de LGN, no fundo da coluna, é bombeada para as próximas etapas do processo, onde ela vai ser tratada para a produção de Etano Petroquímico, GLP e o produto de nafta leve.

Vale ressaltar que a descrição acima tem o intuito de fazer uma apresentação sucinta do processo. Outras considerações sobre a formulação do processo, suposições feitas, simulação etc., podem ser vistas no trabalho de Cordeiro (2011).

A seguir serão apresentados os modelos lineares identificados para as composições de metano no topo e etano no fundo. Para tanto, Cordeiro (2011) desenvolveu um simulador da unidade no Hysys<sup>®</sup> para gerar dados para a identificação. Cordeiro (2011) propôs também que a camada de controle preditivo estaria hierarquicamente acima de uma camada de controle regulatório, ou seja, em cascata com o mesmo. Assim, o MPC controlava as composições no topo e no fundo da coluna manipulando os *setpoints* dos controladores (dois de temperatura e um de pressão) da camada regulatória. A malha de controle MPC proposta pode ser vista na Figura 8.

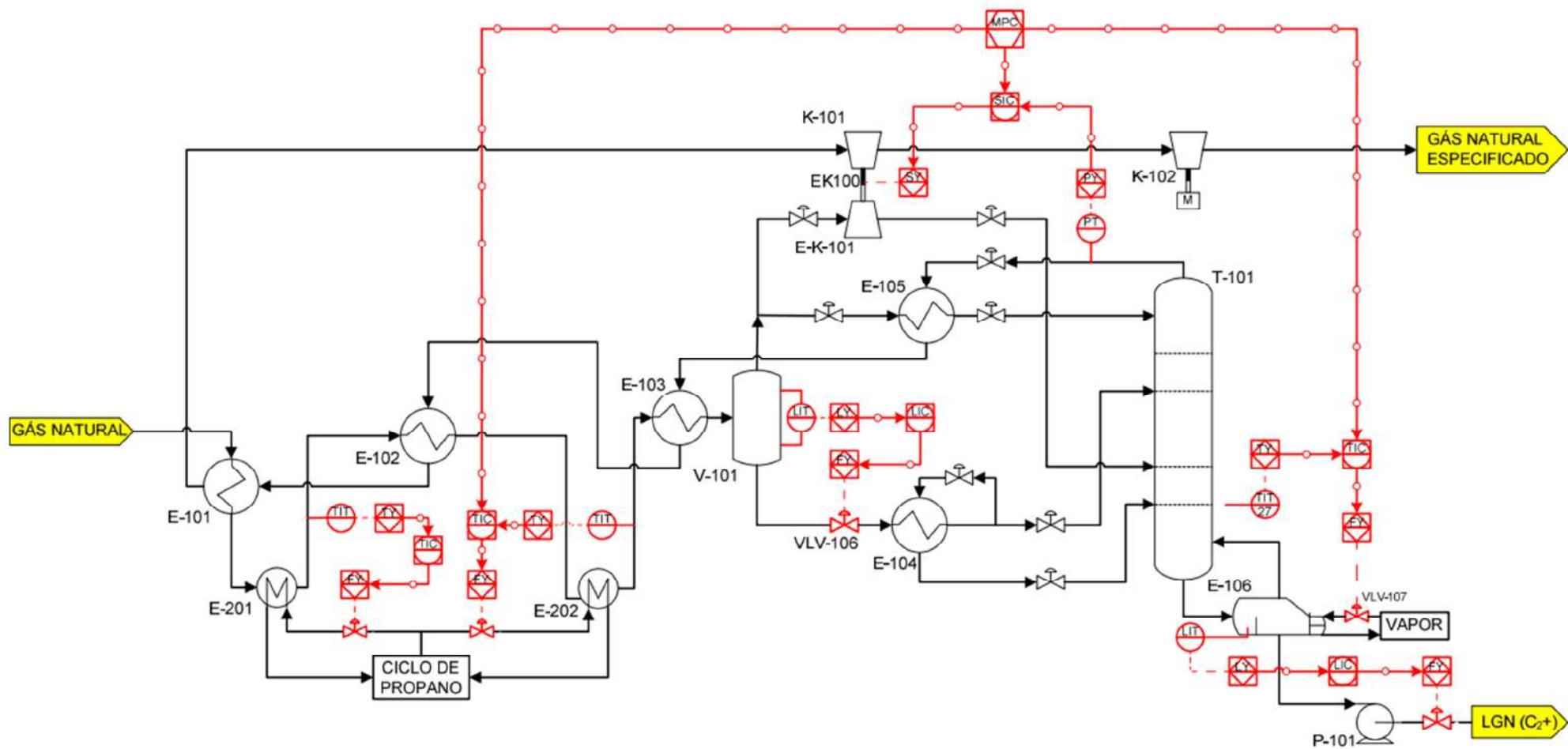


Figura 8 - Malha de Controle Proposta.

A primeira malha de controle regulatório é para controle de temperatura na saída do segundo trocador de propano. O controle é feito pela manipulação da vazão de propano que vai para o trocador E-202. Já a segunda malha é para controle de pressão da coluna, que é feito pela manipulação da velocidade de rotação no eixo do turbo-expansor EK100. A última malha de controle regulatório é para o controle de temperatura no estágio 27 da coluna desmetanizadora e este se dá pela manipulação da vazão de vapor que alimenta o refeedor. Os *setpoints* das variáveis controladas é definido em um nível superior pelo MPC.

#### Temperatura de Saída no Segundo Resfriador de Propano (T1D)

Para a identificação das funções de transferência que descrevem o comportamento das composições no topo e no fundo mediante perturbações no *setpoint* do controlador de temperatura de saída do segundo resfriador de propano, foi aplicado um degrau de -2°C nessa temperatura, dessa forma observou-se o comportamento das composições.

Chegou-se então às seguintes funções de transferência de 1ª ordem com tempo morto:

$$\frac{\overline{X_B}}{\overline{T_{1D,SP}}} = \frac{7,4965 \cdot 10^{-4} e^{-12,1401s}}{13,6210s + 1} \quad (32)$$

$$\frac{\overline{X_D}}{\overline{T_{1D,SP}}} = \frac{2,0180 \cdot 10^{-3} e^{-8,4626s}}{16,2931s + 1} \quad (33)$$

Onde a temperatura está em °C, a fração molar é adimensional e as unidades de tempo estão em minutos.

#### Pressão de Operação da Coluna (P19)

Para identificação, Cordeiro (2011) procedeu da mesma maneira que descrito anteriormente, mas com aplicação de um degrau de 5% no *setpoint* do controlador de pressão do sensor, o que corresponde a 130 kPa. Entretanto, conforme relatado por Cordeiro (2011), a resposta obtida para ambas as composições no topo e no fundo da coluna não apresentaram características de funções de transferência de primeira, segunda e até mesmo de terceira



ordem, com polos e zeros. De modo que se propõe, como alternativa mais rigorosa, um ajuste baseado em modelos que comportem um maior número de parâmetros.

No entanto, para a aplicação proposta, Cordeiro (2011) considerou um modelo simplificado de primeira ordem, no qual se aproxima apenas o ganho estático e o tempo para estabilização, descartando as oscilações iniciais apresentadas pelos dados de simulação:

$$\frac{\overline{X_B}}{\overline{P_{19,SP}}} = \frac{4,3116 \cdot 10^{-5}}{2,3340s + 1} \quad (34)$$

$$\frac{\overline{X_D}}{\overline{P_{19,SP}}} = \frac{5,6120 \cdot 10^{-5}}{0,7059s + 1} \quad (35)$$

Onde a pressão está em kPa, a fração molar é adimensional e as unidades de tempo estão em minutos.

#### Temperatura do Estágio 27 da Coluna (T27)

Conforme já mencionado, a coluna é o equipamento que garante a composição dos produtos. Cordeiro (2011) verificou que o estágio 27 (penúltimo estágio da coluna) era o mais indicado para controle mediante análise de sensibilidade da temperatura de cada prato no estado estacionário. Essa análise foi feita pela verificação dos ganhos relativos em cada prato após perturbações no calor cedido a coluna pelo refeedor ( $Q_{ref}$ ), na vazão molar que passa no trocador E-105, e entra na coluna como refluxo ( $w_{ref}$ ) e na temperatura do separador criogênico V-101 ( $T_{sep}$ ). Foi observado também que o *setpoint* de temperatura no estágio 27 não afetou de forma significativa a composição de topo. E isso ocorre provavelmente pelo fato de não haver condensador, de modo que as condições no topo são determinadas pelas temperaturas das cargas. Assim, essa variável será desconsiderada para o controle de composição no topo.

Realiza-se então um ajuste de função de transferência de primeira ordem para a composição no fundo:

$$\frac{\bar{X}_B}{\bar{T}_{27,SP}} = \frac{1,2320 \cdot 10^{-3}}{0,1985s + 1} \quad (36)$$

Onde a temperatura está em °C, a fração molar é adimensional e as unidades de tempo estão em minutos.

Por fim, chega-se então ao modelo multivariável que será utilizado para validar o método de sintonia aqui proposto, sendo as duas composições,  $X_B$  e  $X_D$ , as variáveis de controle e, como variáveis manipuladas, os *setpoints* de  $T_{1D}$ ,  $P_{19}$  e  $T_{27}$ :

$$\begin{bmatrix} \bar{X}_B \\ \bar{X}_D \end{bmatrix} = \begin{bmatrix} \frac{7,5 \cdot 10^{-4} e^{-12,1s}}{13,6s + 1} & \frac{4,3 \cdot 10^{-5}}{2,3s + 1} & \frac{1,2 \cdot 10^{-3}}{0,2s + 1} \\ \frac{2,0 \cdot 10^{-3} e^{-8,5s}}{16,3s + 1} & \frac{5,6 \cdot 10^{-5}}{0,7s + 1} & 0 \end{bmatrix} \begin{bmatrix} \bar{T}_{1D,SP} \\ \bar{P}_{19,SP} \\ \bar{T}_{27,SP} \end{bmatrix} \quad (37)$$

### 3.3 Sintonia

O objetivo desta seção é propor uma metodologia de sintonia para os parâmetros de controladores MPC, considerando o modelo de processo do tipo *MIMO*, mostrado na seção anterior, utilizando a ferramenta *Model Predictive Control Toolbox™ 3*, disponível no MATLAB®, e usando o algoritmo de Otimização por Enxame de Partículas.

