

**UNIVERSIDADE FEDERAL DE SANTA CATARINA  
DEPARTAMENTO DE AUTOMAÇÃO E SISTEMAS**

Bryant Bruce Picon Yang

**CONTRIBUIÇÕES AOS ALGORITMOS PNMPC E DMC  
COM PASSO DE ESTABILIDADE**

Florianópolis

2016

Bryant Bruce Picon Yang

**CONTRIBUIÇÕES AOS ALGORITMOS PNMPC E DMC  
COM PASSO DE ESTABILIDADE**

Dissertação de mestrado submetida ao Programa de Pós-Graduação em Engenharia de Automação e Sistemas para a obtenção do Grau de Mestre em Engenharia de Automação e Sistemas.  
PGEAS-UFSC: Prof. Dr. Agostinho Plucenio

Florianópolis

2016

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

Yang, Bryant Bruce  
CONTRIBUIÇÕES AOS ALGORITMOS PNMPC E DMC COM PASSO DE  
ESTABILIDADE / Bryant Bruce Yang ; orientador, Agostinho  
Plucenio - Florianópolis, SC, 2016.  
146 p.

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

Inclui referências

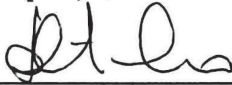
1. Engenharia de Automação e Sistemas. 2. Controle  
Preditivo. 3. Controle Não Linear. 4. PNMPC. 5. MPC com  
Passo de Estabilidade. I. Plucenio, Agostinho. II.  
Universidade Federal de Santa Catarina. Programa de Pós  
Graduação em Engenharia de Automação e Sistemas. III. Título.

Bryant Bruce Picon Yang

**CONTRIBUIÇÕES AOS ALGORITMOS PNMPC E DMC COM PASSO DE ESTABILIDADE**

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

Florianópolis, 09 de Dezembro de 2016.



---

Prof. Daniel Ferreira Coutinho, Dr.  
Coordenador do Curso  
DAS-UFSC



---

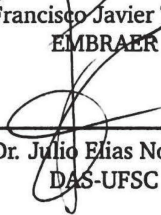
Prof. Dr. Agustinho Plucenio  
Orientador  
DAS-UFSC

**Banca Examinadora:**



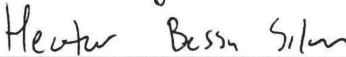
---

Prof. Dr. Francisco Javier Triveño Vargas  
EMBRAER



---

Prof. Dr. Julio Elias Normey Rico  
DAS-UFSC



---

Prof. Dr. Hector Bessa Silveira  
DAS-UFSC

Este trabalho é dedicado à minha família  
e a meus queridos pais e em especial a  
Mimi.



## AGRADECIMENTOS

Primeiramente, a Deus por ter me dado saúde e força para superar as dificuldades.

Aos membros da banca, pelas valiosas contribuições a este trabalho.

Ao meu orientador, Prof. Agustinho Plucenio que considero um excelente professor e orientador e, acima de tudo, um grande amigo, sempre auxiliando de uma forma bastante criativa nas incontáveis dúvidas que surgiram durante a realização deste trabalho.

Agradeço também ao Prof. Julio Normey pelos seus ensinamentos e por me dar a oportunidade e abrir as portas ao programa PRH-34 no curso de mestrado.

Agradeço, de coração, à toda minha família. Em especial, aos meus pais Ricardo e Shao Hwa, e meus irmãos Hans e Estiffany, pelo amor, incentivo e apoio incondicional.

Às meus amigos Josue, Patty, Alita e Irma, Henrique Farina, Mônica Dias, Mirna, Pablo, Eleonor e Andrew que sempre me incentivaram e me deram carinho para continuar seguindo em frente na constante busca pelo conhecimento.

À Carolina Vettorazo, pelas dicas e pela força nesses momentos de tensão.

Um agradecimento especial ao meu amigo Fabrício Saggin, que sempre está disposto a ajudar em tudo momento, pela revisão e contribuições aos capítulos deste trabalho.

A meu amigo que considero quase um irmão Leandro Morita e Eduarda, muito grato pelo apoio e pela amizade ao longo deste tempo.

A meus parceiros da Colômbia, Sergio, Deinis, Dayron, Andrea e Camilo, obrigado pela amizade.

À Miriam, minha amada, presente sempre no meu lado. Agradeço pelas palavras de força e pelos bons momentos que sempre passamos juntos.

Finalmente, gostaria de agradecer à UFSC-DAS pelo ensino de qualidade, ao PRH-34, ANP, sem os quais essa dissertação dificilmente poderia ter sido realizada. E a todos mais que eu não tenha citado nesta lista de agradecimentos, mas que de uma forma ou de outra contribuíram não apenas para a minha dissertação, mas também para eu ser quem eu sou.





Eu tentei 99 vezes e falhei, mas na centésima tentativa eu consegui, nunca desista de seus objetivos mesmo que esses pareçam impossíveis, a próxima tentativa pode ser a vitoriosa.

(Albert Einstein)



## RESUMO

Neste trabalho são abordados estudos de controle preditivo linear e não linear. Grande parte dos processos existentes na indústrias de petróleo e gás utilizam controladores preditivos. A crescente utilização dessas estratégias pode ser atribuída ao fato de elas facilitarem a declaração de problemas de controle relativamente complexos, como os casos de controle de processos multivariáveis, processos integradores, processos instáveis, processos não lineares e processos com atraso de transporte. A abordagem destes estudos é muito interessante para aplicações industriais em geral, onde a maioria dos processos dinâmicos exibem relações não lineares entre variáveis controladas e manipuladas. Dependendo da aplicação, o algoritmo MPC é muitas vezes escolhido entre um tipo linear ou não linear. Entre as famílias dos MPC lineares e não lineares escolhidos neste trabalho se tem o algoritmo DMC-Recursivo, e o MPC Prático para sistemas não lineares PNMPC, que tem apresentando bons resultados tanto no entorno acadêmico quanto no entorno industrial. O presente trabalho encontra-se dividido em duas partes. Na primeira parte são apresentadas diversos métodos de implementação para o algoritmo PNMPC, com o objetivo principal de medir o tempo de execução dos algoritmos implementados para um estudo de caso. Esta análise mostra que o algoritmo PNMPC pode ser adequadamente aplicado para tratar as não-linearidades em diferentes níveis. O desempenho e o custo computacional de seis implementações diferentes para este controlador são exibidos. Os ensaios de simulação da estratégia de controle em malha fechada de cada um dos controladores são comparados e analisados pela resposta do sistema, o tempo de execução do algoritmo e pelo índice de desempenho obtido pelos mesmos, sob perturbações externas e incertezas paramétricas. Na segunda parte, apresenta-se uma abordagem que é orientada ao controle de plantas monovariáveis e multivariáveis, com múltiplas incertezas no modelo incluindo: no ganho, nas constantes de tempo e nos atrasos de transporte, que representam todos os casos possíveis presentes nos processos industriais. Neste caso, é implementado o algoritmo MPC com passo de estabilidade, no qual a ação de controle é calculada em duas etapas, garantindo a estabilidade da planta devido à robustez do controlador, provocando uma melhoria na velocidade da rejeição das perturbações não modeladas. Os ensaios de simulação são comparados com o controlador DMC-Recursivo aplicando a filtragem no erro de predição.

**Palavras-chave:** Controle preditivo; Controle não-linear; PNMPC; Custo computacional; MPC com passo de estabilidade; Tanque cônico; Reator químico.

## ABSTRACT

Techniques of linear and non-linear predictive control are approached in this document. Much of the existing processes in the oil and gas industries use predictive controllers. The increasing use of these strategies can be attributed to the fact that they facilitate the declaration of relatively complex control problems, such as cases like multivariable processes control, integrative processes, unstable processes, non-linear processes and processes with time delays. The approaches of these techniques are very interesting for industrial applications in general because the process in the industry has a non-linear dynamic between the controlled and the manipulated variables. Depending on the application, the MPC algorithm is often chosen from a linear or non-linear type. Among linear and non-linear MPC families chosen in this document has the Recursive DMC algorithm and Practical Non-linear MPC (PNMPC), which has shown good results in the academic and in the industrial environment. This document is divided into two parts. In the first part, several PNMPC algorithm implementation methods are presented, with the main objective of measuring the execution time of each algorithm implemented in a case study. This review shows that the PNMPC algorithm can be applied to treat different levels of nonlinearities. The performance and computational cost of six different implementations for this controller are shown. Simulation tests in the closed-loop control strategy with each controller are compared and analyzed by the system response, the execution time of the algorithm and by the performance index obtained by them under external disturbances and parametric uncertainties. The second part presents an approach that is oriented to the control of mono variables and multivariable plants with multiple uncertainties in the model, including: gains, time constants and time delays, which represent all possible cases in industrial processes. The implemented MPC algorithm with stability step, in which the control action is calculated in two steps guaranteeing the stability of the plant due to the robustness of the controller inducing an improvement in the speed of the rejection of the disturbances not modeled. Simulation tests are compared with the Recursive DMC controller applying the filtering in the prediction error.

**Keywords:** Predictive control. Non-linear control. PNMPC. Computational cost. MPC with stability step. Conical tank. Chemical reactor.



## LISTA DE FIGURAS

Figura 1	Estrutura do MPC.....	36
Figura 2	Resposta livre e resposta forçada.....	37
Figura 3	Trajetória da referência.....	39
Figura 4	Resposta au degrau.....	42
Figura 5	Diagrama de Blocos para o Tratamento do Erro de Predição PNMPC.....	56
Figura 6	Interpretação gráfica $\bar{\mathbf{G}}_{PNMPC}$ calculada em dois passos.....	61
Figura 7	Fluxograma do algoritmo PNMPC da primeira versão.....	62
Figura 8	Interpretação gráfica $\mathbf{G}_{PNMPC}$ calculada em um único passo.....	63
Figura 9	Fluxograma do algoritmo PNMPC da segunda versão.....	64
Figura 10	Faixa de erro de predição admissível.....	69
Figura 11	Tanque cônico.....	70
Figura 12	Simulação da primeira versão do PNMPC.....	75
Figura 13	Simulação da segunda versão do PNMPC.....	77
Figura 14	Matriz $\mathbf{G}_{PNMPC}$ da iteração 25.....	79
Figura 15	Matriz $\mathbf{G}_{PNMPC}$ invariante no tempo.....	80
Figura 16	Simulação da terceira versão do PNMPC.....	81
Figura 17	Simulação da quarta versão do PNMPC.....	83
Figura 18	Simulação da quinta versão do PNMPC.....	85
Figura 19	Simulação da sexta versão do PNMPC.....	87
Figura 20	Comparações das versões do PNMPC: saídas, ações de controle, perturbações e sinais dos erros.....	89
Figura 21	Preditor de Smith Filtrado.....	94
Figura 22	Formas de implementar o vetor $E_L$ .....	98
Figura 23	Esquema do tanque de aquecimento de água.....	104
Figura 24	Simulação do sistema sem atraso e sem incertezas no modelo.....	105
Figura 25	Simulação do sistema com atraso e sem incertezas no modelo.....	105
Figura 26	Saídas preditas com o atraso e sem o atraso.....	106
Figura 27	Diferença entre as saídas predita com o atraso e sem o	

atraso.....	106
Figura 28 DMC-Recursivo com filtro no erro de predição aplicando uma incerteza de 6 min ( $d_n = 30$ ).....	107
Figura 29 DMC-Recursivo com passo de estabilidade aplicando uma incerteza de 6 min ( $d_n = 30$ ).....	108
Figura 30 Fatores de correção do DMC-Recursivo com passo de estabilidade, DMC-Recursivo com filtro no erro de predição e <i>diff</i> .....	109
Figura 31 Esquema do Reator de tanque agitado.....	111
Figura 32 Resposta ao degrau das funções de transferência.....	113
Figura 33 Incertezas ineridas no atraso de transporte e no ganho.....	114
Figura 34 Resultados de simulação.....	115
Figura 35 Comparação dos erros $E_f$ frente a $E_L$ .....	116
Figura 36 Resultados de simulação.....	117
Figura 37 Resultados de simulação.....	117
Figura 38 Resultados de simulação.....	117
Figura 39 Resultados de simulação.....	118



## LISTA DE TABELAS

Tabela 1	Parâmetros do tanque cônico. ....	71
Tabela 2	Parâmetros de sintonia do controlador PNMPC .....	73
Tabela 3	Tempo de execução do algoritmo para a primeira versão do PNMPC. ....	76
Tabela 4	Tempo de execução do algoritmo para a segunda versão do PNMPC. ....	78
Tabela 5	Tempo de execução do algoritmo para a terceira versão do PNMPC .....	82
Tabela 6	Tempo de execução do algoritmo para a quarta versão do PNMPC. ....	84
Tabela 7	Tempo de execução do algoritmo para a quinta versão do PNMPC. ....	86
Tabela 8	Tempo de execução do algoritmo para a sexta versão do PNMPC. ....	88
Tabela 9	Tabela comparativa dos tempos de execução para referências superiores a 1 m. ....	90
Tabela 10	Tabela comparativa dos tempos de execução para referências inferiores a 1 m. ....	90
Tabela 11	Índices de desempenho do controlador das seis versões do PNMPC para referências superiores a 1 m. ....	91
Tabela 12	Parâmetros de sintonia do controlador DMC recursivo com filtragem no erro de predição .....	110
Tabela 13	Parâmetros de sintonia do controlador DMC-Recursivo com passo de estabilidade. ....	111
Tabela 14	Índice de desempenho do controlador. ....	118



## LISTA DE ABREVIATURAS E SIGLAS

MPC	Controle Preditivo Baseado em Modelo (do inglês, <i>Model Based Predictive Control</i> ) . . . . .	25
DMC	Matriz Dinâmica de Controle (do inglês, <i>Dynamical Matrix Control</i> ) . . . . .	25
GPC	Controle Preditivo Generalizado (do inglês, <i>Generalized Predictive Control</i> ) . . . . .	25
NMPC	Modelo de Controle Preditivo Não Linear (do inglês, <i>Non-linear Model Predictive Control</i> ) . . . . .	26
PNMPC	Modelo de Controle Preditivo Prático para sistemas Não Lineares (do inglês, <i>Non-linear Model Predictive Control</i> ) . . . . .	27
ARIMA	Modelo Auto-Regressivo integrado e de Média Móvel Integrada (do inglês, <i>Auto-Regressive and Integrated Moving Average</i> ) . . . . .	35
SISO	Uma Entrada e Uma Saída (do inglês, <i>Single-Input Single-Output</i> ) . . . . .	45
MIMO	Múltiplas Entradas e Múltiplas Saídas (do inglês, <i>Multiple-Input Multiple-Output</i> ) . . . . .	45
QP	Programação Quadrática (do inglês, <i>Quadratic Programming</i> ) . . . . .	57
MSE	Erro Quadrático Médio (do inglês, <i>Mean Squared Error</i> ) . . . . .	90
IAE	Integral do Erro Absoluto (do inglês, <i>Integral of the Absolute Magnitude of the Error</i> ) . . . . .	90



## LISTA DE SÍMBOLOS

$g_i$	Coeficiente da resposta ao degrau .....	33
$\Delta \mathbf{u}$	Vetor da variação de controle .....	33
$p$	Horizonte de predição .....	33
$A(z^{-1})$	Denominador da função de transferência .....	34
$B(z^{-1})$	Numerador da função de transferência .....	34
$y(t)$	Saída do processo .....	34
$u(t)$	Sinal do controle .....	34
$\tilde{y}(t+k)$	Saída predida .....	34
$J$	Função custo .....	37
$d$	Tempo morto ou atraso de transporte .....	37
$m_h$	Horizonte de controle .....	38
$\gamma$	Ponderação no seguimento de referência .....	38
$\lambda$	Ponderação no controle .....	38
$w(t+k)$	Trajetória das referências futuras .....	38
$\alpha$	Filtro de referência .....	38
<b>G</b>	Matriz dinâmica do sistema .....	42
<b>f</b>	Vetor da resposta livre do sistema .....	42
$N$	Horizonte de modelo .....	43
$e(t)$	Erro de predição ou erro verdadeiro .....	45
<b>G</b> <sub>P<sub>N</sub>M<sub>P</sub>C</sub>	Matriz dinâmica ou matriz Jacobiana .....	49
<b>T</b> <sub>I</sub>	Matriz Toeplitz integradora .....	60
$Q_{in}$	Vazão de entrada .....	70
$Q_o$	Vazão de saída .....	70
$R$	raio máximo do tanque cônico .....	70
$r$	raio interno (variável) do tanque cônico .....	70
$H$	Altura máxima do tanque cônico .....	70
$h$	Altura interna (variável) do tanque cônico .....	70
$C_v$	Coeficiente da capacidade do fluxo da válvula de saída ..	70
$\epsilon$	Variação de controle .....	73
<b>E</b> <sub>f</sub>	Vetor do erro de predição filtrado .....	95
<b>E</b> <sub>L</sub>	Vetor do erro passo de estabilidade .....	97
$\beta$	Filtro no erro de predição .....	110