Para a formulação MPC, tem-se que os principais parâmetros de sintonia são o horizonte de controle (M), o horizonte de predição (P), as penalidades das variáveis de saída e as penalidades nas variáveis de entrada. Geralmente, nos métodos propostos de sintonia para controle preditivo, parâmetros como o horizonte de predição e o horizonte de controle são considerados constantes, e tem seus valores determinados de modo que pequenos incrementos nesses valores não tenham efeito significativo no desempenho do controlador. De modo que apenas são considerados para a sintonia as penalidades nas variáveis de saída,  $y_i$ , e nas ações de controle,  $\Delta u$ .

Entretanto, todos os parâmetros citados são importantes para a sintonia, pois todos irão influenciar no desempenho do controlador. Para o horizonte de controle, tem-se que valores pequenos (quando comparados ao horizonte de predição) usualmente levam a respostas do sistema mais lentas e a ações de controle mais conservativas. Por outro lado, quando se tem valores maiores para o horizonte de controle, espera-se um efeito contrário, com respostas mais agressivas do controlador e, além disso, ocorre um aumento do esforço computacional. Já o horizonte de predição tem o efeito de tornar o controlador mais agressivo conforme o seu valor diminui.

Algumas justificativas para uma boa seleção dos horizontes de controle e de predição é dada por Bemporand *et al* (2010), e são conforme segue abaixo:

- Na presença de horizontes de predição suficientemente longos, o controlador pode ver uma violação de restrição em potencial e evita-la ou pelo menos tentar evitar seus efeitos adversos. Observando-se a Figura 9, em que o controlador precisa manter  $y$  abaixo de  $y_{max}$ , verifica-se que um valor para o horizonte de predição igual a  $P_1$ , nenhuma ação é tomada em relação a violação da restrição, o que não ocorre para o valor  $P_2$ , em que a violação é antecipada.

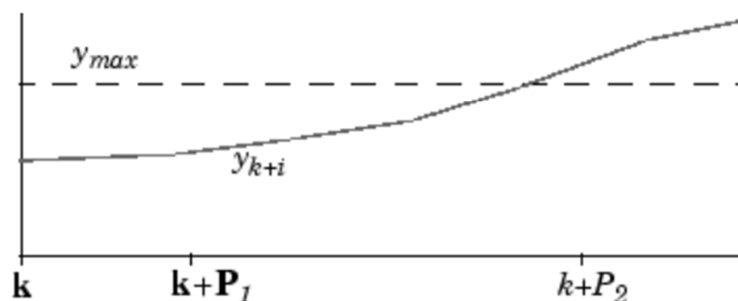


Figura 9 - Efeito do Horizonte de Predição nas Restrições

- Na presença de tempo morto igual a  $D$  intervalos de amostragem, sabe-se que a ação atual de controle não terá efeito até  $y_{k+D+1}$ . Nessa situação, é essencial que  $P \gg D$  e  $M \ll P-D$ ;
- Com plantas que contenham resposta inversa, ou seja, um pequeno termo da resposta em uma direção e todo o restante em outra direção, a otimização deve focar no comportamento global das

variáveis de saída. Caso contrário, o controlador toma ações contrárias, movendo-se na direção errada.

As penalidades são parâmetros mais difíceis de sintonia porque os seus valores dependem da escala das variáveis e, muitas vezes, tais valores não têm significado físico, apenas mensuram, de forma relativa, a importância de cada uma das variáveis. Desse modo, a relação entre os valores tem mais importância do que o seu valor em si.

Para se obter os parâmetros de sintonia, a seguinte função objetivo foi usada para se avaliar a performance do controlador, ou seja, para ser minimizada:

ISE – Integral of the Squared Error:

$$J = \sum_{i=1}^{Tsim} \left[ \sum_{j=1}^{n_y} (w_j^y e_{yij})^2 + \sum_{j=1}^{n_u} [(w_j^u e_{uij})^2 + (w_j^{\Delta u} \Delta u_{ij})^2] \right] \quad (38)$$

A Eq. (38), apresentada acima, está na forma mais geral, de modo que, para o trabalho em questão, o penúltimo termo foi suprimido para permitir que as variáveis manipuladas tenham liberdade para ir para quaisquer valores, sendo feitas apenas restrições nos incrementos delas e no erro da variável controlada em relação ao seu *setpoint*. Assim, a forma utilizada ficou sendo:

$$J = \sum_{i=1}^{Tsim} \left[ \sum_{j=1}^{n_y} (w_j^y e_{yij})^2 + \sum_{j=1}^{n_u} (w_j^{\Delta u} \Delta u_{ij})^2 \right] \quad (39)$$

Onde  $e_{yij}$  é o desvio da variável controlada em relação ao *setpoint* e  $w_j^y$  é a penalização desse erro;  $\Delta u_{ij}$  é o movimento do controlador, sendo  $w_j^{\Delta u}$  a sua penalização. Assim, espera-se que a função objetivo especifique parâmetros que garantam uma boa convergência das variáveis controladas para os seus *setpoint* e evitem mudanças bruscas nas ações de controle.

É importante destacar que as penalizações na função objetivo para estimação dos parâmetros do MPC não são as mesmos da função interna ao controlador, Eq. (6). Esses valores foram escolhidos baseados no objetivo

principal da sintonia, que é atingir o setpoint para as variáveis de saída do processo, e de forma a equalizar o valor de  $\Delta u_{ij}$  para cada variável de entrada. Assim, os valores considerados foram:

$$w_{x_D}^y = w_{x_D}^y = 10 \quad (40)$$

$$w_{T_{1D}}^u = w_{T_{27}}^u = 0,1 \quad (41)$$

$$w_{P_{19}}^u = 1 \quad (42)$$

Na otimização, a PSO permite que restrições sejam impostas para os parâmetros a serem otimizados de forma a evitar que eles caiam em regiões não permitidas. Para os pesos são permitidos quaisquer valores maiores ou iguais a zero, de modo que foi escolhido o intervalo  $[0,1]$ , uma vez que apenas importa a relação entre eles. Para o horizonte de predição, foi escolhido um intervalo entre o valor imediatamente superior ao maior tempo morto da planta, 12,1 min, e o tempo de simulação, de modo que o intervalo ficou  $[7,50]$ , considerando um tempo de amostragem<sup>1</sup> de 2 min. Já para o horizonte de controle foi escolhido um valor entre 1, o menor valor permitido, e 50 intervalos de amostragem.

Entretanto, o universo do SWARM é o dos números reais, dentro das restrições de máximos e mínimos impostas, de modo que qualquer valor pode ser obtido dentro do intervalo permitido a cada iteração. E isso acaba sendo um problema, pois os horizontes de controle e de predição estão restritos a números inteiros e, ainda, em cada avaliação da função objetivo o horizonte de controle tem que ser menor ou igual ao horizonte de predição. Ou seja, uma restrição dinâmica está presente no processo de otimização, pois o limite superior de M é modificado para cada valor de P assumido. Dessa maneira, a PSO teve que ser modificada de forma a incorporar a restrição a valores inteiros para os horizontes de controle e de predição e não permitir que o horizonte de controle fosse maior que o horizonte de predição durante as sucessivas iterações.

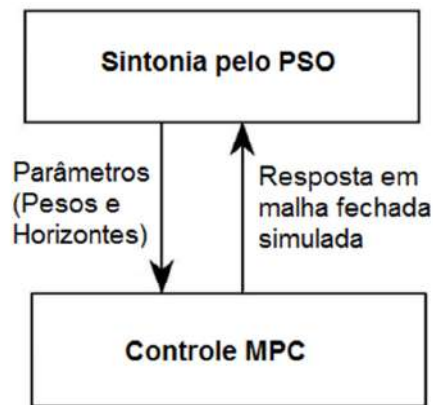
---

<sup>1</sup> Diferente do valor assumido pelo autor do modelo utilizado, Cordeiro (2011), o tempo de amostragem foi considerado igual a 2 minutos, baseado na seguinte diretriz presente em Segorg et al (2011):  $0,2 < \Delta t / \theta < 1,0$

Edgar *et al* (2001) relatam o problema de otimização que envolve variáveis discretas (números inteiros) e contínuas (números reais), chamado de “Programação Inteira Mista”. De acordo com os autores, algumas vezes variáveis inteiras podem ser tratadas como contínuas, especialmente quando elas assumem números grandes, e depois serem arredondadas para o inteiro mais próximo. Entretanto, isso leva a uma solução subótima, ainda que seja aceita do ponto de vista prático. Por exemplo, segundo os autores, quando a resposta é um número como 1,3 o arredondamento prejudica a solução.

Conforme foi relatado, o tratamento dos horizontes foi feito como se fossem números reais. Assim, logo após serem gerados de acordo com as Eq. (30) e Eq.(31), eram arredondados para o inteiro mais próximo. E ainda antes de serem avaliados, faz-se uma conferência com relação à imposição de  $M \leq P$ . No caso de  $M > P$ , o horizonte de controle assume o valor uma unidade menor do que o horizonte de predição.

O conceito da sintonia proposta neste trabalho é mostrado na Figura 10:



**Figura 10 - Conceito de Sintonia MPC-PSO**

Os parâmetros, penalidades e horizontes, são calculados pelo bloco PSO e então enviados para o bloco MPC. Esses parâmetros são avaliados no MPC, gerando uma resposta que é enviada para o bloco de sintonia, onde o PSO avalia e calcula a função objetivo novamente. Esse processo é repetido até atingir o número de iterações limite dentro do PSO, que é o critério de parada.

Pode-se então representar o procedimento PSO-MPC da seguinte maneira:

### **1° Passo: Projeto do Controlador**

Uma vez que modelo de processo seja conhecido, o controlador deve ser projetado de acordo com as Eq. (1) – (11). Neste trabalho, o próprio MATLAB® faz esse projeto, bastando apenas entrar com o modelo de processo e com o comando responsável pelo *design*.

### **2° Passo: Especificar a Função Objetivo**

A função objetivo deve ser especificada, pois ela será entrada para o PSO e a sua avaliação será empregada na otimização. A Eq. (39) foi utilizada neste trabalho.

### **3° Passo: Inicialização**

Valores iniciais são atribuídos aos parâmetros que, juntamente com a função objetivo, vão entrar no PSO. A estimativa inicial irá corresponder aos valores das dimensões (número de variáveis) para o primeiro pássaro, ou seja, a sua posição. Outros valores são então gerados de forma aleatória para os demais pássaros. Nesse ponto, as restrições para cada uma das variáveis já se encontram ativas de modo que os valores gerados são corrigidos caso tenham ultrapassado os limites permitidos.

Cada pássaro é então avaliado pelo controlador e gera valores para a função objetivo. Esses primeiro valores ficam armazenados e são considerados como o  $\vec{p}$ , melhor posição do pássaro, e, dentre todos os pássaros, o que gerou o menor valor para a função objetivo, fica sendo o  $\vec{g}$ , melhor posição no enxame.

### **4° Passo: Atualização da Posição**

Usando os valores de posição anteriores e as Eq. (30) e Eq. (31), assim como  $\vec{p}$  e  $\vec{g}$ , calculam-se novas posições e velocidades para os pássaros. Verificam-se as restrições para avaliar uma possível violação dos limites. Avalia-se então novamente a função objetivo para verificar se houve  $\vec{p}$  e  $\vec{g}$  melhores. Caso haja, esses vetores são atualizados e uma nova iteração é feita.

### **5° Passo: Terminação**

O passo anterior se repete até que o número de gerações seja alcançado. Como resposta é fornecido um vetor com os valores ótimos para os parâmetros, o valor da função objetivo nesse ponto e o número de vezes que a função objetivo foi avaliada.



## 4 RESULTADOS

Algumas considerações devem ser feitas antes da apresentação dos resultados:

- 1) Para as sintonias foram considerados degraus nos *setpoints* das variáveis de controle de  $\pm 5\%$  em corridas diferentes, ou seja, para cada corrida, uma variação em uma variável era considerada, formando um total de quatro combinações, conforme Eq. (43):

$$J = J_{(0,05;0)} + J_{(0;0,05)} + J_{(-0,05;0)} + J_{(0;-0,05)} \quad (43)$$

- 2) O tempo de simulação considerado foi de 100 minutos.

Serão então apresentados os resultados de Cordeiro (2011), tomados como referência para posterior comparação. O autor utilizou diversas simulações variando os parâmetros e aplicando uma variação mútua de *setpoint* nas composições  $X_B$  e  $X_D$  de  $-5\%$ . Os horizontes de controle e de predição foram obtidos em uma primeira etapa, considerando as penalidades,  $w_j^y$  e  $w_j^{\Delta u}$ , iguais a 1. Em uma segunda etapa, considerando os valores obtidos para os horizontes, o autor realizou outras simulações para obtenção das penalidades e chegou então ao seguinte resultado:

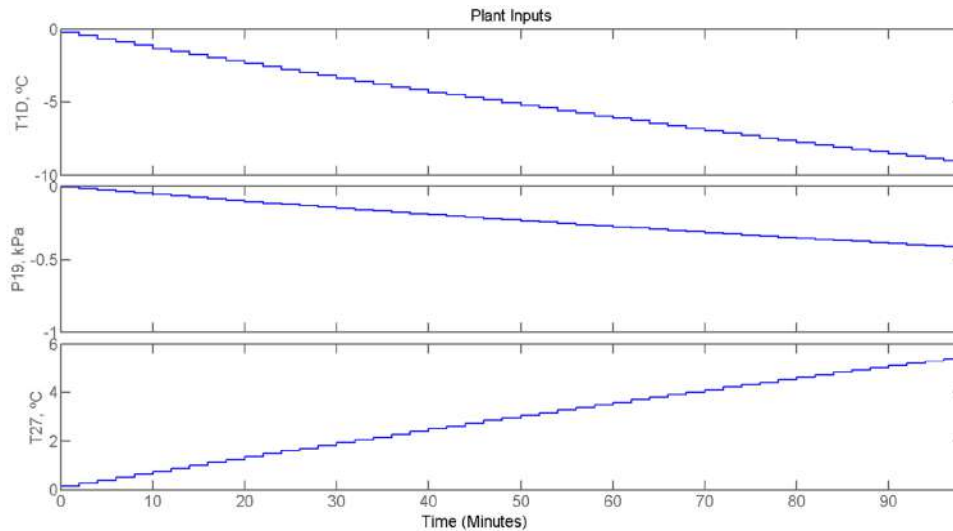
$$P = 60$$

$$M = 10$$

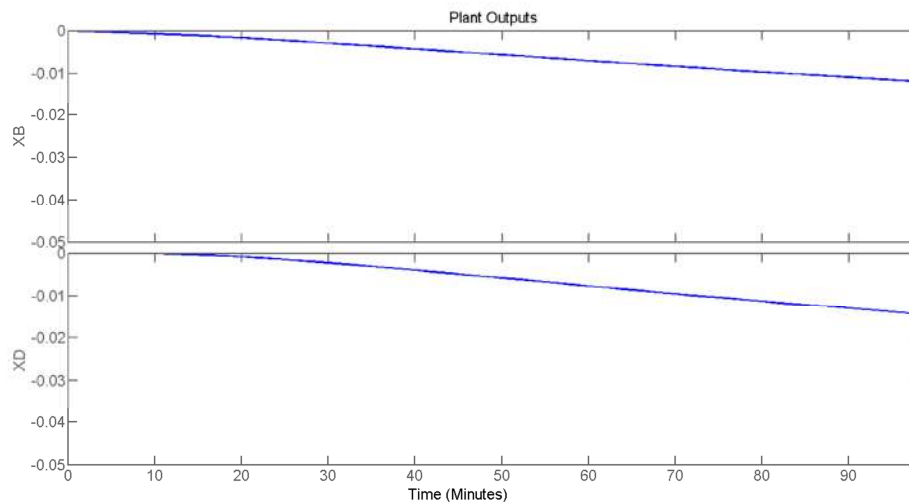
$$w_j^y = 0,6$$

$$w_j^{\Delta u} = 0,1$$

Abaixo seguem os resultados das simulações feitas sob as considerações feitas no início desta seção e considerando os resultados obtidos por Cordeiro (2011):



**Figura 11 - Resposta Cordeiro (2011) - Variáveis Manipuladas**

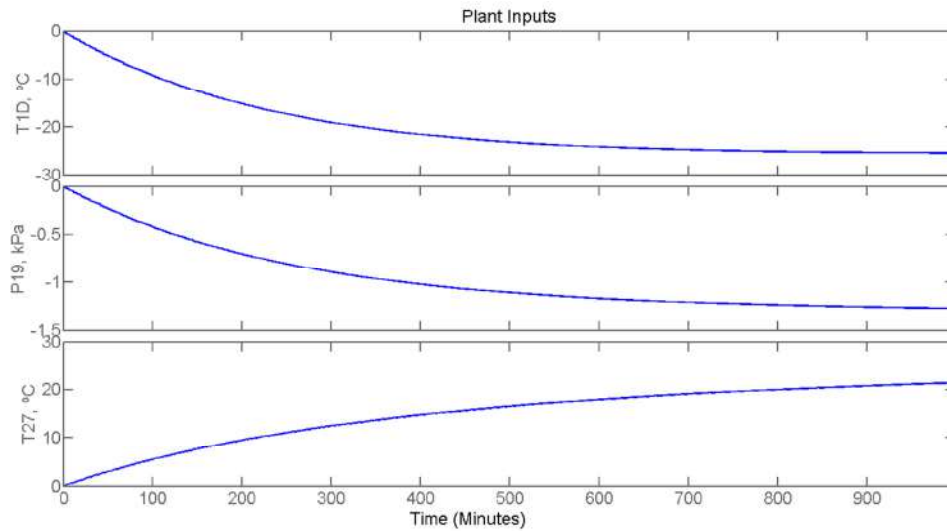


**Figura 12 - Resposta Cordeiro (2011) - Variáveis Controladas**

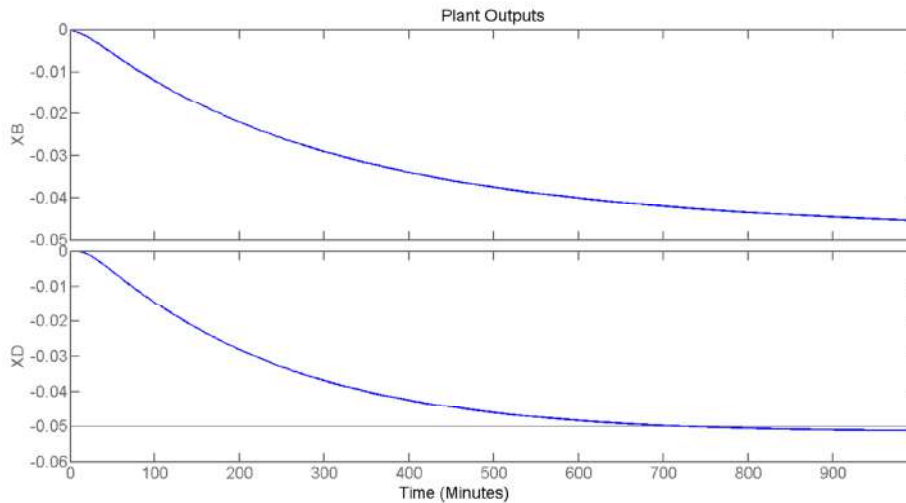
Com o seguinte erro:  $ISE = 19,7103$

Vale ressaltar que as respostas obtidas aqui não são semelhantes às expostas no trabalho de Cordeiro (2011), já que no trabalho do autor o estado estacionário é alcançado de maneira mais rápida dada uma mesma perturbação. Podem ser citadas como origem dessa discordância de resultados o tempo de amostragem, que aqui possui um valor diferente; e o fato de o próprio processo diferir, pois o modelo aqui utilizado foi gerado pelo autor apenas para obter os parâmetros da função degrau e especificar os parâmetros do modelo interno do controlador, de modo que ele controlou um processo mais complexo implementado no simulador de processos Hysys®.