## SUMÁRIO

<b>1</b>	<b>INTRODUÇÃO</b> .....	25
1.1	OBJETIVOS .....	28
1.1.1	Objetivos gerais .....	28
1.1.2	Objetivos específicos .....	28
1.2	CONTRIBUIÇÃO CIENTÍFICA .....	29
1.3	ORGANIZAÇÃO DO TRABALHO .....	29
<b>2</b>	<b>CONTROLE PREDITIVO BASEADO EM MO- DELO MPC</b> .....	31
2.1	CONCEITOS BÁSICOS DO MPC .....	32
2.1.1	Resposta livre e forçada .....	36
2.1.2	Função objetivo .....	37
2.1.3	Trajetória das referências .....	38
2.1.4	Restrições .....	38
2.1.5	Obtenção da Lei de Controle .....	39
2.2	ALGORITMO MPC UTILIZADO NA INDÚSTRIA ....	40
2.2.1	Controle por Matriz Dinâmica-DMC .....	40
2.2.2	DMC Original .....	40
2.2.3	DMC-Recursivo .....	43
2.3	CONCLUSÃO .....	45
<b>3</b>	<b>MPC PRÁTICO PARA SISTEMAS NÃO LINE- ARES - PNMPC</b> .....	47
3.1	OBTENÇÃO NUMÉRICA DAS PREDIÇÕES .....	54
3.2	TRATAMENTO DO ERRO DE PREDIÇÃO E PERTUR- BAÇÃO .....	55
3.3	FUNÇÃO CUSTO .....	57
3.4	CONCLUSÃO .....	58
<b>4</b>	<b>IMPLEMENTAÇÃO DO PNMPC: ALTERNA- TIVAS</b> .....	59
4.1	VERSÕES DA TÉCNICA MPC PRÁTICO PARA SIS- TEMAS NÃO LINEARES (PNMPC) .....	59
4.1.1	Primeira versão: PNMPC original completo com cálculo de $\Delta u_{ini}$ .....	60
4.1.2	Segunda versão: PNMPC sem o cálculo de $\Delta u_{ini}$ ..	63
4.1.3	Terceira versão: PNMPC obtendo só a primeira coluna da matriz $G_{PNMPC}$ .....	65

4.1.4	Quarta versão: PNMPC obtendo só a primeira coluna da matriz $G_{PNMPC}$ recalculada de tempos em tempos. ....	66
4.1.5	Quinta versão: PNMPC calculando a resposta livre de tempos em tempos, utilizando o DMC-Recursivo. ....	67
4.1.6	Sexta versão: PNMPC com reavaliação da matriz $G_{PNMPC}$ e a resposta livre (DMC-Recursivo), conforme o erro de predição. ....	69
4.2	ESTUDO DE CASO: TANQUE CÔNICO .....	69
4.3	RESULTADOS .....	72
4.3.1	Primeira versão: PNMPC original completo com $\Delta u_{ini}$ .....	74
4.3.2	Segunda versão: PNMPC sem o cálculo de $\Delta u_{ini}$ ..	76
4.3.3	Terceira versão: PNMPC obtendo só a primeira coluna da matriz $G_{PNMPC}$ . ....	78
4.3.4	Quarta versão: PNMPC obtendo só a primeira coluna da matriz $G_{PNMPC}$ recalculada de tempos em tempos. ....	82
4.3.5	Quinta versão: PNMPC calculando a resposta livre de tempos em tempos, utilizando o DMC-Recursivo. ....	84
4.3.6	Sexta versão: PNMPC com reavaliação da matriz $G_{PNMPC}$ e a resposta livre (DMC-Recursivo), conforme o erro de predição. ....	86
4.4	CONCLUSÃO .....	88
5	<b>CONTROLE PREDITIVO BASEADO EM MODELO COM PASSO DE ESTABILIDADE.....</b>	93
5.1	PASSO DE ESTABILIDADE PARA COMPENSAÇÃO DE ERROS NO MODELO DE PREDIÇÃO .....	97
5.2	ESTUDO DE CASO.....	110
5.3	CONCLUSÕES .....	118
6	<b>CONSIDERAÇÕES FINAIS.....</b>	121
	<b>REFERÊNCIAS .....</b>	125
	<b>APÊNDICE A – Apêndice.....</b>	131



## Capítulo 1

---

# INTRODUÇÃO

---

O Controle preditivo baseado em modelo (*Model Predictive Control*, MPC ou MBPC) é um conjunto de métodos de controle que foram desenvolvidos considerando um modelo capaz de representar a dinâmica do processo, para prever e obter uma ação de controle futura que minimize uma função objetivo. No entanto, o MPC calcula a lei de controle em cada interação com ajuda de um algoritmo de otimização *on-line*, sendo o modelo o elemento mais importante dentro dos controladores preditivos. Algumas estratégias de controle mais avançadas podem falhar se o modelo não for o adequado (CAMACHO; BORDONS, 1995; MACIEJOWSKI, 2002).

O MPC é uma das poucas estratégias que permite o controle dos sistemas com restrições com um critério ótimo, garantindo a estabilidade e convergência para o ponto de equilíbrio (MAYNE, 2001).

A maioria dos processos petroquímicos utilizam controladores preditivos. De fato, o MPC nasceu dentro das indústrias petroquímicas na década dos 70. Atualmente sua aplicação vem sendo cada vez mais difundida em outros setores (BORDONS, 2000; MORARI; LEE, 1999).

As estratégias mais utilizadas e conhecidas no meio acadêmico e no meio industrial são: o DMC (*Dynamical Matrix Control*), apresentado em Cutler e Remarker (1980); e o GPC (*Generalized Predictive Control*), apresentado em Clarke e Mohtadi (1987a), aplicados ao controle preditivo de processos lineares (CAMACHO; BORDONS, 2007). No entanto, como nem todos os processos são lineares, na prática se faz a linearização do modelo do processo em torno de pontos de operação. Quando os modelos lineares não são suficientemente precisos para representar as não-linearidades do processo real, várias abordagens podem ser utilizadas (FINDEISEN; ALLGOWER, 2002; MORARI; LEE, 1999).

Existem dois comportamentos comumente encontrados nos processos industriais: um de eles é a não linearidade nos processos e a outra são os atrasos de transporte. Os modelos não lineares são geralmente usados para descrever a dinâmica do processo quando os modelos lineares não geram bons resultados (CAMACHO; BORDONS, 1995). Existem diversas técnicas empregadas para determinar o modelo de predição, como ser: modelos fenomenológicos (POMAR; NORMEY; CAMPONOGARA, 2005), modelos utilizando redes neurais (TANG; KEYSER, 1994), modelos empíricos (NELLES, 2001) ou com lógica nebulosa (SKRJANC; MATKO, 1994).

O controle preditivo não linear baseado em modelo (*Non-linear Model Predictive Control*, NMPC) é uma estratégia de controle relativamente nova que vem apresentando bons resultados, pelo que se torna uma alternativa para contornar processos reais (NORMEY; LIMA; SANTOS, 2015).

A aplicação de MPCs lineares nos processos que apresentam não linearidades fortes na sua dinâmica podem não trazer bons resultados, para isso, uma opção é a aplicação de controladores não lineares. A principal vantagem da aplicação do NMPC é o fato que este considera a dinâmica não linear do processo. Por outro lado, a maior desvantagem está na obtenção de um bom modelo não linear a partir de dados experimentais e no custo computacional que representa o cálculo da ação de controle. Num trabalho recente, Lucia, Rumschinski e Krener (2015) propõe técnicas de MPCs não lineares baseada em otimização min-max, sugerindo uma forma de diminuir o custo computacional.

Com o decorrer do tempo, diversas alternativas abordam problemas para contornar e controlar sistemas não lineares utilizando a estratégia MPC, entre elas estão: os modelos lineares dinâmicos e não lineares estáticos, modelos locais (aproximação no ponto de trabalho) e modelos combinados linear não-linear para predições (PLUCENIO, 2010). Evidentemente, as vantagens de uma ou outra alternativa estão diretamente relacionadas ao compromisso, desempenho e custo computacional do controlador (CANNON, 2004).

Entre os modelos lineares dinâmicos e não lineares estáticos são encontrados os trabalhos (GRUBER; BORDONS, 2007; BLOEMEN; BOOM; VERBRUGGEN, 2000; SANTOS, 2005), que utilizam séries de Volterra ou modelos de Hammerstein para MPCs não lineares.

Em Bravo e Normey (2009), apresenta-se uma abordagem para modelos locais (aproximação no ponto de trabalho), o que equivale a um MPC linear adaptativo.

Finalmente, para os modelos combinados que possuem parte da

predição linear e a outra parte não-linear, podem-se citar algumas estratégias encontradas na literatura, como o algoritmo NEPSAC (*Non-linear Extended Predictive Self Adaptive Control*), desenvolvido por Keyser (2007), e o algoritmo MPC Prático para Sistemas Não Lineares PNMPC (*Practical Non-linear Predictive Control*), desenvolvido por Plucenio (2010).

Dentre as técnicas de controle preditivo que tratam processos não linear, tem-se a técnica PNMPC vem apresentando bom desempenho e vasta possibilidade de aplicações ao longo deste tempo (ANDRADE et al., 2012; CASTILLA et al., 2014; THOME, 2013). No entanto, estas estratégias podem demandar um alto desempenho e sobretudo um elevado custo computacional, mesmo assim fornecendo grandes vantagens quanto comparados com as estratégias dos MPCs lineares.

Por outro lado, como já foi mencionado, um outro fenômeno comum dos processos industriais, principalmente nos processos da indústria de petróleo e gás, é o atraso de transporte. Na prática, tem-se muitos processos industriais onde o atraso é causado pelo transporte de material, energias ou pelo acúmulo de informação no sistema. Nesse caso, o sistema não responde imediatamente. Ele leva algum tempo para exibir em sua saída os efeitos de uma entrada (NORMEY, 2001).

Alguns trabalhos apresentam estratégias interessantes para contornar o problema do atraso de transporte e obter melhor desempenho nos controladores. A maioria dessas estratégias é baseadas nas ideias do preditor de Smith (SMITH, 1957).

Erros no atraso de transporte no modelo de predição podem levar o sistema de controle à instabilidade. Em Lima, Normey e Plucenio (2014), a utilização de filtros aplicados ao erro de predição é apresentada como solução no contexto de MPC. Esse conceito foi desenvolvido a partir das ideias do Preditor de Smith Filtrado, apresentado em Normey e Camacho (2007).

Em geral, o filtro não afeta a resposta do sistema quando o modelo é perfeito, mas sim quando existe erro no atraso de transporte, no ganho e nas constantes de tempo entre a planta real e o modelo de predição. Além disso, filtragem no sistema pode resultar numa resposta mais lenta às perturbações inseridas no sistema.

Em Plucenio et al. (2014), apresenta-se uma nova abordagem para o tratamento de erros de predição em plantas estáveis em malha aberta, denominada MPC com passo de estabilidade. Nessa proposta, nenhum filtro é utilizado. Em vez disso, o algoritmo MPC com passo de estabilidade propõe tratar o problema em duas etapas: uma primeira etapa onde se estabiliza o sistema utilizando o segundo método

de estabilidade de Lyapunov (MONTEIRO, 2006) onde se deriva um sinal que somado à predição fornece estabilidade, bom desempenho do controlador frente a erros de modelo, principalmente erros no atraso de transporte, fornecendo rapidez na rejeição às perturbações não modeladas inseridas no sistema. Na segunda etapa, calcula-se a ação de controle.

## 1.1 OBJETIVOS

### 1.1.1 Objetivos gerais

Este trabalho visa propor diversas alternativas de implementação da técnica PNMPC, considerando diferentes aproximações para o cálculo das predições de um dado processo não linear, avaliando os tempos de execução e o desempenhos das diferentes implementações dos controladores PNMPC. Também é objetivo desse trabalho implementar e avaliar o algoritmo MPC com passo de estabilidade implementado no controlador DMC-Recursivo, considerando incertezas no modelo de predição.

### 1.1.2 Objetivos específicos

Para conseguir alcançar os objetivos gerais, os seguintes objetivos específicos foram estabelecidos:

- implementar o controlador com a técnica PNMPC proposta em Plucenio (2010) e avaliar o tempo de execução deste;
- modificar o PNMPC e implementar diversos algoritmos que permitam tratar processos com diversos graus de não linearidade com a finalidade de diminuir o custo computacional;
- comparar o desempenho dos controladores PNMPC implementados;
- implementar o MPC com passo de estabilidade a ser comparando com a técnica de filtragem no erro de predição e avaliar o desempenho dos controladores.

## 1.2 CONTRIBUIÇÃO CIENTÍFICA

O presente trabalho apresenta contribuições aos algoritmos de controle preditivos.

Os principais resultados desta dissertação foram condensados nos artigos: *Practical Non-linear Model Predictive Control PNMPC: Algorithm implementations*, apresentado no XXI Congresso Brasileiro de Automática (CBA2016), realizado de 3 a 7 de outubro de 2016, em Vitória, Espírito Santo (PICON; PLUCENIO, 2016b); e “DMC de forma recursiva aplicando o passo de estabilidade” apresentado no 12<sup>th</sup> *IE-EE/IAS International Conference on Industry Applications* (INDUSCON 2016) realizado de 20 a 23 de novembro de 2016, em Curitiba, Paraná (PICON; PLUCENIO, 2016a).

## 1.3 ORGANIZAÇÃO DO TRABALHO

Os próximos capítulos desta dissertação estão organizados da seguinte maneira:

- o Capítulo 2 traz ao leitor a fundamentação teórica e os conceitos básicos necessários para o desenvolvimento deste trabalho. Os conceitos principais relacionados às estratégias de controle preditivo baseado em modelo MPC são apresentados;
- o Capítulo 3 apresenta a descrição da fundamentação teórica da estratégia PNMPC utilizada para o desenvolvimento deste trabalho;
- no Capítulo 4 são apresentadas diferentes maneiras de implementar o algoritmo PNMPC. Mostra-se que o algoritmo PNMPC pode ser escrito para lidar com não-linearidades em diferentes níveis. São implementados 6 algoritmos diferentes das versões do controlador PNMPC, que utilizam simplificações da técnica original completa de forma a reduzir o tempo de processamento. Nesse capítulo, também é descrita a implementação e a fundamentação matemática com os algoritmos das alternativas das variações da técnica PNMPC para cada versão. Depois é apresentado o estudo de caso, aplicando as versões do PNMPC e os resultados de simulação, comparações dos tempos de execução e desempenho dos controladores;
- no Capítulo 5 é apresentado o MPC com passo de estabilidade,

descrevendo a implementação e a fundamentação matemática. Depois, apresentam-se os resultados de simulação e as comparações dos diferentes controladores aplicados ao estudo de caso, avaliando o desempenho dos controladores;

- no Capítulo 6, são realizadas as considerações finais desta dissertação e possibilidades de trabalhos futuros são indicadas.

## Capítulo 2

---

# CONTROLE PREDITIVO BASEADO EM MODELO MPC

---

A técnica de Controle Preditivo Baseado em Modelo é uma das técnicas de controle moderno mais avançada. Em sua forma mais geral, aceita qualquer tipo de modelo, funções objetivo ou restrições (BORDONS, 2000).

Atualmente, é utilizada em processos industriais que requerem controlar processos multivariáveis ou monovariáveis relativamente complexos, processos lineares ou não lineares, processos que tenham características de atraso de transporte ou ainda processos instáveis (CAMACHO; BORDONS, 2007).

Além da sua ampla gama de aplicação, um outro motivo do êxito desta técnica é a maneira geral na qual os problemas de controle são formulados no domínio do tempo, permitindo que os algoritmos sejam muito mais simples de compreender.

A maior aceitação dos MPCs, ocorre nas indústrias de petróleo, gás, unidades de refino e petroquímicas. Atualmente, também existe uma tendência de incrementar a aplicação de controle MPC nos processos de tratamento primário de produção em plataformas de produção de petróleo *offshore* (PLUCENIO et al., 2007a). O controle preditivo constitui um amplo campo de métodos de controle que foram desenvolvidos com certas características comuns e que integra diversos âmbitos como: o controle ótimo, controle de processos com tempos mortos, controle multivariável, controle estocástico e o controle com restrições (CAMACHO; BORDONS, 2007). As principais características em toda a família dos controladores preditivos são: (BORDONS, 2000):

- o uso explícito de um modelo para prever a saída do processo em instantes de tempo futuros (horizonte);

- cálculos dos sinais de controle, minimizando uma determinada função objetivo;
- estratégia de horizonte deslizante, de forma que, em cada instante de tempo, o horizonte vai se deslocando para o futuro até eliminar o resto dos valores, repetindo o cálculo em cada instante de amostragem.

Este tipo de controle permite manipular as variáveis para estabelecer uma trajetória desejada no futuro, como também as variáveis que estão sendo controladas. Uma vez manipuladas as variáveis, procede-se com o cálculo, otimizando-as com a função objetivo e seguindo os métodos de horizonte móvel.

Finalmente, pode-se ainda destacar como vantagem da estratégia MPC a possibilidade de formular controladores para sistemas complexos e com restrições, tendo em consideração o custo computacional e a sintonia do controlador (LAWRYNCZUK, 2014).

## 2.1 CONCEITOS BÁSICOS DO MPC

As diferenças entre os diversos algoritmos existentes devem-se basicamente à forma de escolher os modelos para o processo e as perturbações, ao tipo de função objetivo, ao procedimento para manipular as restrições e ao cálculo do sinal de controle (PLUCENIO *et al.*, 2007a).

O modelo de predição é o elemento mais importante dentro do projeto do controlador. Ele deve ser capaz de representar adequadamente a dinâmica do processo para permitir o cálculo das predições da saída, deve ser intuitivo e ao mesmo tempo deve permitir uma análise teórica do sistema.

Em geral, o modelo utilizado para o projeto do controlador MPC é separado em duas partes: modelo do processo propriamente dito e modelo das perturbações, ambos necessários para o cálculo das predições (PLUCENIO *et al.*, 2007a).

Os modelos dos processos mais usados são:

- a resposta do sinal de saída ao impulso ou ao degrau dos modelos dos processos em malha aberta. Ela está relacionada com a entrada pelas seguintes equações:



- para a resposta ao impulso:

$$y(t) = \sum_{i=1}^{\infty} g_i u(t - i) \quad (2.1)$$

- para a resposta ao degrau:<sup>1</sup>

$$y(t + k | t) = \sum_{i=1}^{\infty} g_i \Delta u(t + k - i | t) \quad (2.2)$$

Sendo que os valores de  $g_i$  são obtidos ao submeter o processo a um impulso no caso da resposta ao impulso ou um degrau. No caso de ser uma resposta ao degrau, na qual a ação de controle é dada por:  $\Delta \mathbf{u}(t) = u(t) - u(t - 1)$ . A soma é truncada até os primeiros  $p$  elementos, sendo  $p$  o horizonte de predição, no qual só podem ser representados processos estáveis e sem integrador, conforme as equações (2.1) e (2.2):

- para a resposta ao impulso:

$$y(t) = \sum_{i=1}^p g_i \Delta u(t - i) \quad (2.3)$$

- para a resposta ao degrau:

$$y(t + k | t) = \sum_{i=1}^p g_i \Delta u(t + k - i | t) \quad (2.4)$$

As vantagens de se utilizar a resposta impulsiva ou ao degrau é que:

- são intuitivos e podem ser usados em processos multivariáveis sem acrescentar complexidade;
- quando os modelos são identificados não é preciso ter um conhecimento prévio do processo;
- descrevem de maneira simples efeitos complexos da dinâmica do processo, como atrasos e comportamentos de fase não mínima.

---

<sup>1</sup>A notação indica o valor da variável no instante  $(t + k)$  calculado no instante  $t$

Por outro lado apresentam alguns inconvenientes:

- não podem ser usados com plantas instáveis;
  - necessitam utilizar um grande número de parâmetros para descrever o modelo, sobretudo se o processo tiver um atraso grande.
- modelos em função de transferência.  
Utilizam o conceito de função de transferência  $G = B/A$ , representada da seguinte forma no domínio do tempo:

$$A(z^{-1})y(t) = B(z^{-1})u(t)$$

sendo:

$A(z^{-1})$	denominador da função de transferência;
$B(z^{-1})$	numerador da função de transferência;
$y(t)$	saída do processo;
$u(t)$	sinal do controle.

em que a saída predita é:

$$\tilde{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})}u(t+k|t)$$

A principal vantagem deste tipo de modelos é que pode ser usado para plantas instáveis e que precisa, em geral, de poucos parâmetros para descrever o comportamento do sistema (o atraso, por exemplo, pode ser descrito apenas com um parâmetro). Porém, o principal inconveniente é a necessidade de conhecer a priori a ordem dos polinômios do numerador e denominador quando o modelo deve ser identificado a partir de dados experimentais (PLUCENIO et al., 2007a).

- modelos em espaços de estado, tem a seguinte representação:

$$\begin{aligned}x(t) &= Ax(t-1) + Bu(t-1) \\y(t) &= Cx(t)\end{aligned}$$

onde  $x$  é o estado e  $A, B$  e  $C$  são as matrizes de entrada e saída do sistema respectivamente. A saída predita é calculada pela

seguinte equação:

$$\tilde{y}(t+k|t) = C\tilde{x}(t+k|t) = C \left[ A^k x(t) + \sum_{i=1}^k A^{i-1} B u(t+k-i|t) \right] \quad (2.5)$$

A vantagem deste tipo de modelo é que pode ser usado diretamente para processos multivariáveis lineares. Como inconvenientes, cabe mencionar que em geral os estados não tem significado físico e que na maioria das vezes é necessário o uso de observadores, aumentando assim a complexidade de cálculo do controle (PLUCENIO et al., 2007a).

A escolha de um modelo para as perturbações é tão importante quanto a do processo.

O modelo mais utilizado para a descrição de perturbações determinísticas e estocásticas é o Modelo Auto-Regressivo integrado e de Média Móvel Integrada (*Auto-Regressive and Integrated Moving Average*, ARIMA) (CAMACHO; BORDONS, 2007). Neste tipo de modelo, a diferença entre a saída medida e a calculada pelo modelo é dada pela equação:

$$n(t) = \frac{C(z^{-1})e(t)}{D(z^{-1})} \quad (2.6)$$

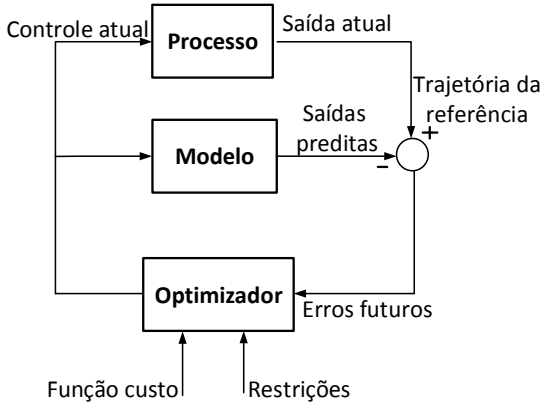
no qual o polinômio  $D(z^{-1})$  inclui explicitamente  $\Delta = 1 - z^{-1}$ , e  $e(t)$  é um ruído de média zero e, geralmente, o polinômio  $C(z^{-1})$  é considerado igual a um. Assim,  $n(t)$  representa a diferença entre a saída do modelo e o processo.

Este modelo permite representar mudanças aleatórias, “*off-sets*” e outros fenômenos normalmente encontrados nos meios industriais. Esse modelo é usado diretamente no GPC proposto por (CLARKE; MOHTADI, 1987a), e tem recebido bastante atenção no meio acadêmico e em diversas aplicações industriais (CAMACHO; BORDONS, 1995) e (CLARKE; MOHTADI, 1987b).

Na Figura 1 mostra-se a estrutura geral de um MPC, onde pode ser observado o funcionamento do algoritmo. Para prever as saídas futuras do processo, é usado o modelo da planta, baseando-se nos sinais de controle futuros. Esses sinais são calculados pelo otimizador tendo em consideração a função custo e as restrições. Pelo que o modelo cumpre um papel importante no controlador MPC, ele deve conseguir representar a dinâmica do processo. O otimizador é outra parte fundamental da estrutura mostrada na Figura 1, já que proporciona as ações de controle do sistema. Se a função custo é quadrática, pode-se obter

o mínimo com uma função explícita entre as entradas, saídas passadas e a trajetória de referência. Quando existem restrições a solução é calculada por métodos numéricos.

Figura 1 – Estrutura do MPC.



Fonte: (CAMACHO; BORDONS, 2007).

### 2.1.1 Resposta livre e forçada

Os conceitos de resposta livre e resposta forçada, são utilizados nos controladores MPC. A idéia é representar as ações de controle como uma soma de dois sinais, conforme equação (BORDONS, 2000):

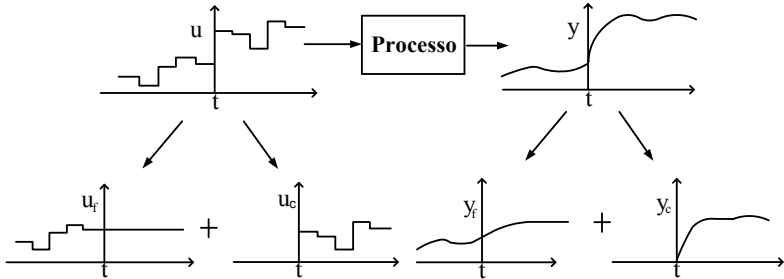
$$u(t) = u_f(t) + u_c(t) \quad (2.7)$$

onde  $u_f(t)$  são as entradas passadas (anteriores ao instante  $t$ ) e  $u_c(t)$  é o sinal dos controles futuros. Para o cálculo da ação de controle, o sinal  $u_f(t)$  é mantido constante e igual ao último valor da variável manipulada; o sinal  $u_c(t)$  é considerado 0 no passado. A saída predita é separada em duas partes, como se mostra na Figura 2.

A resposta livre  $y_f(t)$  corresponde à predição da saída quando a variável manipulada é  $u_f(t)$ . A resposta forçada  $y_c(t)$  corresponde à

predição da saída quando o sinal de controle é  $u_c(t)$ .

Figura 2 – Resposta livre e resposta forçada.



Fonte: (CAMACHO; BORDONS, 2007).

A resposta livre é a evolução do processo devido a seu estado atual, incluindo o efeito das ações passadas. A resposta forçada é a resposta do processo às ações de controle futuras.

### 2.1.2 Função objetivo

Cada algoritmo MPC propõe distintas funções custo para a obtenção da lei de controle. Em geral, se deseja que as saídas futuras, em um horizonte definido, sigam um determinado sinal de referência, ao mesmo tempo em que se possa penalizar o esforço de controle requerido. A expressão geral da função objetivo é mostrada na seguinte equação (CAMACHO; BORDONS, 2007):

$$J(p_1, p_2, m_h) = \sum_{i=p_1}^{p_2} \gamma(i) [y(t+i|t) - w(t+i)]^2 + \sum_{i=1}^{m_h} \lambda(i) [\Delta u(t+i-1)]^2 \quad (2.8)$$

Em alguns métodos, o segundo somatório, que considera o esforço de controle, não é considerado, mas em outros aparece diretamente os valores do sinal de controle. Os parâmetros  $p_1$  e  $p_2$  são as variáveis mínimo e máximo da predição, marcam os limites dos instantes que se deseja que a saída siga a referência. Se o processo apresenta tempo morto  $d$ , não tem sentido que  $p_1$  seja menor que  $d$ , já que a saída só evoluirá no instante  $t+d$  e se o processo é de fase não mínima,  $p_1$  permite eliminar da função objetivo os primeiros instantes da resposta

inversa. Por outro lado,  $m_h$  é o horizonte de controle.

Os coeficientes  $\gamma(i)$  e  $\lambda(i)$  são sequências que ponderam o comportamento futuro. Geralmente são considerados valores constantes ou sequências exponenciais, como se mostra na seguinte equação:

$$\gamma(i) = \varphi^{p2-i} \quad (2.9)$$

Se  $\varphi$  está entre 0 e 1, significa que está se penalizando mais os erros que estão afastados do instante  $t$  que os mais próximos, gerando um controle mais suave requerendo um menor esforço. Por outro lado, se  $\varphi > 1$  estão se penalizando mais os primeiros erros, provocando um controle mais agressivo.

### 2.1.3 Trajetória das referências

Uma vantagem do controle preditivo é que as referências são conhecidas *a priori*. Assim, o sistema pode começar a agir antes que a mudança seja efetivamente realizada, evitando os efeitos do retardo na resposta do processo.

A maioria dos métodos utiliza no critério de minimização uma trajetória de referência  $w(t+k)$ , que geralmente é uma aproximação suave de um sistema de primeira ordem do valor atual da saída  $y(t)$  à referência, mostrada na equação (2.10).

$$\begin{aligned} w(t) &= y(t) \\ w(t+k) &= \alpha w(t+k-1) + (1-\alpha) \\ k &= 1 \dots\dots p \end{aligned} \quad (2.10)$$

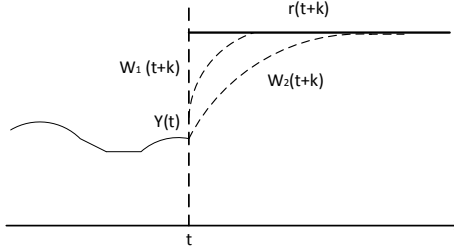
$\alpha$  é um parâmetro entre 0 a 1 (em quanto mais perto de um mais suave é a aproximação), que constitui um valor ajustável que influenciará na resposta dinâmica do sistema. Na Figura 3, mostra-se a forma da trajetória quando a referência  $r(t+k)$  é constante.

Na Figura 3, mostra-se que  $W_1$  com um valor pequeno de  $\alpha$  se tem um seguimento de referência rápido, mas para  $W_2$  com um valor maior para  $\alpha$  a resposta fica mais suave.

### 2.1.4 Restrições

É importante considerar restrições no sistema, já que todos os processos estão sujeitos a limitações. Pode-se citar como exemplo as

Figura 3 – Trajetória da referência.



**Fonte: (CAMACHO; BORDONS, 2007).**

restrições mecânicas de atuadores e válvulas, que estão limitadas pela capacidade mecânica de abertura e fechamento, assim como pela velocidade de resposta. As restrições também podem ser impostas pelos sensores do processo. Sendo assim, faz-se necessária a introdução de restrições na função custo para representar as limitações do sistema.

Geralmente, são considerados os limites na amplitude do sinal de controle, na variação do sinal de controle e limites nos sinais de saídas, como observado nas equações:

$$\begin{aligned}
 u_{\min} &\leq u(t) && \leq u_{\max} \quad \forall t \\
 \Delta u_{\min} &\leq u(t) - u(t-1) && \leq \Delta u_{\max} \quad \forall t \\
 y_{\min} &\leq y(t) && \leq y_{\max} \quad \forall t
 \end{aligned} \tag{2.11}$$

### 2.1.5 Obtenção da Lei de Controle

O último passo que compõe a obtenção dos valores de controle nas estratégias de Controle Preditivo é a determinação do valor de  $\Delta u(t+k | t)$  com  $i = 0 \dots m_h - 1$ . Para isso, seguem-se os seguintes passos:

1. cálculo do valor de  $\tilde{y}(t+k | t)$  como função dos controles futuros;
2. substituição dos valores  $\tilde{y}$  e  $w$  na função  $J$ ;
3. minimização da função  $J$ , considerando as ponderações e as restrições, se existirem.

Desta forma, é possível obter uma sequência de ações de controle futuras, mas a aplicação do valor de controle segue o conceito do

horizonte deslizante. Assim, apesar de se obter uma sequência futura de controle, apenas a primeira ação de controle é aplicada (BORDONS, 2000).