Uma resposta mais aproximada pode ser encontrada aumentando-se o tempo de simulação:



**Figura 13 - Resposta Cordeiro (2011) - Variáveis Manipuladas**



**Figura 14 - Resposta Cordeiro (2011) - Variáveis Controladas**

Entretanto,  $X_B$  continua com offset e  $X_D$  leva em torno de 600 minutos para atingir o novo estado estacionário, o que não é desejado. O próprio autor relata que foi uma sintonia não ótima, mas, diferente dos resultados aqui obtidos, ele consegue obter respostas satisfatórias e sem offset.

Além das considerações já feitas, vale ressaltar que para a metodologia proposta MPC-PSO, foram feitas dois tipos de sintonia:

- 1) Considerando os pesos diferentes para cada variável de entrada e para cada variável de saída, de forma que a função objetivo ficou com sete parâmetros de sintonia, a saber, horizonte de controle,  $M$ , horizonte de predição,  $P$ , pesos dos movimentos de controle,  $w_j^{\Delta u}$  ( $j = 1, \dots, 3$ ) e penalizações nas variáveis de saída,  $w_j^y$  ( $j = 1, 2$ );
- 2) Considerando os mesmos pesos para as variáveis de saída,  $w^y$ , e o mesmo peso para as ações de controle em cada variável,  $w^{\Delta u}$ , de forma que a função objetivo neste caso possui quatro parâmetros de sintonia.

Conforme já foi mencionado, existem duas formas para a função objetivo do controlador no MATLAB®. Entretanto, como não estamos lidando com penalizações variáveis dentro do horizonte de predição, elas se tornam similares, bastando que se escreva na equação alternativa os pesos elevados à segunda potência como os elementos da matriz diagonal:

$$w^y = Q = \begin{bmatrix} w_1^{y^2} & 0 \\ 0 & w_2^{y^2} \end{bmatrix}$$

$$w^{\Delta u} = R_{\Delta u} = \begin{bmatrix} w_1^{\Delta u^2} & 0 & 0 \\ 0 & w_2^{\Delta u^2} & 0 \\ 0 & 0 & w_3^{\Delta u^2} \end{bmatrix}$$

De modo que toda a formulação foi feita usando-se a primeira forma da função objetivo do controlador, Eq. (6).

Com relação ao PSO, foram considerados os seguintes valores para os parâmetros, quase todos mantidos com os valores default, conforme pode ser visto no apêndice A1, exceto o tamanho da população, que foi modificado visando uma maior espectro na geração dos valores iniciais:

*Numero de iterações* = 50;

*Tamanho da população* = 20;

$\omega = 0,9$  – Decaindo com o número de iterações a um passo de  $\frac{\omega - 4 \cdot 10^{-3}}{\text{Nro de iterações}}$

$C_1 = C_2 = 1$

A estratégia de decaimento de  $\omega$  é uma configuração próxima a ótima para muitos problemas de otimização, uma vez que ela permite ao enxame explorar o espaço de busca no início da corrida e, conforme diminui, gerencia uma busca local mais refinada.

Na rotina de PSO utilizada ainda é possível entrar com restrições para as variáveis, de modo que não se caia em uma região que não existe durante as iterações. Dessa maneira, o problema de otimização ficou sujeito a:

$$0 \leq w_j^y \leq 1, \quad j = 1,2$$

$$0 \leq w_j^{\Delta u} \leq 1, \quad j = 1,2,3$$

$$7 \leq P \leq 50$$

$$1 \leq M \leq 50$$

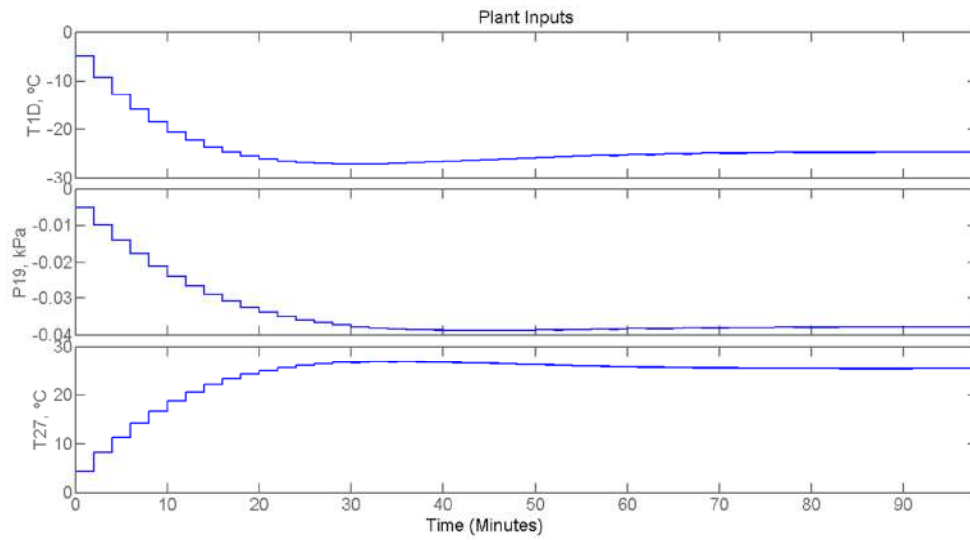
As restrições nos pesos foram impostas de maneira arbitrária, uma vez que a única coisa que importa é a relação entre eles. O horizonte de predição foi colocado igual ao tempo de simulação, observando-se que, em malha aberta e aplicando-se degraus nas variáveis manipuladas, o maior tempo para atingir o novo estado estacionário era cerca de 90 minutos. O horizonte de controle foi escolhido também de forma arbitrária, visando limitar o esforço computacional.

Como a otimização por enxame de partículas é baseada em números aleatórios, diferentes estimativas iniciais foram geradas, o que levou a diferentes resultados, mas todos eles muito próximos. A Tabela 5 resume os resultados, que foram obtidos a partir de 200 loops de otimização considerando todos os pesos diferentes, conforme consideração feita anteriormente:

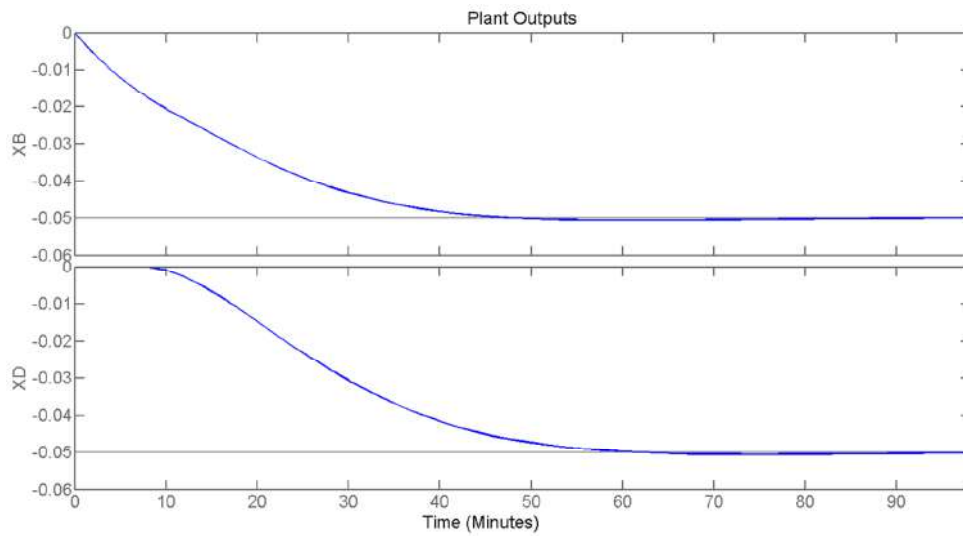
**Tabela 5 - Resultados da Sintonia MPC-PSO para pesos diferentes**

	J	$W_{T1D}$	$W_{P19}$	$W_{T27}$	$W_{XB}$	$W_{XD}$	M	P
Melhor	11,7220	0,0098	0,0980	0,0099	0,9922	0,9792	29	49
Pior	11,7316	0,0064	0,1327	0,0096	0,9559	0,6180	17	49

Seguem então os resultados das simulações para ambos os casos e considerando uma variação mútua no setpoint das variáveis controladas de -5%:



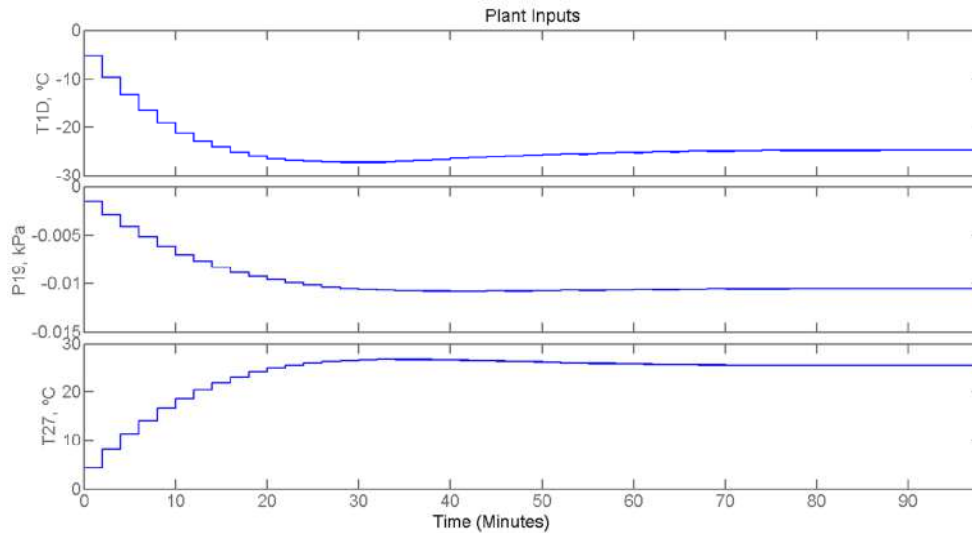
**Figura 15 - Melhor Resultado para Pesos Diferentes - Variáveis Manipuladas**