## 2.2 ALGORITMO MPC UTILIZADO NA INDÚSTRIA

Se apresenta a continuação um dos principais algoritmos de Controle Preditivo, o método considerado o mais representativo no entorno industrial, o DMC. A seguir apresentam-se suas principais características e formulações matemáticas.

### 2.2.1 Controle por Matriz Dinâmica-DMC

O algoritmo de controle preditivo denominado Controle por Matriz Dinâmica (DMC) é um dos algoritmos mais difundidos na indústria. O DMC foi desenvolvido no final dos anos setenta por (CUTLER; REMARKER, 1980) na Shell Oil, e foi muito bem aceito no meio industrial, principalmente na indústria petroquímica.

Primeiro é apresentado o algoritmo original do DMC proposto em (RICHALET; RAULT; TESTUD, 1978), e depois apresenta-se o DMC-Recursivo, proposto em (CUTLER; REMARKER, 1980), que modifica como a resposta livre é calculada de forma a obter um algoritmo mais simples de ser implementado e com a vantagem de não afetar o controlador original do DMC.

### 2.2.2 DMC Original

O modelo utilizado pelo DMC, é um modelo baseado na resposta ao degrau do sistema, enquanto as perturbações são consideradas constantes ao longo do horizonte  $p$ .

A forma de obter as previsões é mostrado a seguir. O modelo da resposta ao degrau pode ser expresso pela equação (2.12).

$$y(t) = \sum_{i=1}^{\infty} g_i \Delta u(t-i) \quad (2.12)$$

As previsões ao longo do horizonte  $p$ , são calculadas adicionando



a predição de perturbação  $\tilde{n}(t+k|t)$ , como mostrado:

$$\tilde{y}(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) + \tilde{n}(t+k|t)$$

$$\tilde{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) + \tilde{n}(t+k|t) \quad (2.13)$$

Os termos da equação (2.13) dependentes do controle foram separados em duas somas: uma com os valores passados e outra com os valores futuros. A perturbação é considerada constante. Da forma:  $\tilde{n}(t+k|t) = \tilde{n}(t|t) = y_m(t) - \tilde{y}(t|t)$ , assim pode-se reescrever:

$$\tilde{y}(t+k|t) = \sum_{i=1}^{\infty} g_i \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \Delta u(t-i) \quad (2.14)$$

$$\tilde{y}(t+k|t) = \sum_{i=1}^k g_i \Delta u(t+k-i) + f(t+k)$$

onde  $f(t+k)$ , é a resposta livre do sistema, dada pela equação (2.15).

$$f(t+k) = y_m(t) + \sum_{i=1}^{\infty} (g_{k+1} - g_i) \Delta u(t-1) \quad (2.15)$$

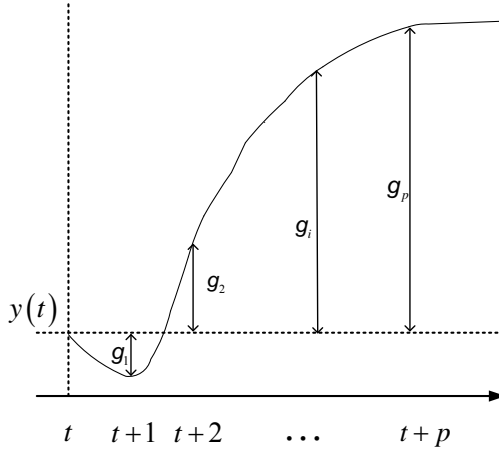
Se o sistema é estável, os coeficientes  $g_i$  da resposta ao degrau tendem a um valor constante após  $p$  amostras, como é mostrado na Figura 4, logo  $g_{k+1} - g_i \approx 0$ , para  $i > p$ , com isso a resposta livre da equação (2.15) pode ser calculada conforme a equação (2.16).

$$f(t+k) = y_m(t) + \sum_{i=1}^p (g_{k+1} - g_i) \Delta u(t-1) \quad (2.16)$$

Assim, pode-se calcular as predições ao longo do horizonte de predição ( $k = 1, \dots, p$ ), considerando  $m_h$  ações de controle:

$$\begin{aligned} \tilde{y}(t+1|t) &= g_1 \Delta u(t) + f(t+1) \\ \tilde{y}(t+2|t) &= g_2 \Delta u(t) + g_1 \Delta u(t+1) + f(t+2) \\ &\vdots \\ \tilde{y}(t+p|t) &= \sum_{i=p-m_h+1}^p g_i \Delta u(t+p-i) + f(t+p) \end{aligned} \quad (2.17)$$

Figura 4 – Resposta au degrau.



Fonte: (CAMACHO; BORDONS, 2007).

As predições do sistema de forma matricial podem ser escritas como:

$$\tilde{\mathbf{y}} = \mathbf{f} + \mathbf{G}\Delta\mathbf{u} \quad (2.18)$$

onde,  $\mathbf{G}$  é a matriz dinâmica do sistema,  $\Delta\mathbf{u}$  é o vetor das ações de controle futuras, e  $\mathbf{f}$  é o vetor da resposta livre do sistema:

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{m_h} & g_{m_h-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ g_p & g_{p-1} & \cdots & g_{p-m_h+1} \end{bmatrix}, \Delta\mathbf{u} = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m_h) \end{bmatrix}, \mathbf{f} = \begin{bmatrix} f(t+1) \\ f(t+2) \\ \vdots \\ \vdots \\ \vdots \\ f(t+p) \end{bmatrix} \quad (2.19)$$

Como é observado na equação (2.19), para o cálculo de  $\tilde{\mathbf{y}}$  seria necessário armazenar os valores passados do controle. Uma forma de simplificar este cálculo consiste no uso do DMC-Recursivo, descrito na seção seguinte.

### 2.2.3 DMC-Recursivo

Existe uma outra forma para implementar o algoritmo DMC, chamada de DMC-Recursivo, apresentada por (CUTLER; REMARKER, 1980). As predições de malha aberta em  $t + k$ , dadas as informações em  $t$  e  $t - 1$ , são respectivamente dados por:

$$y_0(t + k | t) = \sum_{i=k+1}^{\infty} g_i \Delta u(t + k - i); \quad (2.20)$$

$$y_0(t + k | t - 1) = \sum_{i=k+2}^{\infty} g_i \Delta u(t + k - i), \quad (2.21)$$

na qual a diferença entre a predição de malha aberta em  $t + k$  dado  $t$ , e em  $t + k$  dado  $t - 1$ , é apenas a nova ação de controle que não é conhecida em  $\Delta u(t - 1)$ . Subtraindo as equações (2.20) e (2.21) tem-se:

$$\begin{aligned} y_0(t + k | t) - y_0(t + k | t - 1) = \\ g_{k+1} \Delta u(t - 1) + \sum_{j=k+2}^{\infty} g_j \Delta u(t + k - j) + \\ - \sum_{j=k+2}^{\infty} g_j \Delta u(t + k - j) \end{aligned} \quad (2.22)$$

Assim, obtém-se:

$$y_0(t + k | t) = g_{k+1} \Delta u(t - 1) + y_0(t + k | t - 1) \quad (2.23)$$

No cálculo do DMC-Recursivo, a resposta livre é calculada da seguinte forma:

$$\mathbf{Y}_{livre} = [y_0(t | t - 1), \dots, y_0(t + N - 1 | t - 1)]^T \quad (2.24)$$

onde  $N$  é chamado de horizonte de modelo, mantendo em memória um vetor com  $N$  elementos<sup>2</sup>. Seus elementos são as respostas livres dadas as ações de controle conhecidas até o instante  $t - 1$ . Ao iniciar o controlador DMC no instante  $t_0$ , dado que o sistema encontra-se em regime permanente, considera-se que as predições de malha aberta sejam constantes e iguais ao valor de  $y(t_0)$ .

Ao iniciar a execução do algoritmo DMC no instante  $t$ , é preciso atualizar o vetor da resposta livre, pois o incremento da ação de controle  $\Delta \mathbf{u}(t - 1)$  agora é conhecido. O vetor  $\mathbf{Y}_{livre}$  da equação (4.7) é

---

<sup>2</sup>O valor de  $N$  deve ser significativamente maior que  $p$ .

atualizado.

$$\mathbf{Y}_{livre} = \mathbf{Y}_{livre} + \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} \Delta \mathbf{u}(t-1) \quad (2.25)$$

Após obter o novo sinal de controle  $u(t)$ , é preciso deslocar os valores dentro do vetor  $\mathbf{Y}_{livre}$ . Isto porque no instante posterior  $t+1$ , serão necessárias respostas livres de  $t+1$  até  $t+N$  baseadas nos dados até  $t$ . Então, o primeiro elemento é descartado, pois é a predição relativa ao instante  $t$ , que já passou. O problema é que, ao se fazer este deslocamento, o último elemento, no caso a predição em  $y(t+N|t)$ , é desconhecida. Porém, se o sistema for estável, pode-se considerar  $y_0(t+N|t) \cong y_0(t+N-1|t)$ . Assim, o novo vetor  $\mathbf{Y}_{livre}$  será dado pela equação (4.8):

$$\mathbf{Y}_{livre} = \begin{bmatrix} y_0(t+1|t) \\ y_0(t+2|t) \\ \vdots \\ y_0(t+N-1|t) \\ y_0(t+N-1|t) \end{bmatrix} \quad (2.26)$$

A condição  $y_0(t+N|t) \cong y_0(t+N-1|t)$  é mostrada como:

$$\begin{aligned} y_0(t+N-1|t) &= \sum_{i=N}^{\infty} g_i \Delta u(t+N-1-i) \\ y_0(t+N|t) &= \sum_{i=N+1}^{\infty} g_i \Delta u(t+N-i) = \sum_{i=N}^{\infty} g_{i+1} \Delta u(t+N-1-i) \\ y_0(t+N|t) - y_0(t+N-1|t) &= \sum_{i=N}^{\infty} (g_{i+1} - g_i) \Delta u(t+N-1-i) \end{aligned}$$

Como  $g_{i+1} - g_i \cong 0, \forall i > N$ , tem-se que  $y_0(t+N|t) \cong y_0(t+N-1|t)$ , como se queria demonstrar.

Assim, o cálculo da resposta livre considerada no DMC-Recursivo é realizado através da equação:

$$\mathbf{f} = \begin{bmatrix} y_0(t+1|t) \\ \vdots \\ y_0(t+p|t) \end{bmatrix} + \mathbf{1}_{p \times 1} e \quad (2.27)$$

No algoritmo DMC, o erro entre a saída medida e a saída predita em malha aberta no instante  $k$  é usado para corrigir as predições de

cada estado de todos os elementos dentro do horizonte de predição  $p$ . Nesse caso, a equação do erro do modelo é:

$$e(t) = y_m(t) - y_0(t|t) \quad (2.28)$$

Como é mostrado na equação (2.18), a representação das predições, considerando o cálculo da resposta livre com o DMC-Recursivo, é obtida com a equação (4.11):

$$\tilde{\mathbf{Y}} = \mathbf{f} + \mathbf{G}\Delta\mathbf{u} \quad (2.29)$$

onde:

- $\tilde{\mathbf{Y}}$  saída predita  $p \times 1$  ;
- $\mathbf{f}$  vetor da resposta livre de dimensão  $p \times 1$  ;
- $\mathbf{G}$  matriz Jacobiana de  $\tilde{\mathbf{Y}}$  de dimensão  $p \times m_h$ ;
- $\Delta\mathbf{u}$  vetor de dimensão  $m_h \times 1$  que contém os incrementos na ação de controle .

Onde  $p$  é o horizonte de predição e  $m_h$  é o horizonte de controle.

Utilizando esta forma de implementação, os cálculos necessários para obter a saída predita tornam-se mais simples, não sendo necessário o armazenamento dos incrementos de controle passados. Com este procedimento, só é alterado o modo de calcular a resposta livre, o restante do algoritmo de controle não é modificado.

## 2.3 CONCLUSÃO

Neste capítulo fez-se uma revisão da teoria de Controle Preditivo, mostrando a importância de utilizar um modelo de predição para as saídas futuras e a utilização de um procedimento específico de otimização para a obtenção dos valores de controle futuro.

O algoritmo mais utilizado na aplicação industrial foram descritas nesta seção. O DMC original e o DMC-Recursivo com as respectivas formulações matemáticas.

Os algoritmos DMC para o caso SISO (*Single-Input Single-Output*) foram apresentados na seção 2.2, e podem ser estendidos para o caso MIMO (*Multiple-Input Multiple-Output*), para um sistema com  $m$  entradas e  $n$  saídas.

Nos capítulos posteriores, será aplicado o DMC-Recurso em diversos estudos de caso. Neste capítulo descreveu-se a notação matemática da técnica, para sua posterior implementação, tornando-se assim a base deste trabalho.

## Capítulo 3

---

# MPC PRÁTICO PARA SISTEMAS NÃO LINEARES - PNMPC

---

Este capítulo está centrado em aspectos de Controle Preditivo Não Linear, especificamente na técnica PNMPC.

Atualmente, no contexto de Controle Preditivo para processos lineares, as estratégias mais utilizadas na indústria, em geral, são o DMC e GPC (CAMACHO; BORDONS, 2007).

Para conseguir implementar o Controle Preditivo, é muito importante ter um modelo do processo. Como a maioria dos processos são efetivamente não lineares, na maioria das vezes recorre-se às técnicas de linearização para a obtenção do modelo. Para muitos processos este procedimento é aceitável, seja pelo pequeno nível de não linearidade do processo ou pelo fato da planta operar sempre em uma região muito próxima do ponto de operação (ALLGOWER; FINDEISEN, 2004). Porém, quando os processos têm a dinâmica não linear ou quando a faixa de operação é variável, então necessariamente deverá ser utilizado o modelo não linear para o projeto de controle, de forma que permita manter o desempenho desejado para o sistema em malha fechada (BORDONS, 2000).

Do ponto de vista conceitual, o MPC não linear (NMPC) não apresenta dificuldades. Considera a mesma função objetivo  $J$  que no caso linear e não linear, para o cálculo das previsões. O objetivo consiste novamente em encontrar uma ação de controle que minimize a função objetivo num algoritmo de horizonte deslizante. Porém, do ponto de vista prático, são várias as dificuldades encontradas neste tipo de projeto (se comparado ao caso linear):

1. determinação do modelo do processo quando este deve ser determinado através de uma técnica de identificação;

2. obtenção do sinal de controle requer, em geral, a solução de um problema de otimização não convexo;
3. análise de estabilidade e robustez da solução.

Por estes motivos, o NMPC é um campo aberto para pesquisas tanto na área de determinação de modelos como nos procedimentos e algoritmos de otimização para o cálculo de controle (CAMACHO; BORDONS, 2007).

De acordo com vários autores (PRADA, 2004) e (QIN; BADGWELL, 2003), as tendências no controle avançado de processos, e o desenvolvimento tecnológico da próxima geração de MPCs estão associados a controladores não lineares baseados em modelos (NMPC).

Existe uma grande quantidade de técnicas MPC que, direta ou indiretamente, tratam sistemas não-lineares. Estas técnicas se classificam em três grupos:

1. técnicas que utilizam diretamente um modelo não linear genérico e um algoritmo de otimização não linear de propósito geral;
2. técnicas que utilizam modelos não lineares particulares e métodos de otimização específicos para o modelo escolhido;
3. técnicas que utilizam um ou vários modelos lineares para representar o processo e algoritmos de otimização lineares.

Evidentemente, as vantagens de uma ou outra alternativa estão diretamente relacionadas ao compromisso desempenho-custo computacional (CAMACHO; BORDONS, 1995).

Geralmente, muitos processos não lineares são modelados através da utilização de algoritmos de identificação, com exceção de redes neurais, que fornecem as saídas nominais dos processos dadas as entradas e os estados atuais. Com a premissa da existência de tais abordagens de modelagem, mais recentemente, em (PLUCENIO, 2010), foi desenvolvida uma estratégia denominada PNMPC para o controle de muitos desses processos não lineares.

Dos três grupos de técnicas de controle preditivo não linear citadas acima, o terceiro se encaixa com a técnica PNMPC, que vem apresentando bom desempenho e a vasta possibilidade de aplicações em processo com dinâmica não linear, o que se torna interessante quando é avaliada a simplicidade da sua implementação.

A estratégia PNMPC tem como principal função a aproximação linear do modelo de predição não linear a cada instante de amostragem



e a consecutiva otimização quadrática, similar à formulação tradicional dos algoritmos DMC e GPC. No entanto, o diferencial dessa técnica é a maneira pela qual ela define a linearização do modelo de predição não linear a cada instante de amostragem. Nela também é utilizada a formulação tradicional de cálculo da predição, dividindo a resposta livre da resposta forçada, enquanto se mantém maior proximidade da resposta real do processo ao utilizar um modelo não linear para determinar a resposta livre.

O principal objetivo do algoritmo PNMPC, consiste em fornecer um caminho simples e eficaz para o Controle Preditivo de sistemas não lineares. A premissa inicial é o fato de que os algoritmos de controle, como o DMC e o GPC, utilizam uma representação para a predição ao longo do horizonte de predição,  $\tilde{\mathbf{y}}$ , como uma função do vetor de incrementos futuros de controle  $\Delta \mathbf{u}$ , conforme a equação (PLUCENIO, 2010):

$$\tilde{\mathbf{y}} = \mathbf{f} + \mathbf{G}\Delta \mathbf{u} \quad (3.1)$$

Sabe-se também que, mesmo através de caminhos diferentes, em ambas as estratégias (GPC e DMC), o princípio de superposição presente em modelos lineares permite uma forma simples de calcular a saída predita  $\tilde{\mathbf{y}}$ . No entanto, para o caso não linear, o princípio da superposição não é válido, e assim, busca-se descrever a evolução dos estados do sistema ao longo do horizonte de predição  $p$ , utilizando uma representação linear dos estados em relação aos incrementos de controle futuro, similar à equação (3.1).

Como, independentemente da linearidade dos sistemas, as únicas variáveis que podem alterar o valor das predições futuras são de fato os incrementos futuros  $\Delta \mathbf{u}$ , utiliza-se uma linearização em função dessas variáveis e não o conceito de ponto de equilíbrio. Então, assume-se que as predições para esse caso,  $\tilde{\mathbf{y}}$ , dependem dos controles passados  $\overleftarrow{u}$ , das saídas presentes e passadas  $\overleftarrow{y}$  e dos incrementos de controle futuros  $\Delta \mathbf{u}$ , como se mostra na equação (3.2).

$$\tilde{\mathbf{y}} = \mathbf{f}(\overleftarrow{y}, \overleftarrow{u}, \Delta \mathbf{u}) \quad (3.2)$$

Assim, com a aproximação proposta na equação (3.2), o vetor de predições pode ser reescrito por:

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{G}_{PNMPC}\Delta \mathbf{u} \quad (3.3)$$

onde  $\mathbf{F} = \mathbf{f}(\overleftarrow{y}, \overleftarrow{u})$  e  $\mathbf{G}_{PNMPC} = \frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}}$ . Vale ressaltar que a repre-

sentação das predições da equação (3.3), nada mais é do que uma linearização de primeira ordem da série de Taylor para  $\Delta \mathbf{u} = 0$ , podendo também ser indicada como uma linearização pela série de Maclaurin.

Para demonstrar a formulação da estratégia, considera-se o sistema dinâmico não linear contínuo e diferenciável com uma entrada e uma saída, representado por:

$$\begin{aligned} y(k) &= f(y(k - n_a), y(k - n_a + 1), \dots, y(k - 1), \\ u(k - n_u), u(k - n_u + 1), \dots, u(k - 1)), \end{aligned} \quad (3.4)$$

sendo  $n_a$  e  $n_u$  os maiores atrasos de  $y$  e  $u$ , respectivamente.

Considerando um horizonte de controle futuro  $m_h$ , deverão aparecer as entradas futuras na forma representada abaixo:

$$\begin{aligned} u(k) &= u(k - 1) + \Delta u(k) \\ u(k + 1) &= u(k) + \Delta u(k + 1) \\ u(k + 2) &= u(k - 1) + \Delta u(k) + \Delta u(k + 1) \\ &\vdots \\ u(k + m_h - 1) &= u(k - 1) + \Delta u(k) + \dots + \Delta u(k + m_h - 1) \end{aligned} \quad (3.5)$$

O horizonte futuro é limitado a  $m_h$ , onde deve-se assumir que as entradas posteriores a  $k + m_h - 1$  são iguais a  $u(k + m_h - 1)$ . Utilizando a equação (3.4) para calcular as predições, tem-se:

$$\begin{aligned} \tilde{y}(k + 1) &= f(y(k - n_a + 1), \dots, y(k), u(k - n_u + 1), \dots, u(k)) \\ \tilde{y}(k + 2) &= f(y(k - n_a + 2), \dots, y(k + 1), u(k - n_u + 2), \dots, u(k + 1)) \\ &\vdots \\ \tilde{y}(k + p) &= f(\tilde{y}(k - n_a + p), \dots, \\ \tilde{y}(k + p - 1), u(k - n_u + 1), \dots, u(k + m - 1)) \end{aligned} \quad (3.6)$$

Substituindo as relações apresentadas em (3.5), o conjunto de equações apresentadas em (3.6) passa a ser dado por:

$$\begin{aligned}
\tilde{y}(k+1) &= f(y(k-n_a+1), \dots, y(k), u(k-n_u+1), \dots, u(k-1), \Delta u(k)) \\
\tilde{y}(k+2) &= f(y(k-n_a+2), \dots, \tilde{y}(k+1), u(k-n_u+2), \dots, u(k-1), \\
&\quad \Delta u(k), \Delta u(k+1)) \\
&\vdots \\
\tilde{y}(k+p) &= f(\tilde{y}(k-n_a+p), \dots, \tilde{y}(k+p-1), u(k-n_u+1), \dots, \\
&\quad u(k-1), \Delta u(k), \Delta u(k+1), \dots, \Delta u(k+m_h-1))
\end{aligned} \tag{3.7}$$

Para simplificar a notação, o conjunto de entradas passadas  $\overleftarrow{u}$  e das saídas passadas  $\overleftarrow{y}$ , passa a ter a forma da expressão em (3.7) e se reduzem à seguinte equação:

$$\begin{aligned}
\tilde{y}(k+1) &= f(\overleftarrow{y}, \overleftarrow{u}, \Delta u(k)) \\
\tilde{y}(k+2) &= f(\overleftarrow{y}, \overleftarrow{u}, \Delta u(k), \Delta u(k+1)) \\
&\vdots \\
\tilde{y}(k+p) &= f(\overleftarrow{y}, \overleftarrow{u}, \Delta u(k), \dots, \Delta u(k+m_h-1))
\end{aligned} \tag{3.8}$$

Observando a equação (3.8), é possível concluir que as únicas variáveis capazes de alterar o valor das saídas futuras são os incrementos futuros da ação de controle  $\Delta \mathbf{u}(k)$ ,  $\Delta \mathbf{u}(k+1)$ , ... e  $\Delta \mathbf{u}(k+m_h-1)$ .

Dadas as definições mostradas em (3.9), sendo  $\mathbf{l}$  um vetor de dimensão  $m_h \times 1$  com todos os elementos iguais a um, e  $\mathbf{D}$  uma matriz triangular inferior  $m_h \times m_h$  com todos os elementos não nulos iguais a um, é possível simplificar a notação como:

$$\begin{aligned}
\overleftarrow{y} &: \text{Conjunto de valores passados de } y \\
\overleftarrow{u} &: \text{Conjunto de valores passados de } u \\
u &: u_0 + \delta u \\
u_0 &: u(k-1)\mathbf{l} \\
\delta u &: \mathbf{D}\Delta \mathbf{u} \\
\Delta \mathbf{u} &: [\Delta \mathbf{u}(k), \Delta \mathbf{u}(k+1) \dots \Delta \mathbf{u}(k+m_h-1)]^T
\end{aligned} \tag{3.9}$$

Assim, as predições em (3.7) podem ser reescritas como segue:

$$\tilde{\mathbf{y}} = \mathbf{f}(\overleftarrow{y}, \overleftarrow{u}, \Delta \mathbf{u}) \tag{3.10}$$

Utilizando o mesmo procedimento exposto na equação (3.3), é possível

encontrar a seguinte relação:

$$\tilde{\mathbf{y}} = \mathbf{f}(\overleftarrow{\mathbf{y}}, \overleftarrow{\mathbf{u}}, u_0) + \left. \frac{\partial \tilde{\mathbf{y}}}{\partial u} \right|_{u=u_0} \delta u \quad (3.11)$$

Substituindo em (3.11) as equações (3.9), tem-se a equação:

$$\frac{\partial \tilde{\mathbf{y}}}{\partial u} \delta u = \frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}} \Delta \mathbf{u} \quad (3.12)$$

Finalmente, definindo  $\mathbf{F} = \mathbf{f}(\overleftarrow{\mathbf{y}}, \overleftarrow{\mathbf{u}}, u_0)$  e  $\mathbf{G}_{PNMPC} = \left. \frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}} \right|_{u=u_0}$ , sendo  $\mathbf{F}$  o vetor de predições que seria obtido para  $\Delta \mathbf{u} = 0$  e  $\mathbf{G}_{PNMPC}$  o gradiente das saídas previstas com relação ao vetor dos incrementos de controle  $\Delta \mathbf{u}$  calculado para  $\Delta \mathbf{u} = 0$ , tem-se a equação (3.13), voltando à notação presente na equação (3.3).

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{G}_{PNMPC} \Delta \mathbf{u} \quad (3.13)$$

$\mathbf{G}_{PNMPC}$  é a matriz de PNMPC, que pode ser obtida pela definição do Jacobiano como se mostra em (3.14), que por sua vez é a matriz dinâmica generalizada e é aplicado para sistemas lineares e não lineares.

$$\mathbf{G}_{PNMPC} = \begin{bmatrix} \frac{\partial \tilde{y}(k+1)}{\partial \Delta u(k)} & 0 & 0 & 0 \\ \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k+1)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+1)} & \cdots & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+m_h-1)} \end{bmatrix} \quad (3.14)$$

A forma triangular inferior da matriz  $\mathbf{G}_{PNMPC}$ , deve-se ao fato do sistema ser causal e de se considerar que não há acoplamento direto entre entrada e saída. Em (PLUCENIO et al., 2007b) prova-se que, para sistemas lineares, a matriz  $\mathbf{G}_{PNMPC}$  é igual a matriz  $\mathbf{G}$  obtida pela resposta ao degrau unitário nas técnicas DMC e GPC.

A equação (3.13) pode ser escrita para um sistema de uma entrada e uma saída como se mostra na expressão seguinte:

$$\begin{bmatrix} \tilde{y}(k+1) \\ \tilde{y}(k+2) \\ \vdots \\ \tilde{y}(k+p) \end{bmatrix} = \begin{bmatrix} f_1(\overleftarrow{y}, \overleftarrow{u}) \\ f_2(\overleftarrow{y}, \overleftarrow{u}) \\ \vdots \\ f_p(\overleftarrow{y}, \overleftarrow{u}) \end{bmatrix} + \begin{bmatrix} \frac{\partial \tilde{y}(k+1)}{\partial \Delta u(k)} & 0 & 0 & 0 \\ \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k+1)} & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+1)} & \cdots & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+m_h-1)} \end{bmatrix}_{\Delta u=0} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m_h-1) \end{bmatrix} \quad (3.15)$$

Nota-se em (3.15) que a forma triangular inferior da matriz Jacobiano deve-se à causalidade do sistema, dado por (3.16):

$$\frac{\partial \tilde{y}(k+j)}{\partial \Delta u(k+i)} = 0 \quad \text{para } i \geq j \quad (3.16)$$

A expressão (3.15) representa a equação (3.13) de forma mais detalhada, onde:

- $\tilde{\mathbf{Y}}$  é o vetor  $p \times 1$  contendo as previsões no horizonte de predição  $p$ .
- $\mathbf{F}$  é o vetor  $p \times 1$  que contém as saídas previstas, supondo que as ações de controle no horizonte de controle  $m_h$  são mantidas iguais à ação de controle no instante  $k-1$ ,  $u(k) = u(k+1) = \dots = u(k+m_h-1) = u(k-1)$ .
- $\mathbf{G}_{PNMPC}$  é a versão não linear da matriz dinâmica utilizada no Controle Preditivo DMC ou a versão generalizada, válida para sistemas lineares ou não lineares (para sistemas contínuos e diferenciáveis). Sua dimensão é  $p \times m_h$ .
- $\Delta \mathbf{u}$  é o vetor  $m_h \times 1$  que contém os incrementos nas ações de controle no horizonte de controle  $m_h$ .

Apesar de ser possível, em alguns casos, obter-se uma expressão para a matriz  $\mathbf{G}_{PNMPC}$  em função da expressão matemática do modelo não-linear e procurando uma alternativa onde a montagem da matriz não dependa da estrutura matemática utilizada para a representação do modelo, optou-se pela determinação numérica da matriz Jacobiana (PLUCENIO et al., 2007a).

Diferentemente das estratégias de Controle Preditivo lineares GPC e o DMC, nas quais a técnica PNMPC se baseia, o enfoque dado por (PLUCENIO, 2010) foi a solução numérica do problema de prever as saídas futuras. Isto se deve principalmente ao fato de existir uma grande quantidade de caminhos diferentes para a modelagem de processos não lineares, como por exemplo: redes neurais artificiais, modelos NARMAX (*Non-linear Auto-Regressive with Moving Average and Exogenous inputs*), etc. Assim, ao invés de pressupor a disponibilidade de um modelo matemático do processo para se obter as previsões, para esse propósito foi concebida uma estrutura compatível à modelagem de processos através de algoritmos numéricos, aumentando a praticidade da técnica.

### 3.1 OBTENÇÃO NUMÉRICA DAS PREDIÇÕES

O algoritmo numérico necessário para calcular as previsões deve ser capaz de fornecer as saídas futuras  $\tilde{\mathbf{Y}}$ , quando fornecidos os valores dos sinais de controles e saídas passados com o vetor de incrementos de controles futuros  $\Delta \mathbf{u}$ . Para um sistema SISO, por exemplo, a cada iteração executa-se o seguinte procedimento para a obtenção numérica de  $\mathbf{F}$  e  $\mathbf{G}_{PNMPC}$ :

1. obter o vetor  $p \times 1$ ,  $\tilde{\mathbf{Y}}_p^0$ , executando o algoritmo que modela o processo, utilizando para isso incrementos futuros nulos  $\Delta \mathbf{u} = [0 \dots 0]^T$ ;
2. fazer  $\mathbf{F} = \tilde{\mathbf{Y}}_p^0$ ;
3. obter o vetor  $\tilde{\mathbf{Y}}_p^1$  executando o algoritmo que modela o processo, utilizando para isso as entradas e saídas passadas, com um valor  $\varepsilon$  pequeno e não nulo como primeiro incremento futuro, por exemplo,  $\frac{u_{k-1}}{1000}$ . Para o caso de uma entrada  $\Delta \mathbf{u} = [\varepsilon \ 0 \ \dots \ 0]^T$ ;
4. fazer  $\mathbf{G}_{PNMPC}(:, 1) = \frac{\tilde{\mathbf{Y}}_p^1 - \tilde{\mathbf{Y}}_p^0}{\varepsilon}$ , ou seja, a derivada parcial aproximada para a saída e a entrada em questão;
5. obter o vetor  $\tilde{\mathbf{Y}}_p^2$  executando o algoritmo que modela o processo utilizando para isso as entradas e saídas passadas, com um valor  $\varepsilon$  pequeno e não nulo como segundo incremento futuro, por exemplo,  $\frac{u_{k-1}}{1000}$ . Para o caso de uma entrada  $\Delta \mathbf{u} = [0 \ \varepsilon \ \dots \ 0]^T$ ;

6. fazer  $\mathbf{G}_{PNMPC}(:, 2) = \frac{\tilde{\mathbf{Y}}_p^2 - \tilde{\mathbf{Y}}_p^0}{\varepsilon}$ , ou seja, a derivada parcial aproximada para a saída e a entrada em questão;
7. proceder com o mesmo cálculo das demais colunas da matriz  $\mathbf{G}_{PNMPC}$  até a última coluna onde obtém-se o vetor  $\tilde{\mathbf{Y}}_p^{m_h}$  executando o modelo com as entradas e saídas passadas e com  $\Delta \mathbf{u} = [0 \ 0 \ \dots \ \varepsilon]$ .  $\mathbf{G}_{PNMPC}(:, m_h) = \frac{\tilde{\mathbf{Y}}_p^{m_h} - \tilde{\mathbf{Y}}_p^0}{\varepsilon}$

Após a obtenção do vetor de predições  $\tilde{\mathbf{Y}}$  da equação (3.13), utiliza-se o mesmo método de cálculo numérico presente nos algoritmos GPC e DMC para a obtenção da ação de controle. É interessante ressaltar que no procedimento de cálculo de  $\Delta \mathbf{u}$ , a única diferença do PN MPC perante o GPC e o DMC, é o fato da resposta livre ser não linear (PLUCENIO, 2010).