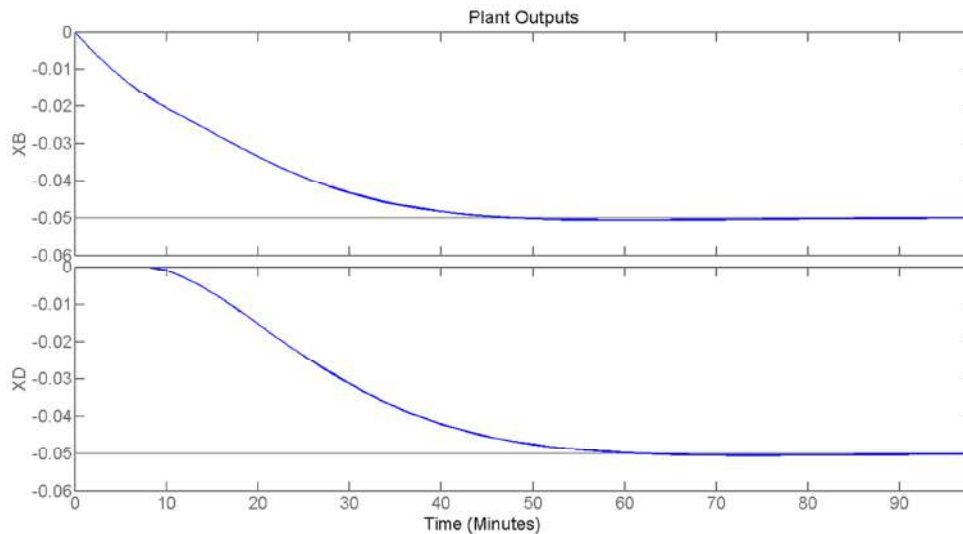
**Figura 16 - Melhor Resultado para Pesos Diferentes - Variáveis Controladas**

Para essa simulação, obteve-se um ISE de 5,5772.

Agora, para o pior caso, tem-se:



**Figura 17 - Pior Resultado para Pesos Diferentes - Variáveis Manipuladas**



**Figura 18 - Pior Resultado para Pesos Diferentes - Variáveis Controladas**

Para essa simulação obteve-se um ISE de 5,5794. Ou seja, comportamentos praticamente idênticos. Isso mostra que, apesar da aleatoriedade do método, ele sempre converge para um ponto ótimo ou próximo ao ótimo.

Além da aleatoriedade do método como causa para os diferentes valores dos parâmetros convergidos, podem ser citadas ainda a quantidade de parâmetros para otimização. No total são sete, o que torna a função objetivo extremamente complexa e difícil de ser avaliada. Pode-se ainda citar o fato de não haver restrições, no processo tomado como referência, para as variáveis manipuladas e nem para as suas variações, de modo que deixou-se que elas

variassem livremente, objetivando apenas minimizar o esforço total e o erro em relação ao setpoint para as variáveis controladas.

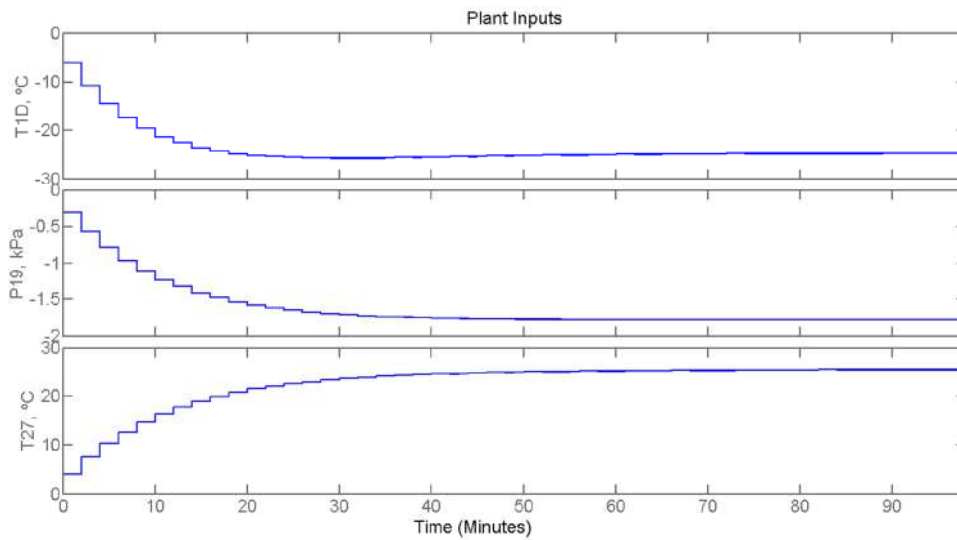
Agora, considerando o segundo caso, no qual o conjunto de variáveis de entrada foi considerado com o mesmo peso,  $w^{\Delta u}$ , e o outro conjunto, das variáveis de saída, foi considerado também com o mesmo peso,  $w^y$ .

Da mesma maneira, após os 200 loops de otimização chega-se ao resultado mostrado na Tabela 6:

**Tabela 6 - Resultados da Sintonia MPC-PSO para pesos iguais**

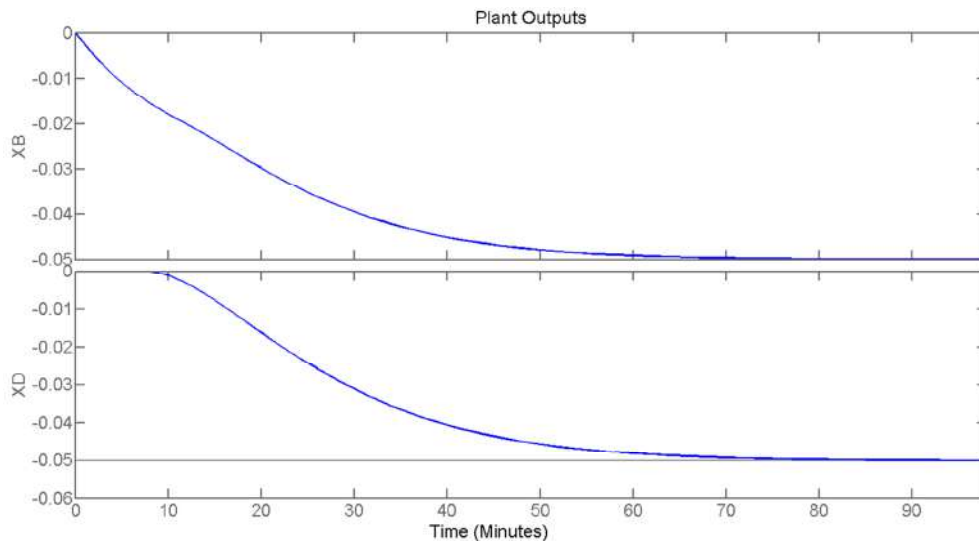
	J	$w^{\Delta u}$	$w^y$	M	P
Melhor	12,2919	0,0226	0,9993	2	50
Pior	12,7778	0,0128	0,9919	40	46

Seguem então os resultados das simulações para ambos os casos e considerando uma variação mútua no setpoint das variáveis controladas de -5%:



**Figura 19 - Melhor Resultado para Pesos Iguais - Variáveis Manipuladas**

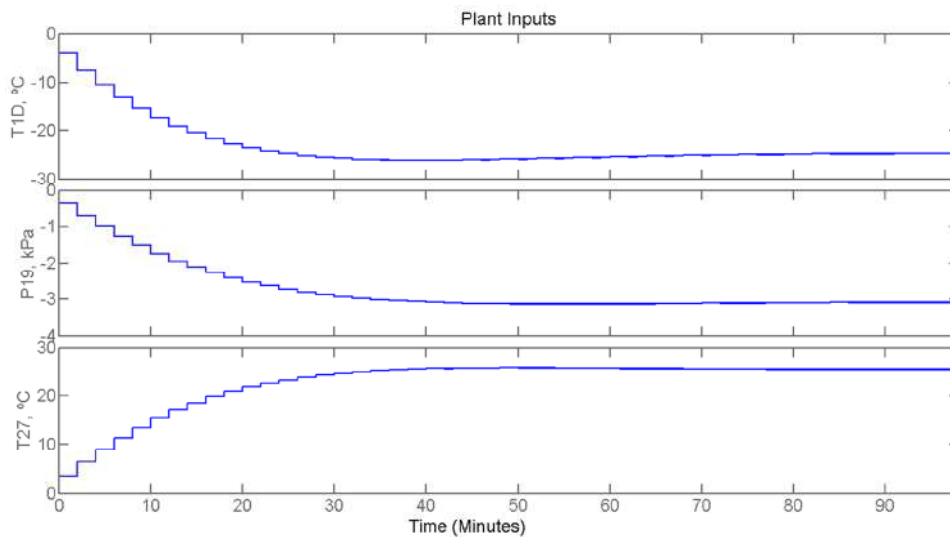




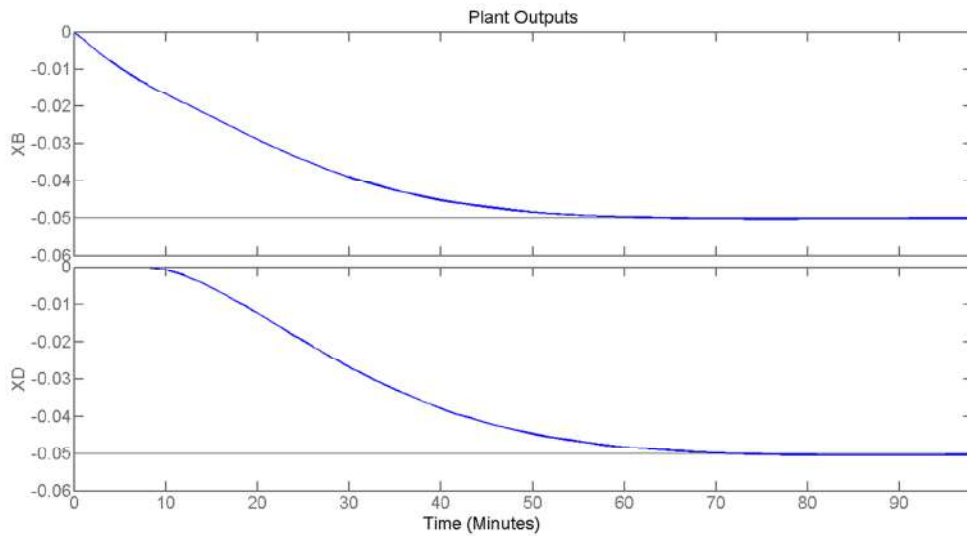
**Figura 20 - Melhor Resultado para Pesos Iguais - Variáveis Controladas**

Para essa simulação obteve-se um ISE de 5,9516, resultado pior que o obtido para o melhor e o pior caso na simulação com pesos iguais.