### 3.2 TRATAMENTO DO ERRO DE PREDIÇÃO E PERTURBAÇÃO

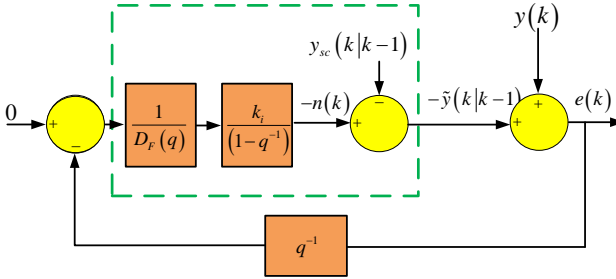
Os algoritmos de controle preditivo, como o DMC e o GPC, utilizam um mecanismo para corrigir as predições. Por exemplo, a técnica de Controle DMC corrige as predições com o erro entre o valor medido da variável controlada e seu valor predito com o modelo no instante  $k$  (PLUCENIO, 2010). Esta técnica aplicada a sistemas lineares garante erro nulo em regime permanente, já que o incremento do esforço de controle necessário para a correção do erro no instante  $k$  será o mesmo nos instantes futuros, devido à relação linear entre a entrada e saída, isto faz com o que o ganho seja constante.

A aplicação deste mesmo procedimento para a técnica PN MPC, aplicada a sistemas não lineares, não garante erro nulo em regime permanente, já que o ganho varia com o ponto de operação do sistema (PLUCENIO, 2010). A estratégia PN MPC utiliza uma versão explícita do mecanismo de correção das predições utilizado pelo algoritmo GPC.

Para o tratamento de erros de predição, ruído e perturbações encontrado em (PLUCENIO, 2010), o algoritmo PN MPC soma a cada predição a integral do erro de predição filtrado. Na Figura 5, observa-se o diagrama de blocos para a correção das predições utilizando-se o algoritmo PN MPC.

O modelo de perturbação funciona como uma espécie de observador, que fornece uma correção a ser adicionada a cada predição. A Figura 5 contém o diagrama de blocos desse observador utilizado no PN MPC, onde a cada instante de amostragem se calcula o erro entre

Figura 5 – Diagrama de Blocos para o Tratamento do Erro de Predição PN MPC.



Fonte: (PLUCENIO, 2010).

a saída que o preditor obtém para o instante  $k$ ,  $\tilde{y}(k|k-1)$  e a saída real,  $y(k)$ , conforme equação:

$$e(k) = y(k) - \tilde{y}(k|k-1) \quad (3.17)$$

Considerando a análise para uma variável controlada  $y(t)$ , substituindo o operador  $q$  por  $z$  e definindo  $D_F(z) = 1 - f_d z^{-1}$ , chegamos à função de transferência apresentada em (3.18).

$$e(z) = \frac{z^2 - (1 + f_d)z + f_d}{z^2 - (1 + f_d - k_i)z + f_d} (y(k) - \tilde{y}_{sc}(k)) \quad (3.18)$$

onde

$\tilde{y}_{sc}(k)$  é o valor da saída sem correção

$y(k)$  é o valor real da saída

Supondo que o sistema em malha fechada é estável, podemos aplicar o Teorema do Valor Final para verificarmos que a presença de  $(1 - z^{-1})$  no numerador da função de transferência (3.18) garante erro nulo em regime permanente para uma entrada do tipo degrau de  $y(k) - \tilde{y}_{sc}(k)$ .

O projeto de  $k_i$  e  $f_d$  consiste em comparar o polinômio denominador da função de transferência (3.18) com um polinômio desejado  $p_d(z) = z^2 - 2az + a^2$ . Chegamos a igualdades que permitem escolhermos o valor de  $a$  para determinar  $f_d$  e  $k_i$ , sendo que a escolha de  $a$  tem impacto na velocidade com que o erro responde à perturbação.

$$f_d = a^2 \quad (3.19)$$



$$k_i = 1 + a^2 - 2a \quad (3.20)$$

### 3.3 FUNÇÃO CUSTO

Uma função custo típica para sistemas SISO utilizada na técnica PNMPC é apresentada na equação:

$$J = \sum_{j=N_1}^{N_1+p-1} \gamma_j (\tilde{y}(k+j) - r(k+j))^2 + \sum_{i=0}^{m_h-1} \lambda_i \Delta u(k+i)^2 \quad (3.21)$$

Na equação (3.21),  $\gamma_j$  e  $\lambda_i$  são as ponderações, respectivamente sobre o seguimento de referência e o esforço de controle. Em muitos casos, as ponderações para o caso SISO são constantes.  $N_1$  é o número de amostras que determina o início do horizonte de predição. O problema a ser resolvido, consiste em obter o vetor:

$$\Delta \mathbf{u} = [\Delta u(k), \Delta u(k+1), \dots, \Delta u(k+m_h-1)] \quad (3.22)$$

que minimiza a função Custo  $J$  sujeito a restrições sobre os incrementos de controle  $\Delta \mathbf{u}$ , as ações de controle:

$$\mathbf{u} = [u(k), u(k+1), \dots, u(k+m_h-1)]^T \quad (3.23)$$

e a saída predita:

$$\tilde{\mathbf{y}}(k+j) = [\tilde{y}(k+1), \tilde{y}(k+2), \dots, \tilde{y}(k+p)]^T \quad (3.24)$$

Em sistemas que apresentam restrições na ação de controle ou na saída do sistema, a minimização da função custo é obtida com a aplicação de algoritmos de programação quadrática (*Quadratic Programming*, QP). Para isto, manipula-se a equação (3.21) para reescrevê-la conforme equação:

$$J = \frac{1}{2} \Delta \mathbf{u}^T \Phi \Delta \mathbf{u} + \Theta^T \Delta \mathbf{u} \quad (3.25)$$

sujeito à  $\mathbf{A} \Delta \mathbf{u} \leq \mathbf{B}$ , sendo:

$$\begin{aligned} \Phi &= \mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q} \\ \Theta &= \mathbf{G}^T \mathbf{R}^T (\mathbf{F} - \mathbf{r}) \end{aligned} \quad (3.26)$$

na qual  $\mathbf{R}$  é uma matriz diagonal contendo as ponderações sobre o seguimento de referência ao longo do horizonte de predição,  $\mathbf{r}$  é o vetor contendo as trajetórias futuras no horizonte de predição e  $\mathbf{Q}$  é uma matriz diagonal contendo as ponderações sobre o esforço de controle. Tendo conhecimento das restrições do sistema, basta passá-las para a forma matricial e realizar a minimização da função custo com a utilização de programação quadrática. A lei de controle resultante é mostrada na equação (3.27).

$$\Delta \mathbf{u} = (\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R}^T (\mathbf{F} - \mathbf{r}) \quad (3.27)$$

### 3.4 CONCLUSÃO

Neste capítulo apresentou-se uma discussão dos aspectos de implementação da técnica MPC Prático para Sistema Não Lineares, o PNMPC, apresentado por (PLUCENIO, 2010).

A técnica PNMPC mostra uma forma de obter uma linearização que fornece uma predição com aproximação, similar à que se obteria com uma aproximação de Taylor de segunda ordem.

Para a minimização da função objetivo, pode-se continuar utilizando toda a tecnologia já desenvolvida para sistemas lineares, como o já consagrado método de Programação Quadrática. Além disso, a obtenção numérica da matriz  $\mathbf{G}_{PNMPC}$  permite a utilização de qualquer estrutura de modelo. O método funciona igualmente para sistemas de uma entrada e uma saída como para sistemas multivariáveis e tem a sua maior área de aplicação em processos tipicamente não lineares.

A semelhança do método com os tradicionais algoritmos de LMPC (*Linear Model Predictive Control*) bem conhecidos no meio industrial, tanto no que se refere à filosofia de projeto dos controladores quanto ao ajuste e à sintonia do controlador, tornam o PNMPC uma ferramenta com grande potencial para ser adotada na indústria do petróleo e gás.

Este trabalho baseia-se na técnica PNMPC. No próximo capítulo são apresentadas as alternativas de implementação da técnica PNMPC aplicado em um estudo de caso, que foi desenvolvido ao longo deste trabalho de mestrado.

## Capítulo 4

---

# IMPLEMENTAÇÃO DO PNMPC: ALTERNATIVAS

---

A maioria dos processos dinâmicos exhibe relações não lineares entre variáveis controladas e manipuladas. Dependendo da aplicação, o algoritmo MPC é muitas vezes escolhido entre um tipo linear e um tipo não linear (CAMACHO; BORDONS, 2007).

Este capítulo apresenta várias maneiras de implementar o algoritmo PNMPC. Mostra-se que o algoritmo PNMPC pode ser escrito para lidar com a não-linearidade em diferentes níveis. Por uma questão de simplicidade, os algoritmos são implementados para controlar o nível de um tanque cônico. Nessa aplicação, a principal finalidade do controlador é manipular a abertura da válvula de saída do tanque, a fim de seguir a referência de nível do tanque. O desempenho e o custo computacional de seis implementações diferentes de controlador PNMPC são exibidos. Os resultados desses controladores são comparados e analisados pelo sistema de verificação de resposta de simulação, o tempo de execução dos algoritmos do controlador PNMPC e pelos índices de desempenho dos controladores.

### 4.1 VERSÕES DA TÉCNICA MPC PRÁTICO PARA SISTEMAS NÃO LINEARES (PNMPC)

A teoria da técnica PNMPC apresentada na seção anterior pode ser implementado de diversas formas, considerando diferentes aproximações para o cálculo das previsões de um dado processo não linear. O objetivo desta seção é estudar o desempenho e o custo computacional do controlador PNMPC, usando seis diferentes versões de imple-

mentação. Essas versões utilizam simplificações da técnica PNMPC original completa, de forma a reduzir o tempo de processamento. Variando a forma de cálculo das previsões, o PNMPC pode ser simplificado até se converter em um controlador MPC linear.

Primeiramente, é descrito a implementação e a matemática com os algoritmos das alternativas das variações da técnica PNMPC para cada versão. Depois é apresentado o estudo de caso aplicando as versões do PNMPC, mostrando os resultados respectivamente.

#### 4.1.1 Primeira versão: PNMPC original completo com calculo de $\Delta \mathbf{u}_{ini}$ .

Seguindo a teoria do PNMPC desenvolvida por Plucenio (2010) e descrito no Capítulo 3, a primeira versão PNMPC proposta neste trabalho considera resposta livre não linear e o cálculo da matriz  $\mathbf{G}_{PNMPC}$  em dois passos.

O primeiro procedimento do algoritmo é calcular em uma determinada iteração uma variação de controle inicial  $\Delta \mathbf{u}_{ini}$ , que é calculada em função da matriz  $\mathbf{G}_{PNMPC}$  já existente, ou seja, da matriz  $\mathbf{G}_{anterior}$  equivalente a  $\mathbf{G}_{PNMPC}(k-1)$ . Após obter a variação de controle inicial, calculam-se as ações de controle futuras de acordo com a equação:

$$\mathbf{U}_{m_h \times 1} = \mathbf{H}_{m_h \times 1} u(k-1) + \mathbf{T}_{\mathbf{I}_{m_h \times m_h}} \Delta \mathbf{u}_{ini_{m_h \times 1}} \quad (4.1)$$

onde:

$m_h$	horizonte de controle;
$\mathbf{H}$	vetor de uns;
$u(k-1)$	ação de controle no instante anterior;
$\mathbf{T}_{\mathbf{I}}$	matriz Toeplitz integradora;
$\Delta \mathbf{u}_{ini}$	variação de controle inicial.

Com a ação de controle  $\mathbf{U}$  já calculada, obtém-se uma saída predita inicial  $\tilde{\mathbf{Y}}_p^0$ .

Na sequência, é realizado o cálculo de uma nova ação de controle, na qual é aplicada um valor  $\varepsilon$ , e com esta nova ação de controle  $\mathbf{U}$ , é obtida a saída predita  $\tilde{\mathbf{Y}}$ . O valor  $\varepsilon$  vai-se deslocando em  $\Delta \mathbf{u}$  para o

cálculo da matriz  $\mathbf{G}_{PNMPC}$ , conforme descrito pela equação (4.2).

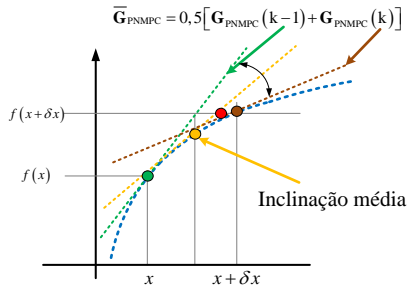
$$\mathbf{G}_{PNMPC}(:, m_h) = \frac{\tilde{\mathbf{Y}}_p^{m_h} - \tilde{\mathbf{Y}}_p^0}{\varepsilon} \quad (4.2)$$

Tendo calculada a matriz  $\mathbf{G}_{PNMPC}(k)$  no instante atual e a matriz  $\mathbf{G}_{PNMPC}(k-1)$  calculada no instante anterior, é possível obter a média da matriz  $\overline{\mathbf{G}}_{PNMPC}$ , que é calculada em dois passos:

$$\overline{\mathbf{G}}_{PNMPC} = 0,5[\mathbf{G}_{PNMPC}(k-1) + \mathbf{G}_{PNMPC}(k)] \quad (4.3)$$

A Figura 6 mostra a interpretação gráfica do procedimento de cálculo da matriz  $\overline{\mathbf{G}}_{PNMPC}$ , calculada em dois passos para uma função, representada pela linha azul tracejada. Se a posição inicial é  $f(x)$  (representada pelo primeiro ponto verde) para um  $\Delta \mathbf{u}_{ini}$  e o ponto final desejado é  $f(x + \delta x)$  (representado pelo ponto vermelho), então, para se obter o ganho da matriz  $\mathbf{G}_{PNMPC}$  em  $f(x)$ , é preciso calcular a tangente neste ponto inicial, isto é derivar  $f(x)$ . Tendo a matriz  $\mathbf{G}_{PNMPC}$  já calculada, pode-se resolver o problema de otimização quadrática para se calcular a nova variação de controle. Com esta variação de controle, calcula-se a nova ação de controle, que é representada pelo ponto marrom. Nesse ponto, é calculada a tangente da curva, a fim de se obter o ganho da nova matriz  $\mathbf{G}_{PNMPC}$ . Calculadas as matrizes  $\mathbf{G}_{PNMPC}(k-1)$  e  $\mathbf{G}_{PNMPC}(k)$ , é calculada a inclinação média. A partir dessa inclinação média, obtém-se um novo ponto (ponto amarelo), e a tangente nesse ponto é, então, o ganho da matriz  $\overline{\mathbf{G}}_{PNMPC}$ . Este novo ponto se encontra mais próximo do ponto final desejado, ficando mais próximo das referências desejadas.

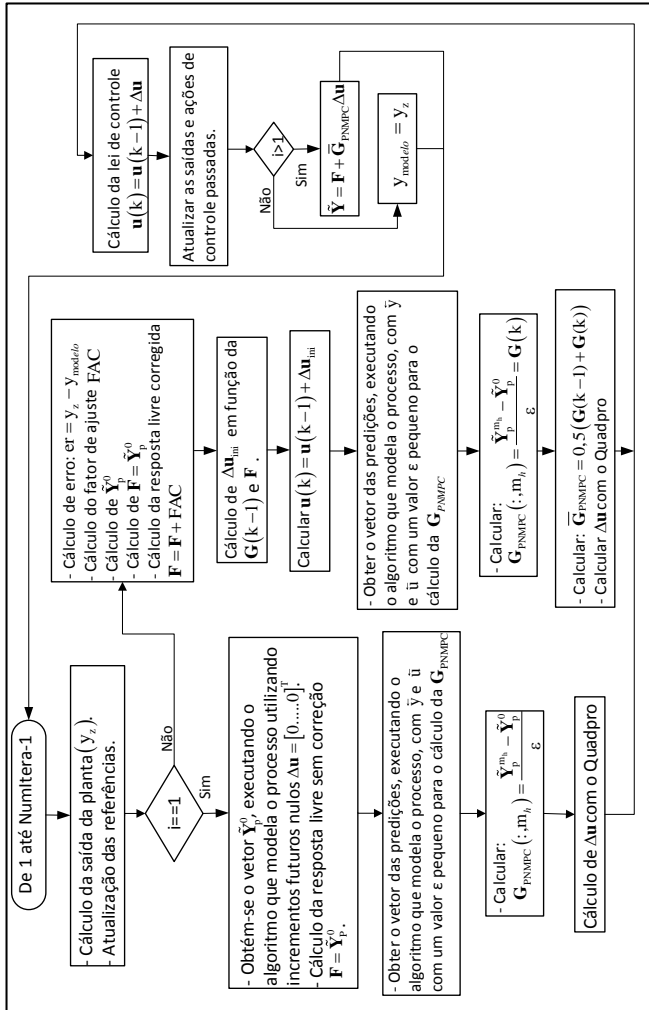
Figura 6 – Interpretação gráfica  $\overline{\mathbf{G}}_{PNMPC}$  calculada em dois passos.



Fonte: Modificado de (PLUCENIO, 2010).

O algoritmo implementado para esta versão é detalhado no fluxograma da Figura 7.

Figura 7 – Fluxograma do algoritmo PNMPC da primeira versão.



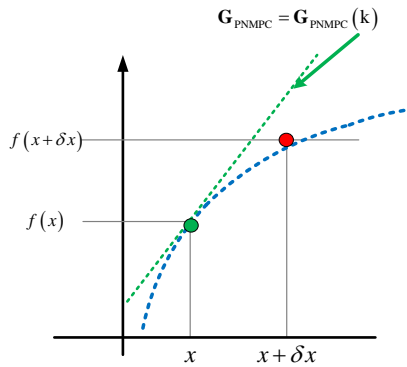
Fonte: Própria.

#### 4.1.2 Segunda versão: PNMPC sem o cálculo de $\Delta \mathbf{u}_{ini}$ .

Uma das vantagens da técnica de controle PNMPC é a sua flexibilidade, já que permite o tratamento de processos tanto lineares como não lineares. No entanto, essa flexibilidade pode acarretar em custos computacionais desnecessários. Assim, pode-se fazer uma análise prévia de processo que se tem a fim de adequar a técnica PNMPC ao tipo de processo ao qual está sendo aplicada. Por exemplo, se existe processo com característica não linear muito importante, é recomendável optar pela técnica PNMPC original completa com  $\Delta \mathbf{u}_{ini}$  (primeira versão). Por outro lado, se o processo apresenta uma não linearidade suave, pode-se optar por reduzir o custo computacional, eliminando o cálculo de  $\Delta \mathbf{u}_{ini}$ .

Com o objetivo de diminuir a carga computacional do algoritmo, propõe-se uma versão de matriz  $\mathbf{G}_{PNMPC}$  calculada em um único passo. Nessa versão, o cálculo do incremento da ação de controle inicial  $\Delta \mathbf{u}_{ini}$  é eliminado. Para isto, utiliza-se o valor da variação do controle da iteração anterior para calcular a ação de controle requerida para obter  $\tilde{\mathbf{Y}}_p^0$ . Assim, é calculada a matriz  $\mathbf{G}_{PNMPC}$  obtida no instante de amostragem anterior, em um único passo, conforme apresentado pela Figura 8.

Figura 8 – Interpretação gráfica  $\mathbf{G}_{PNMPC}$  calculada em um único passo.



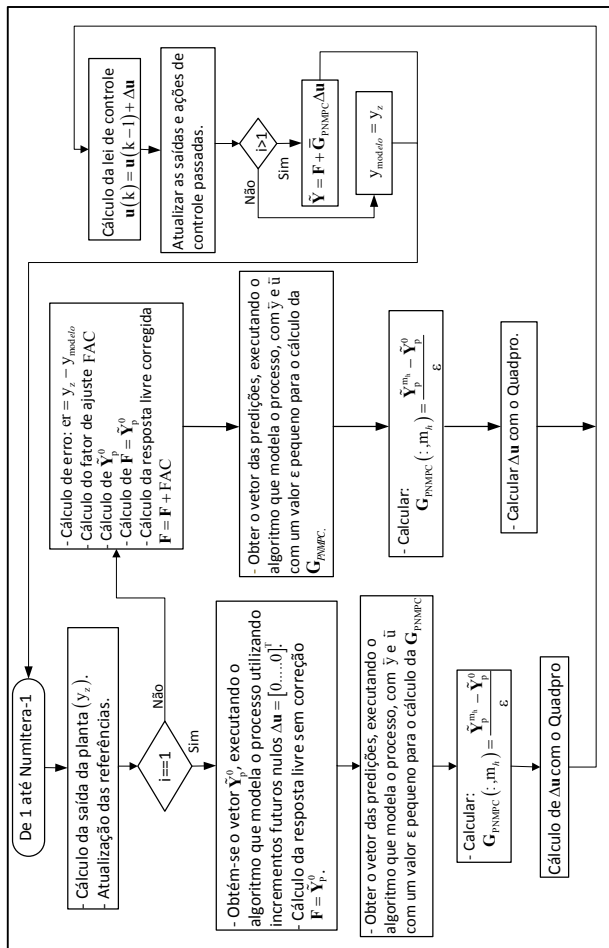
Fonte: Modificado de (PLUCENIO, 2010).

O fato de eliminar o cálculo de  $\Delta \mathbf{u}_{ini}$  diminui a precisão com

a qual o PN MPC calcula a  $\mathbf{G}_{PNMPC}$ , já que  $\Delta u_{ini}$  influencia diretamente no cálculo da  $\hat{\mathbf{Y}}_p^0$ , que está explicitamente envolvida no cálculo da matriz  $\mathbf{G}_{PNMPC}$ .

O algoritmo desta segunda versão é apresentado no fluxograma da Figura 9:

Figura 9 – Fluxograma do algoritmo PN MPC da segunda versão.



Fonte: Própria.



### 4.1.3 Terceira versão: PNMPC obtendo só a primeira coluna da matriz $\mathbf{G}_{PNMPC}$ .

Conforme descrito por Camacho e Bordons (2007), a matriz  $\mathbf{G}$  é a encarregada de guardar os valores da dinâmica do processo e  $\mathbf{G}_{PNMPC}$  é a versão não linear da matriz dinâmica utilizada no MPC prático para sistemas não lineares PNMPC (PLUCENIO, 2010). A matriz  $\mathbf{G}_{PNMPC}$ , também denominada matriz Jacobiana, como apresentado na seção 1.1.1 e 1.1.2. (primeira e segunda versão) respetivamente, são calculados todos os elementos da matriz  $\mathbf{G}_{PNMPC}$  iterativamente, dando como resultado um custo computacional elevado. Como o objetivo de diminuir ainda mais o custo computacional, uma das possibilidades é calcular somente os elementos da primeira coluna da matriz  $\mathbf{G}_{PNMPC}$  pelo procedimento normal do PNMPC e montar a matriz  $\mathbf{G}_{PNMPC}$ , considerando o sistema como invariante no tempo, o que é comumente realizado em sistemas lineares (KEYSER, 1998). Desta forma, o problema de otimização não linear pode ser transformado numa sequência de problemas de otimização linear. Para realizar esse procedimento, é preciso verificar a variação dos elementos da diagonal principal da matriz  $\mathbf{G}_{PNMPC}$ . No caso onde as variações não são significativas, pode-se optar pela técnica utilizada nos sistemas lineares, realizar apenas o cálculo da primeira coluna da matriz  $\mathbf{G}_{PNMPC}$  e deslocá-la até obter uma matriz triangular inferior, conforme equação (4.4).

$$\frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}} = \mathbf{G}_{PNMPC} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{y}}(k+1)}{\partial \Delta \mathbf{u}(k)} & 0 & 0 & 0 \\ \frac{\partial \tilde{\mathbf{y}}(k+2)}{\partial \Delta \mathbf{u}(k)} & \frac{\partial \tilde{\mathbf{y}}(k+1)}{\partial \Delta \mathbf{u}(k)} & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ \frac{\partial \tilde{\mathbf{y}}(k+p)}{\partial \Delta \mathbf{u}(k)} & \frac{\partial \tilde{\mathbf{y}}(k+p-1)}{\partial \Delta \mathbf{u}(k)} & \dots & \frac{\partial \tilde{\mathbf{y}}(k+1)}{\partial \Delta \mathbf{u}(k)} \end{bmatrix} \quad (4.4)$$

No caso de um processo não linear, os elementos das diagonais da matriz  $\mathbf{G}_{PNMPC}$  apresentam variações significativas. Assim, é necessário realizar o cálculo para cada um dos elementos da matriz  $\mathbf{G}_{PNMPC}$ , como é mostrado na equação (4.5). Este procedimento de cálculo é realizado a cada iteração, o que implica num incremento no custo computacional. No entanto, no caso de um processo linear ou

um processo com uma não linearidade suave, os elementos das diagonais da matriz  $\mathbf{G}_{PNMPC}$  não sofrem modificações significativas. Dessa forma, pode-se optar por diminuir o custo computacional, aplicando a terceira versão do PNMPC, onde a matriz  $\mathbf{G}_{PNMPC}$  é calculada com a equação (4.4). Nesta versão, só os elementos da primeira coluna da matriz  $\mathbf{G}_{PNMPC}$  são calculados e esta é deslocada  $m_h$  vezes, até se obter uma matriz triangular inferior, onde  $m_h$  é o horizonte de controle. O fato de só calcular a primeira coluna, faz com que o sistema seja tratado como um sistema onde as relações entre as previsões não lineares e os controles futuros são invariantes no tempo. Se o sistema verifica esta propriedade, o custo computacional para o cálculo de  $\mathbf{G}_{PNMPC}$  diminui razoavelmente.

$$\frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}} = \mathbf{G}_{PNMPC} = \begin{bmatrix} \frac{\partial \tilde{\mathbf{y}}(k+1)}{\partial \Delta \mathbf{u}(k)} & 0 & 0 & 0 \\ \frac{\partial \tilde{\mathbf{y}}(k+2)}{\partial \Delta \mathbf{u}(k)} & \frac{\partial \tilde{\mathbf{y}}(k+2)}{\partial \Delta \mathbf{u}(k+1)} & \dots & 0 \\ \vdots & \vdots & \dots & 0 \\ \frac{\partial \tilde{\mathbf{y}}(k+p)}{\partial \Delta \mathbf{u}(k)} & \frac{\partial \tilde{\mathbf{y}}(k+p)}{\partial \Delta \mathbf{u}(k+1)} & \dots & \frac{\partial \tilde{\mathbf{y}}(k+p)}{\partial \Delta \mathbf{u}(k+m_h-1)} \end{bmatrix} \quad (4.5)$$

#### 4.1.4 Quarta versão: PNMPC obtendo só a primeira coluna da matriz $\mathbf{G}_{PNMPC}$ recalculada de tempos em tempos.

Esta versão é similar à terceira versão, só que agora a matriz  $\mathbf{G}_{PNMPC}$  é recalculada somente de tempos em tempos e não assim a cada iteração.

Esta versão é uma opção muito interessante e flexível. Assim como na versão anterior, esta quarta versão pode ser aplicada em um processo linear ou em um processo com uma não linearidade suave. A diferença entre esta versão e a terceira, é a periodicidade do cálculo da primeira coluna da matriz  $\mathbf{G}_{PNMPC}$ . Nesta versão, este cálculo é realizada a cada  $n$  iterações, e não a cada iteração.

O operador, com conhecimento do processo e do grau de não linearidade que este apresenta, pode ajustar o valor de  $n$  para cada aplicação.

A dificuldade na utilização desta versão se encontra na escolha do

valor de  $n$ . Uma escolha inadequada pode produzir perda de informação de parte da dinâmica do processo. Por outro lado, a vantagem desta versão é a redução do custo computacional, já que a matriz  $\mathbf{G}_{PNMPC}$  é recalculada só de tempos em tempos.

#### 4.1.5 Quinta versão: PNMPC calculando a resposta livre de tempos em tempos, utilizando o DMC-Recursivo.

A quinta versão do PNMPC procura-se a resposta livre de tempos em tempos, utilizando a ideia proposto por Cutler e Remarker (1980)(DMC-Recursivo), para outros instantes de amostragem. A resposta livre é calculado de forma recursiva a cada  $n$  iterações. Para as outras iterações a resposta livre é calculado da forma como PNMPC.

Esta versão procura diminuir o custo computacional em processos onde a não linearidade não é significativa, inserindo na técnica PNMPC, o cálculo da resposta livre como é feito no DMC-Recursivo.

Como apresentado nas versões anteriores, a resposta livre no PNMPC utiliza o modelo não linear para determinar a resposta livre. Dessa forma, obtém-se o vetor  $\tilde{\mathbf{Y}}_p^0$ , de dimensão  $p \times 1$  executando o algoritmo que modela o processo, utilizando incrementos futuros nulos, isto é,  $\Delta \mathbf{u} = [0 \dots 0]^T$ .

Na quinta versão, a resposta livre se torna linear e o cálculo da resposta forçada não muda. A resposta forçada é calculado a cada iteração como apresentados no Capítulo 3, e nas versões 1 e 2 deste Capítulo, o que faz se tornar o controlador linear para as  $n$  iterações.

Conforme apresentado no Capítulo 2, o cálculo da resposta livre considerada no DMC-Recursivo é realizado através da equação:

$$\mathbf{Y}_{livre} = [y_0(t|t-1), \dots, y_0(t+N-1|t-1)]^T \quad (4.6)$$

Ao iniciar a execução do algoritmo DMC- Recursivo no instante  $t$ , é preciso atualizar o vetor da resposta livre, pois o incremento da ação de controle  $\Delta \mathbf{u}(t-1)$  agora é conhecido. O vetor  $\mathbf{Y}_{livre}$  da equação (4.7) é atualizado.

$$\mathbf{Y}_{livre} = \mathbf{Y}_{livre} + \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} \Delta \mathbf{u}(t-1) \quad (4.7)$$

Após obter o novo sinal de controle  $u(t)$ , é preciso deslocar os valores dentro do vetor  $\mathbf{Y}_{livre}$ . Isto porque no instante posterior  $t + 1$ , serão necessárias respostas livres de  $t + 1$  até  $t + N$ . Assim, o novo vetor  $\mathbf{Y}_{livre}$  será dado pela equação (4.8):

$$\mathbf{Y}_{livre} = \begin{bmatrix} y_0(t+1|t) \\ y_0(t+2|t) \\ \vdots \\ y_0(t+N-1|t) \\ y_0(t+N-1|t) \end{bmatrix} \quad (4.8)$$

Dessa forma, o cálculo da resposta livre  $f$  considerada no DMC-Recursivo é realizado através da equação:

$$\mathbf{f} = \begin{bmatrix} y_0(t+1|t) \\ \vdots \\ y_0(t+p|t) \end{bmatrix} + \mathbf{1}_{p \times 1} e \quad (4.9)$$

O erro entre a saída medida e a saída predita em malha aberta no instante  $k$  é usado para corrigir as predições de cada estado de todos os elementos dentro do horizonte de predição  $p$ . Nesse caso, a equação do erro do modelo de predição é:

$$e = y_m(t) - y_0(t|t) \quad (4.10)$$

A representação das predições, considerando o cálculo da resposta livre com o DMC de forma recursiva, é obtida com a equação (4.11):

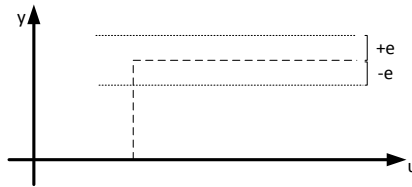
$$\tilde{\mathbf{Y}} = \mathbf{f} + \mathbf{G}\Delta\mathbf{u} \quad (4.11)$$

Utilizando esta forma de implementação, os cálculos necessários para obter a resposta livre se tornam mais simples, não sendo necessário o armazenamento dos incrementos passados de controle. Com este procedimento, só é alterado o modo de calcular a resposta livre, o restante do algoritmo de controle não é modificado.