Tem-se então para o pior caso:



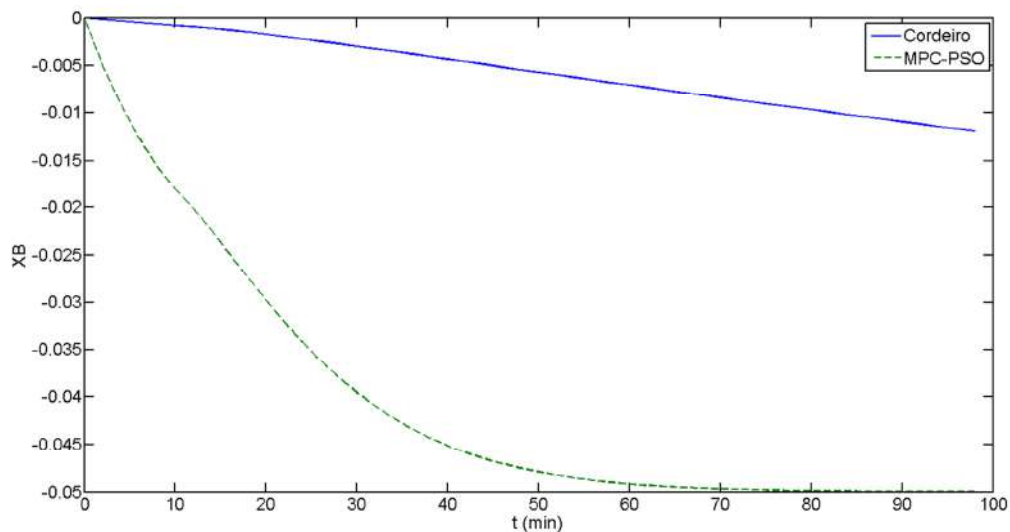
**Figura 21 - Pior Resultado para Pesos Iguais - Variáveis Manipuladas**



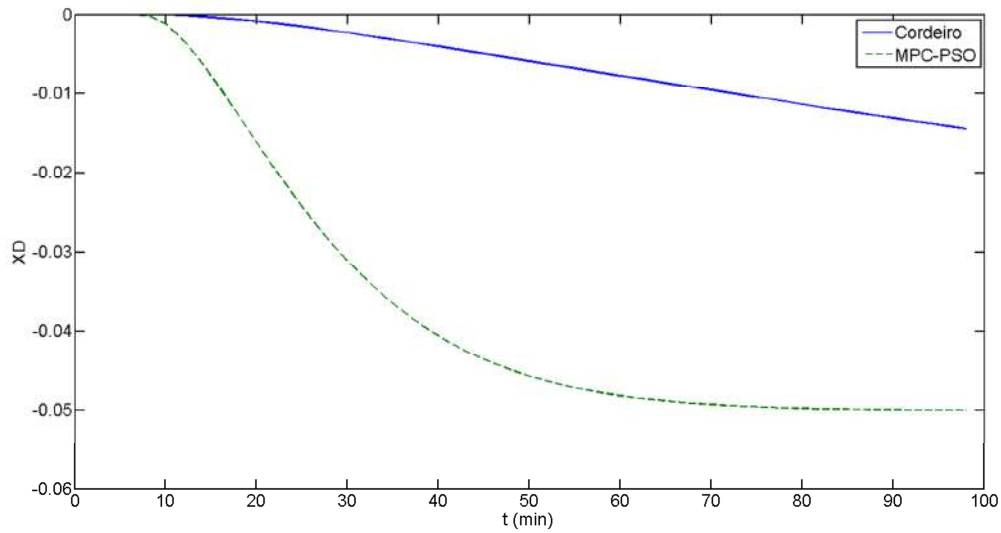
**Figura 22 - Pior Resultado para Pesos Iguais - Variáveis Controladas**

Para esta simulação obteve-se um ISE de 6,3609. De modo que pode-se observar um aumento gradativo do ISE conforme se percorre a sintonia considerando penalidades diferentes e penalidades iguais, melhor e pior caso.

Agora, fazendo uma comparação entre os resultados obtidos com os parâmetros de Cordeiro (2011) e pela metodologia MPC-PSO para o melhor caso, tem-se:



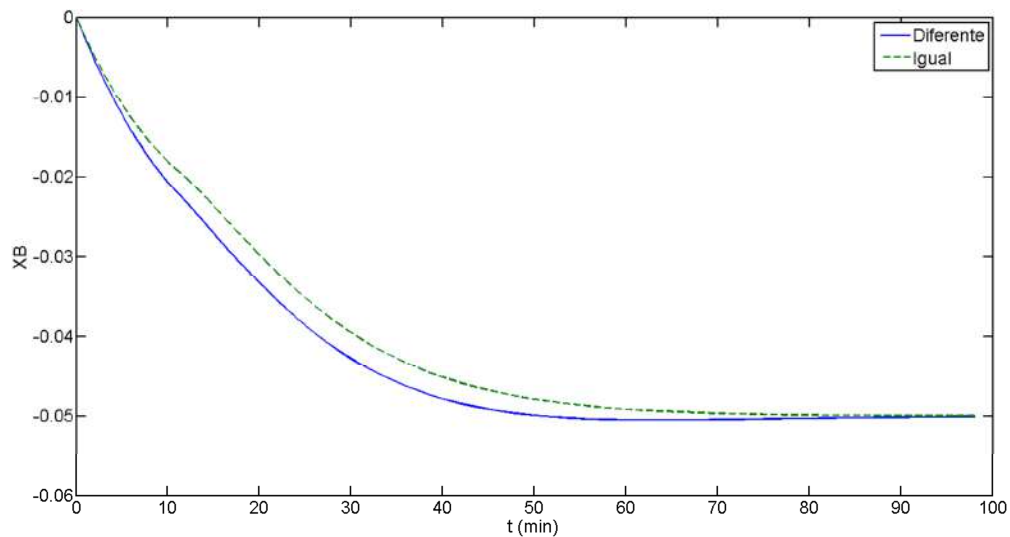
**Figura 23 - Comparação Cordeiro MPC-PSO para XB**



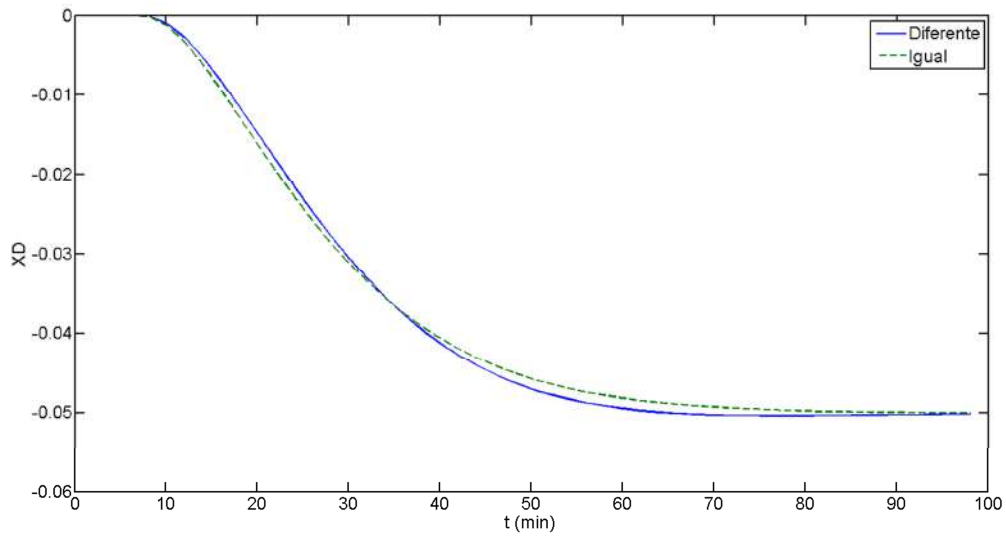
**Figura 24 - Comparação Cordeiro MPC-PSO para XD**

A discrepância entre as duas sintonias é visível. Entretanto, não se pode esquecer das justificativas feitas, pois as condições em que as simulações foram realizadas não são as mesmas.

Fazendo-se agora uma comparação entre as melhores sintonias MPC-PSO, com penalidades diferentes e iguais, chega-se a:



**Figura 25 - Comparação MPC-PSO com os melhores resultados para XD**



**Figura 26 - Comparação MPC-PSO com os melhores resultados para XD**

De modo que verifica-se a concordância de resultados tanto usando pesos iguais para as variáveis de saída e de entrada, quanto usando pesos diferentes. Mostrando que a metodologia pode alcançar valores ótimos sob diferentes restrições.

## 5 CONCLUSÕES E SUGESTÕES

A sintonia de controladores preditivos é um campo ainda não estabelecido, e diversos estudos vêm sendo realizados para o desenvolvimento de metodologia para ajuste dos parâmetros de sintonia, conforme pode ser visto na revisão bibliográfica. E neste trabalho uma metodologia, até então pouco abordada, de sintonia baseada em otimização por enxame de partículas foi proposta.

Nos testes realizados, mostrou-se que os resultados foram satisfatórios, havendo convergência das variáveis de controle para o valor desejado. Mostrou-se também a semelhança entre as simulações utilizando os valores obtidos para os parâmetros considerando pesos distintos para todas as variáveis, ou seja, a função objetivo com sete parâmetros de sintonia, e também considerando um peso para o grupo de entrada e outro para o grupo de saída, ou seja, a função objetivo com quatro variáveis, evidenciando a convergência do método para o ótimo mesmo sob diferentes condições.

A metodologia, conforme apresentada no desenvolvimento deste trabalho, apresenta alguns méritos em relação a outras até então propostas:

- Abordagem é feita considerando um número de parâmetros maior do que os que foram abordados até agora na literatura (pesos e horizontes) e ainda mostra como lidar com o universo permitido para cada um deles (programação inteira e inteira mista);
- A abordagem foi elaborada de modo a considerar toda a estrutura de projeto de MPC do MATLAB<sup>®</sup> (que é uma abordagem considerada o “estado da arte”, baseada em variáveis de estado);

Algumas limitações também podem ser listadas:

- Metodologia pouco testada: para este trabalho apenas um modelo de processo foi testado, e mesmo juntando todos os modelos usados nos trabalhos de referência MPC-PSO, a metodologia ainda carece de testes;
- Não se testou explicitamente o uso de restrições;

- Todo o procedimento de sintonia foi baseado nos valores padrões do algoritmo de otimização utilizado;
- A sintonia deve ser feita *off-line* e demanda um esforço computacional devido ao número de loops para o cálculo a partir de diferentes estimativas iniciais.

Desta maneira, deixa-se essas limitações como sugestão para trabalhos futuros para a sintonia de controladores preditivos.

## REFERÊNCIAS BIBLIOGRÁFICAS

Al-GHAZZAWI, A., Ali, E., NOUH, A., & ZAFIRIOU, E., Online Tuning Strategy for Model Predictive Controllers, *Journal of Process Control*, vol. 11, pp. 265-284, 2001.

BEMPORAND, A., MORARI, M., & RICKER, N. L. *Model Predictive Control Toolbox™ 3 – User's Guide*. Matlab. The Mathworks, Inc., 2010.

CORDEIRO, Alexandre de França. *Estudo de Simulação e Controle de uma Unidade de Processamento de Gás Natural*, Dissertação de Mestrado, EQ/UFRJ, Rio de Janeiro, pp. 138., 2011

COSTA, Livia Pereira Lemos. *Desenvolvimento de Metodologia de Tomada de Decisão para Problemas de Otimização Multiobjetivo de Engenharia Química*, Dissertação de Mestrado, COPPE/UFRJ, pp. 81, 2009

CUTLER, C. R., & RAMAKER, B. L.. *Dynamic Matrix Control – A Computer Control Algorithm*. AICHE National Meeting, Houston, TX, Abril, 1979.

CUTLER, C., MORSHEDI, A., & HAYDEL, J. *An Industrial Perspective on Advanced Control*. Em AICHE Anual Meeting, Washington, DC, Outubro, 1983.

DAS, S., ABRAHAM, A., & KONAR, A. *Particle Swarm Optimization and Differential Evolution Algorithms: Technical Analysis, Applications and Hybridization Perspectives*. *Studies in Computational Intelligence (SCI)* 116, pp. 1-38, 2008

EDGAR, T. F., HIMMELBLAU, D. M, & LASDON, L. S. *Optimization of Chemical Processes*. 2nd ed. McGraw-Hill, 2001

FROISY, J.B., & MATSKO, T. *IDCOM-M Application to the Shell Fundamental Control Problem*. AICHE annual meeting, Novembro 1990.

GARCIA, C. E., & MORSHEDI, A. M. *Quadratic Programming Solution of Dynamic Matrix Control (QDMC)*. *Chemical Engineering Communications*, 46, 73-87, 1986.

GARCIA, C. E., PRETT, D. M. & MORARI, M. *Model Predictive Control: Theory and Practice – A Survey*. *Automatica*, 25(3), pp. 335-348, 1989

GROSDIDIER, P., FROISY, B., & HAMMAN, M. (1988). *The IDCOM-M controller*. Em T. J. McAvoy, Y. Arkun, & Zafiriou (Eds.), *Proceedings of the 1988 IFAC Workshop on Model Based Process Control*, pp. 31-36, Oxford: Pergamon Press.

LEE, J.H., W.Y. SVRCEK & B.R. YOUNG. A tuning algorithm for model predictive controllers based on genetic algorithms and fuzzy decision making. ISA Transactions, Volume 47, Issue 1, Janeiro 2008, Pages 53-59.

KANO, M., & OGAWA, M. The state of the art in chemical process control in Japan: Good practice and questionnaire survey. Journal of Process Control, vol. 20, pp. 969–982, 2010

LEE, J. H., & YU, Z. H. Tuning of Model Predictive Controllers for Robust Performance. Comput. Chem. Eng., Vol.18, pp. 15-37, 1994.

MACARTHUR, J. W. RMPCT: A New Robust Approach to Multivariable Predictive Control for the Process Industries. Proceedings of the 1996 Control System Conference, Halifax, Canada, pp. 53-60, 1996.

MARQUIS, P. & BROUSTAIL, J. P. SMOC, A Bridge Between State-Space and Model Predictive Controllers: Application to the Automation of a Hydrotreating unit. Em T. J. McAvoy, Y. Arkun, & E. Zafiriou (Eds.), Proceedings of the 1988 IFAC Workshop on Model Based Process Control, pp. 37-43, Oxford: Pergamon Press.

MEDEIROS, José Antônio Carlos Canedo. Exame de Partículas como Ferramenta de Otimização em Problemas Complexos de Engenharia Nuclear, Tese de Doutorado, COPPE/UFRJ, pp. 108, 2005.

MORARI, M. & LEE, J. H. Model Predictive Control: Past, Present and Future. Computers and Chemical Engineering. 23, pp. 667-682, 1999

QIN, S. J. & BADGWELL. A Survey of Model Predictive Control Technology. Control Engineering Practice. Vol. 11, Issue 7, pp. 733-764, 2003

RICHALET, J., RAULT, A., TESTUD, J. L., & PAPON, J. Algorithmic Control of Industrial Processes. Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation, pp. 1119-1167, 1976

RICHALET, J., RAULT, A., TESTUD, J. L., & PAPON, J. Model Predictive Heuristic Control: Applications to Industrial Processes. Automatica, vol. 14, pp. 413-428, 1978

SEBORG D. E., EDGAR T. F., MELLICHAMP, D. A. Process Dynamics and Control. 3rd ed., Wiley, 2011

SHRIDHAR, R., & COOPER, D. J. A Novel Tuning Strategy for Multivariable Model Predictive Control. ISA Trans., vol. 36, pp. 273-280, 1998.



SUZUKI, R., KAWAY, F., Ito, R., NAKAZAWA, C., MATSUI, T., AIYOSHI, E. Parameter Optimization of Model Predictive Control Using PSO. SICE Annual Conference, pp. 1981-1988, 2008.

SUZUKI, R., KAWAY, F., ITO, R., NAKAZAWA, C., FUKUYAMA, Y., AIYOSHI, E., Automatic Tuning of Model Predictive Control Using Particle Swarm Optimization. Preceedingd of the 2007 IEEE Swarm Intelligence Symposium.

TRIERWEILER, J. O, FARINA, L. A. RPN Tuning Strategy for Model Predictive Control, Journao Of Process Control, vol. 13, pp. 591-598, 2003

WANG, L. Model Predictive Control System Design and Implementation Using Matlab. Advances in Industrial Control. Springer 1<sup>st</sup> ed, 2009.

WOJSZNIS, M., GUDAZ, J., BLEVINS, T., & MEHTA, Ashish. Practical Approach to Tuning MPC. ISA Transactions 42, pp. 149-162, 2003

YOUSFI C., & TOURNIER, R. Steady-state optimization inside model predictive control. Em Proceedings of ACC'91, Boston, MA , pp. 1866–1870, 1991.

## APÊNDICE A1 – Códigos Utilizados

### A1.1 – Rotina Principal: tuning.m

```
clear all
clc

global T sp MPCobj wts
%% Escolha de Métodos e Chutes Iniciais:

controlador;

% Estimativas Iniciais (Baseado nos Valores Ótimos do Alexandre):

% 1) Considerando otimização para cada um dos pesos:
WDT1D = 0.1; % Penalização da taxa de variação do controlador de T1D.
WDP19 = 0.1; % Penalização da taxa de variação do controlador de P19.
WDT27 = 0.1; % Penalização da taxa de variação do controlador de T27.
WXB   = 0.6; % Penalização da Variável de Saída XB.
WXD   = 0.6; % Penalização da Variável de Saída XD.

% 2) Considerando otimização com 1 Pesos das Entradas e 1 Peso das
Saídas:
WDu = 0.1;
WDy = 0.6;

M   = 5; % Horizonte de Controle.
P   = 30; % Horizonte de Predição.

% Trajetória:
T   = 50;
sp  = [0.05 0;0 0.05;-0.05 0;0 -0.05];

%% Otimização:
wts = input('Pesos iguais ou diferentes (eq/d)?\n','s');

it = 200;

% Alocação de Memória:
WDT1D_v = zeros(it,1);
WDP19_v = zeros(it,1);
WDT27_v = zeros(it,1);
WXB_v   = zeros(it,1);
WXD_v   = zeros(it,1);
M_v     = zeros(it,1);
P_v     = zeros(it,1);
ISE_v   = zeros(it,1);

WDu_v = zeros(it,1);
WDy_v = zeros(it,1);
```

```

% loop:

for cont = 1:it

switch wts
    case 'd'
        [xo,Ot,nS]=SWARM('parameters',[WDT1D WDP19 WDT27 WXB WXD M
P],0,...
                        [1 1 1 1 1 1 1]',[0 0 0 0 0 1 7],[1 1 1 1 1 50
50]);

        WDT1D_v(cont,1) = xo(1);
        WDP19_v(cont,1) = xo(2);
        WDT27_v(cont,1) = xo(3);
        WXB_v(cont,1)   = xo(4);
        WXD_v(cont,1)   = xo(5);
        M_v(cont,1)     = xo(6);
        P_v(cont,1)     = xo(7);
        ISE_v(cont,1)   = Ot;

    case 'eq'
        [xo,Ot,nS]=SWARM('parameters',[WDu WDy M P],0,...
                        [1 1 1 1]',[0 0 1 7],[1 1 50 50]);

        WDu_v(cont,1) = xo(1);
        WDy_v(cont,1) = xo(2);
        M_v(cont,1)   = xo(3);
        P_v(cont,1)   = xo(4);
        ISE_v(cont,1) = Ot;

end

end

```

## A1.2 – Rotinas Auxiliares

### A1.2.1 – controlador.m:

```

global MPCobj
%% Modelo de Processo:

model;

%% Síntese do controlador:

Ts = 2;
MPCobj = mpc(G,Ts);

%% Unidades De Engenharia

MPCobj.MV(1).Units = '°C';
MPCobj.MV(2).Units = 'kPa';
MPCobj.MV(3).Units = '°C';

```

## A1.2.2 – model.m:

```
%% Modelo de Processo:
g11 = tf(7.4965e-4, [13.6210 1], 'iodelay', 12.1401);
g12 = tf(4.3116e-5, [2.3340 1]);
g13 = tf(-1.2320e-3, [0.1985 1]);
g21 = tf(2.0180e-3, [16.2931 1], 'iodelay', 8.4626);
g22 = tf(5.6120e-5, [0.7059 1]);
g23 = 0;

G = [g11 g12 g13; g21 g22 g23];

% VARIÁVEIS DE ENTRADA:

% T1D = Temperatura de Saída do Segundo Resfriador de Propano [°C]
% P19 = Pressão de Operação da Coluna [kPa]
% T27 = Temperatura do Estágio 27 da Coluna [°C]
G.inputname = {'T1D', 'P19', 'T27'};

% VARIÁVEIS DE SAÍDA:

% XB = Composição de Metano no Fundo
% XD = Composição de Etano no Topo
G.outputname = {'XB', 'XD'};
```

### A1.2.3 – parameters.m:

```
function J = parameters(param)

global T sp MPCobj wts

% Quantidade de Perturbações:
[m n] = size(sp);

switch wts

    case 'eq'
        MPCobj.W.ManipulatedVariablesRate(1) = param(1);
        MPCobj.W.ManipulatedVariablesRate(2) = param(1);
        MPCobj.W.ManipulatedVariablesRate(3) = param(1);
        MPCobj.W.OutputVariables(1) = param(2);
        MPCobj.W.OutputVariables(2) = param(2);
        MPCobj.P = param(4);
        MPCobj.controlhorizon = param(3);

    case 'd'
        MPCobj.W.ManipulatedVariablesRate(1) = param(1);
        MPCobj.W.ManipulatedVariablesRate(2) = param(2);
        MPCobj.W.ManipulatedVariablesRate(3) = param(3);
        MPCobj.W.OutputVariables(1) = param(4);
        MPCobj.W.OutputVariables(2) = param(5);
        MPCobj.P = param(7);
        MPCobj.controlhorizon = param(6);

end

% Função:
for i=1:m
    [y,t,u] = sim(MPCobj,T,sp(i,:));
    ise(i) = sumerro(y,u,sp(i,:));
end

J = sum(ise);
```

#### A1.2.4 – sumerro.m:

```
% Função para o cálculo do ISE

function J = sumerro(y,u,sp)

% Variável de Saída
y1 = y(:,1);
y2 = y(:,2);

sp1 = sp(1)*ones(length(y1),1);
sp2 = sp(2)*ones(length(y2),1);

% Movimentos de Controle:
u0 = [0 0 0];
du = zeros(length(u),3);
du(1,:) = u(1,:) - u0;
for k = 2:length(u)
    du(k,:) = u(k,:) - u(k-1,:);
end
du1 = du(:,1);
du2 = du(:,2);
du3 = du(:,3);

% Função
J = sum(10^2*[(sp1 - y1).^2 + (sp2 - y2).^2]) + ...
    (0.1^2)*sum(du1.^2) + sum(du2.^2) + (0.1^2)*sum(du3.^2);
```

## A1.2.4 – swarm.m:

```
function [xo,Ot,nS]=SWARM(S,x0,ip,isb,Lb,Ub,R,nger,npas,problem,c1,c2)
global pest
global kp1_vazao
global kp1_temp
global tau1
global tau2
global texp
global pexp
global mag_step_vazao
global mag_step_temp
global real_param
global f_vazao
global param

global wts

% Hard constrained global optimization using SWARM.
%
% [xo,Ot,nS]=swarm(S,x0,ip,isb,Lb,Ub,R,nger,npas,problem,c1,c2)
%
% S: objective function
% x0: initial point
% ip: (0): no plot (default), (>0) plot figure ip with pause, (<0)
plot figure ip
% isb: isb(i)=1 is i-th variable is bounded, otherwise isb(i)=0
(default = 0)
% Lb, Ub: lower and upper bound vectors (default = x0*(1+/-2) for
plot if isb(i)=0)
% R: axis vector of the hyperellipse centered in x0 (default =
max(0.1*abs(x0+~x0),1))
% nger: number of generations or flights (default = 50)
% npas: number of birds (default = 10)
% problem: (-1): minimum (default), (1): maximum
% c1: local search parameter (default = 1)
% c2: global search parameter (default = 1)
% xo: optimal point
% Ot: optimal values of S
% nS: number of objective function evaluations

% $Revision: 1.0 $ $Date: 2002/09/15 23:10:05 $
% Author: Esly Ferreira da Costa Junior (versao FORTRAN, 10/2001)
% MATLAB version: Argimiro R. Secchi (arge@enq.ufrgs.br)

if nargin < 2,
    error('swarm requires two input arguments');
end

x0=x0(:);
n=size(x0,1);

if nargin < 3 | isempty(ip),
    ip=0; %default plot
end
if nargin < 4 | isempty(isb),
    isb=zeros(n,1);
```

```

end
if nargin < 5 | isempty(Lb),
    Lb=-x0-~x0;
end
if nargin < 6 | isempty(Ub),
    Ub=2*x0+~x0;
end
if nargin < 7 | isempty(R),
    R=max(0.1*abs(x0+~x0),1);
end
if nargin < 8 | isempty(nger),
    nger=50;
end
if nargin < 9 | isempty(npas),
    npas=20;
end
if nargin < 10 | isempty(problem),
    problem=-1;
end
if nargin < 11 | isempty(c1),
    c1=1;
end
if nargin < 12 | isempty(c2),
    c2=1;
end

% Local variables:
% ipg = index of best position
% tw = inertial factor reduction rate
% p = matrix of best position of each bird
% v = bird velocity matrix
% x = bird location matrix
% w = inertial factor

                                % initialization

R=abs(R(:));
Lb=Lb(:);
Ub=Ub(:);
w=0.9;
tw=(w-4e-3)/nger;
Ot=feval(S,x0)*problem;
y(1)=Ot;
p=zeros(n,npas);
v=zeros(n,npas);
x=zeros(n,npas);

if ip && n == 2,
    figure(abs(ip));
    [X1,X2]=meshgrid(Lb(1):(Ub(1)-Lb(1))/20:Ub(1),Lb(2):(Ub(2)-
Lb(2))/20:Ub(2));
    [n1,n2]=size(X1);
    f=zeros(n1,n2);
    for i=1:n1,
        for j=1:n2,
            f(i,j)=feval(S,[X1(i,j);X2(i,j)]);
        end
    end
    mxf=max(max(f));
    mnf=min(min(f));
    df=mnf+(mxf-mnf)*(2.^(( [0:10]/10).^2)-1);
    [v1,h]=contour(X1,X2,f,df); hold on;

```



```

h1=plot(x0(1),x0(2),'ro');
legend(h1,'start point');

if ip > 0,
    disp('Pause: hit any key to continue...'); pause;
end
end

del=Ub-Lb;

for j=1:npas,
    x(:,j)=Lb+rand(n,1).*del;
    %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
    % BY JOAO CORREA LIMA NETO
    switch wts
        case 'd'
            x(6,j) = round(x(6,j));
            x(7,j) = round(x(7,j));
            if x(6,j) > x(7,j)
                x(6,j) = x(7,j)-1;
            end
        case 'eq'
            x(3,j) = round(x(3,j));
            x(4,j) = round(x(4,j));
            if x(3,j) > x(4,j)
                x(3,j) = x(4,j)-1;
            end
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
v(:,j)=0.1*rand(n,1).*del;
p(:,j)=x(:,j);
end

x(:,1)=x0;
p(:,1)=x0;
nS=1;
ipg=1;

for j=2:npas,
    y(j)=feval(S,x(:,j))*problem;
    nS=nS+1;

    if y(j) > Ot,
        Ot=y(j);
        ipg=j;
    end
end

if ip && n == 2,
    plot([x(1,ipg) x0(1)], [x(2,ipg) x0(2)], 'r');
    if ip > 0,
        disp('Pause: hit any key to continue...'); pause;
    end
end

xo=x(:,ipg);

for ig=1:nger,

```

```

rnd1=ones(n,1)*rand(1,npas);
rnd2=ones(n,1)*rand(1,npas);
v=w*v+c1*rnd1.*(p-x)+c2*rnd2.*(p(:,ipg)*ones(1,npas)-x);
x=x+v;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BY JOAO CORREA LIMA NETO
switch wts
case 'd'
    x(6,:) = round(x(6,:));
    x(7,:) = round(x(7,:));
    mp = find(x(6,*)>x(7,));
    pos = length(mp);
    for k = 1:pos
        x(6,mp(k)) = x(7,mp(k))-1;
    end
case 'eq'
    x(3,:) = round(x(3,:));
    x(4,:) = round(x(4,:));
    mp = find(x(3,*)>x(4,));
    pos = length(mp);
    for k = 1:pos
        x(3,mp(k)) = x(4,mp(k))-1;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%
for i=1:n,
    if isb(i),
        j=find(x(i,*) > Ub(i));
        if ~isempty(j),
            x(i,j)=Ub(i);
            v(i,j)=0;
        end

        j=find(x(i,*) < Lb(i));
        if ~isempty(j),
            x(i,j)=Lb(i);
            v(i,j)=0;
        end
    end
end

for j=1:npas,
    val=feval(S,x(:,j))*problem;
    nS=nS+1;

    if val > y(j),
        y(j)=val;
        p(:,j)=x(:,j);

        if val > Ot,
            Ot=val;
            ipg=j;
        end
    end
end
end

```

```

if ip && n == 2,
    plot([x(1,ipg) xo(1)], [x(2,ipg) xo(2)], 'r');
    if ip > 0,
        disp('Pause: hit any key to continue...'); pause;
    end
end

xo=x(:,ipg);

w=w-tw;
end

if ip && n == 2,
    h2=plot(xo(1),xo(2), 'r*');
    legend([h1,h2], 'start point', 'optimum');
end
param=x0;
Ot=Ot*problem;

real_param=abs(param);

```