#### 4.1.6 Sexta versão: PNMPC com reavaliação da matriz $\mathbf{G}_{PNMPC}$ e a resposta livre (DMC-Recursivo), conforme o erro de predição.

A ideia desta versão, é trabalhar com o valor do erro de modelo de predição. Para isso, cria-se uma faixa de trabalho aceitável como se mostra na Figura 10, ou seja, define-se um valor de erro de predição admissível ( $\pm e$ ). Para isso, a cada iteração, é calculado o valor absoluto do erro de predição. Se o valor absoluto do erro de predição se encontra dentro da faixa admissível, o cálculo da matriz  $\mathbf{G}_{PNMPC}$  e da resposta livre não é efetuado. Nesse caso, são usados os valores da iteração anterior para a matriz  $\mathbf{G}_{PNMPC}$  e a resposta livre.

Figura 10 – Faixa de erro de predição admissível.



**Fonte: Própria.**

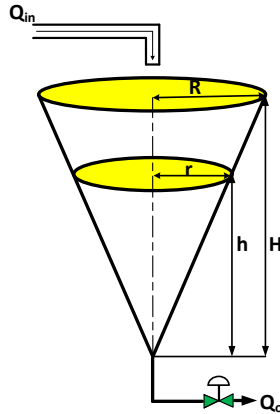
## 4.2 ESTUDO DE CASO: TANQUE CÔNICO

Para demonstrar a aplicação das versões de implementação da técnica PNMPC apresentadas nas seções anteriores, é considerado um processo com dinâmica não linear. A fim de simplificar a demonstração, o controle de nível do tanque cônico (sistema SISO) é usado.

Os tanques comumente encontrados na indústria são de tipo retangular, cilíndrico, cônico ou esférico, podendo ser abertos ou fechados. Nos tanques retangulares, a área transversal é constante em toda faixa de variação do nível. Mas isso não acontece nos tanques de tipo cilíndricos, esféricos ou cônicos, cujas seções transversais mudam com a altura do líquido no interior do tanque, ocasionando mudanças no ganho total do processo. Esta característica é utilizada em testes de sistemas de controle de processos não lineares (MORAES et al., 2013).

Para realizar a modelagem do sistema, considera-se o tanque cônico apresentado pela Figura 11:

Figura 11 – Tanque cônico.



Fonte: Própria.

onde:

- $Q_{in}$  vazão de entrada;
- $Q_o$  vazão da saída;
- $R$  raio máximo do tanque cônico;
- $r$  raio interno (variável) do tanque cônico;
- $H$  altura máxima do tanque cônico;
- $h$  altura interna (variável) do tanque cônico.

Fazendo a análise do balanço de massas, tem-se a seguinte equação:

$$Q_{in} - Q_o = A \frac{dh}{dt} \quad (4.12)$$

onde a área da seção transversal do tanque  $A$  pode ser descrita por:

$$A = \pi \left[ r - (R - r) \left( \frac{h}{H} \right) \right]^2 \quad (4.13)$$

A vazão de saída,  $Q_o$  pode ser expressa pela equação:

$$Q_o = C_v U_c \sqrt{h} \quad (4.14)$$

onde:

$C_v$  coeficiente da capacidade do fluxo da válvula de saída;  
 $U_c$  sinal de controle da válvula de saída.

Substituindo as equações (4.13) e (4.14) em (4.12), tem-se a equação diferencial não linear que descreve o comportamento dinâmico do tanque cônico:

$$\frac{dh}{dt} = \frac{Q_{in} - C_v U_c \sqrt{h}}{\pi [r - (R - r) (\frac{h}{H})]^2} \quad (4.15)$$

Nesse modelo, a vazão de entrada,  $Q_{in}$ , é tratada como uma perturbação ao sistema. Por outro lado, a variável manipulada é o sinal de controle da válvula de saída  $U_c$ , e a variável controlada a altura do tanque cônico  $h$ .

No presente estudo, serão utilizados os dados apresentados na Tabela 1.

Tabela 1 – Parâmetros do tanque cônico.

Parâmetros	Valor
$C_v$	0,04
$H$	5 m
$R$	1 m

As simulações serão realizadas em duas faixas: uma para alturas inferiores a 1 m e a outra para alturas superiores a 1 m. Isso com a finalidade de avaliar o desempenho dos controladores com dinâmicas rápidas e dinâmicas lentas. No primeiro teste, com referências maiores que 1 m para o nível do tanque, o tanque apresenta um comportamento mais lento, levando à escolha de um tempo de amostragem de 10 s. No segundo teste, com referências menores que 1 m para o nível do tanque, a dinâmica é muito mais rápida, levando à adoção de um período de amostragem de 1 s.

A perturbação na vazão de entrada  $Q_{in}$  para referências superiores a 1 m, foi calculada com o sistema em equilíbrio ( $\frac{dh}{dt} = 0$ ) a partir da equação (4.15), obtendo como resultado a equação (4.16)

$$Q_{in} = C_v U_c \sqrt{h_0} \quad (4.16)$$

Considerando uma altura em equilíbrio ( $h_0$ ) de 1 m e o  $U_c = 0,5$  o  $Q_{in}$  usado para referência superiores a 1 m é  $0,02 \text{ m}^3/\text{min}$ . Para referências inferiores a 1 m o  $Q_{in}$  usado é  $0,09 \text{ m}^3/\text{min}$ . Esses valores de perturbações de entrada são aplicados nas simulações para as seis versões.

### 4.3 RESULTADOS

Um dos objetivos deste trabalho é apresentar diversas maneiras de implementar o algoritmo PNMPC. Este algoritmo pode ser escrito para lidar com a não-linearidade em diferentes níveis. Por uma questão de simplicidade, os algoritmos são implementados para controlar o nível de um tanque cônico. A principal finalidade do controlador é manipular a abertura da válvula de saída, a fim de seguir o nível de referência.

No modelo matemático obtido na seção anterior, são avaliados os controladores das versões do PNMPC descritos nas seções 4.1.1 à 4.1.6.

Nesta seção, são exibidos o desempenho e o custo computacional das seis versões do controlador PNMPC. Os resultados dos controladores são comparados e analisados pelo sistema de verificação de resposta, o tempo de execução do algoritmo e o desempenho do controlador.

Como a finalidade de avaliar o custo computacional exigido pela implementação da técnica PNMPC, serão mostrados os tempos obtidos no cálculo do algoritmo para cada versão do PNMPC aplicado ao controle de nível para o tanque cônico.

Todas as simulações foram realizadas no *Matlab*, versão 2014, em um PC com a seguinte configuração: processador A8 de 4 núcleos, AMD de 1,9 GHz; 6 GB de RAM; e sistema operacional Windows 8.

Ainda para conseguir comparar as diferentes versões das estratégias PNMPC, foram comparadas todas as versões através dos índices de desempenho do controlador, o MSE (*mean squared error*) e o IAE (*integral of the absolute magnitude of the error*).

Primeiramente, para obter uma aproximação precisa do modelo, foi utilizado o método de integração através da função *ode45* do pacote computacional do *Matlab*. Esta função é baseada em um algoritmo do tipo Runge-Kutta, que foi desenvolvido a partir do método de Euler.

Por motivos de comparação, os parâmetros do controlador PNMPC são os mesmos para todas as versões. Estes valores são mostrados na



Tabela 2.

Tabela 2 – Parâmetros de sintonia do controlador PNMPC		
Parâmetros	Definição	Valor
$p$	horizonte de predição	12
$m_h$	horizonte de controle	3
$N$	horizonte de modelo	20
$\lambda$	ponderação no controle	3
$\gamma$	ponderação no seguimento da referência	1
$\alpha$	filtro da referência	0,25
$\epsilon$	valor na variação de controle	0,01
$a$	parâmetro no fator de ajuste	0,15

Os processos reais apresentam variações relacionadas a qualquer modelo proposto para representá-lo, e essas variações podem ser dadas por diversas fontes de incertezas. As mais comuns são: ruídos, perturbações não mensuráveis e erros na modelagem do processo. A fim de simular o processo que represente a realidade, foi inserida uma incerteza de 65% nos parâmetros do modelo que representa o tanque cônico. Essa variação pode causar problemas relacionados à estabilidade ou violações de restrições, no caso de controladores preditivos.

Além do seguimento de referência, deseja-se a rejeição de perturbações do tipo constante. Isso é possível através de um modelo de perturbação incorporado no esquema de predição mostrado na Figura 5. No modelo de perturbação adotado originalmente no PNMPC, obtém-se uma estimativa das incertezas futuras através de um observador particular, que realiza um tratamento do erro de predição no instante anterior. Assim, percebe-se que o erro de predição do instante anterior é filtrado para se obter uma correção para as predições futuras.

É importante considerar restrições no sistema, já que todos os processos estão sujeitos a limitações. É imposto a restrição mecânica no atuador, que está limitada pela capacidade mecânica de abertura e fechamento, assim como pela velocidade de resposta que geralmente são considerados nos limites na amplitude do sinal de controle como observado na equação:

$$u_{\min} \leq u(t) \leq u_{\max} \quad \forall t \quad (4.17)$$

A seguir, se apresentam as respostas das simulações para cada versão das implementações do PNMPC.

Para observar a resposta obtida com cada uma das versões mostra-

se o nível do tanque de cada simulação

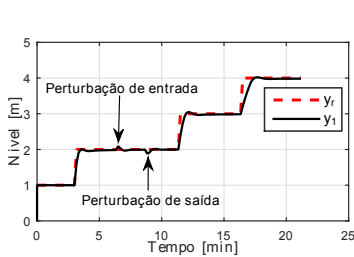
### 4.3.1 Primeira versão: PN MPC original completo com $\Delta u_{ini}$

Como descrito na seção 4.1.1 o primeira versão, o cálculo da matriz  $\mathbf{G}_{PNMPC}$  é realizado em dois passos, capturando a dinâmica do processo com mais precisão. No entanto, o fato de utilizar a resposta livre não linear, linearizar a cada passo e obter uma média dessa resposta faz com que o cálculo apresente maior complexidade e maior custo computacional.

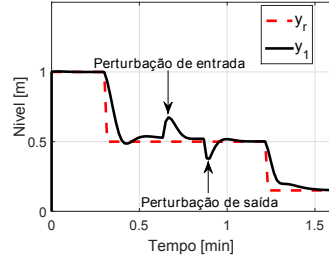
Na Figura 12, mostram-se as respostas das simulações obtidas com a implementação desta versão para referências com alturas superiores a 1 m Figuras (a), (c), (e) e (g) e para alturas inferiores a 1 m Figuras (b), (d), (f) e (h). Nessas figuras  $y_r$  representa a referências do sistema,  $y_1$  é a saída da planta obtida aplicando a primeira versão,  $U_1$  é a ação de controle que representa a abertura da válvula e por último  $Erro_1$  é a diferença entre a saída da planta ( $y_z$ ) e o modelo de predição ( $y_{mod}$ ).

Nas Figuras (a) e (b) mostra-se que ambas simulações conseguem bom seguimento de referências é rejeição à perturbação.

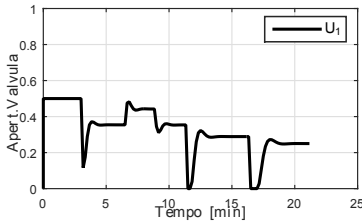
Figura 12 – Simulação da primeira versão do PNMPC.



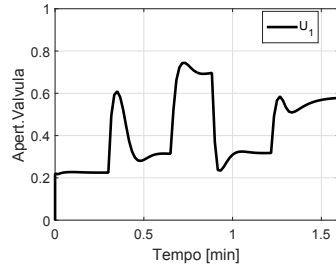
(a) Saída para  $ref > 1m$



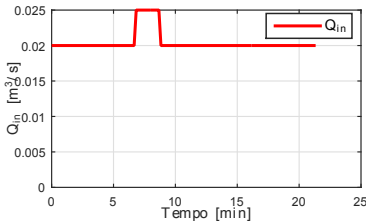
(b) Saída para  $ref < 1m$



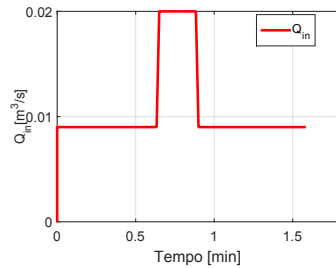
(c) Ação de controle para  $ref > 1m$



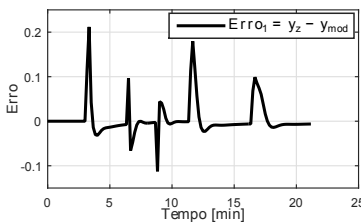
(d) Ação de controle para  $ref < 1m$



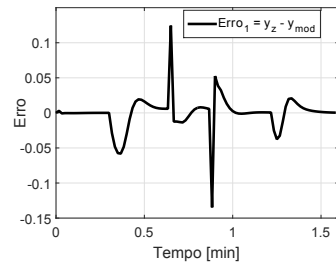
(e) Perturbação para  $ref > 1m$



(f) Perturbação para  $ref < 1m$



(g) Sinais de Erro para  $ref > 1m$



(h) Sinais de Erro para  $ref < 1m$

Fonte: Própria.

Como um dos objetivos deste trabalho é mostrar o tempo de execução, o algoritmo foi rodado 10 vezes. Na Tabela 3, mostra-se o tempo de execução para o algoritmo da primeira versão. Os tempos estão em segundos e são medidos para cada rodada. Na tabela são apresentados o tempo total (T.total) das 10 rodadas e o tempo médio de execução (T. médio), para referências com alturas superiores a 1 m e alturas inferiores a 1 m.

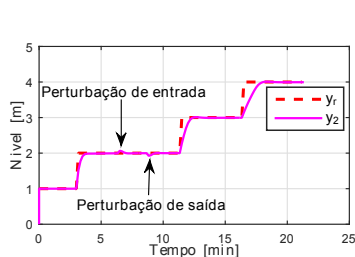
Tabela 3 – Tempo de execução do algoritmo para a primeira versão do PNMPC.

Execução	T. execução (s)	
	$y_r > 1$ m	$y_r < 1$ m
1	40,22	32,39
2	39,58	30,84
3	38,89	30,77
4	38,95	31,26
5	39,18	31,12
6	39,09	32,51
7	39,12	31,27
8	39,15	32,24
9	39,19	32,02
10	38,98	32,50
T. total [s]	392,35	316,92
T. médio [s]	39,23	31,69

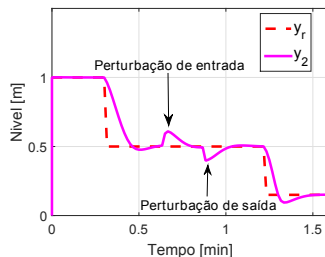
### 4.3.2 Segunda versão: PNMPC sem o cálculo de $\Delta \mathbf{u}_{ini}$

Como foi descrito na seção 4.1.2, nesta versão o cálculo da matriz  $\mathbf{G}_{PNMPC}$  é realizado em um único passo, eliminando o cálculo de  $\Delta \mathbf{u}_{ini}$ . Da mesma forma que na primeira versão, para a segunda versão foram feitos dois testes para avaliar o desempenho do controlador e o tempo de execução. A estrutura e nomenclatura das figuras desta seção é similar à seção 4.3.1 com a diferença que  $y_2$  é a saída da planta obtida aplicando a segunda versão e  $U_2$  é a ação de controle.

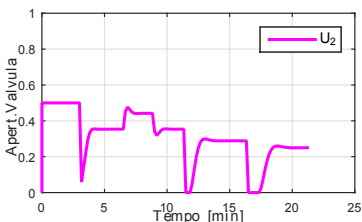
Figura 13 – Simulação da segunda versão do PNMPC.



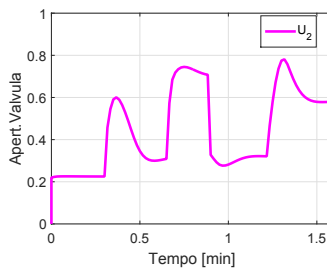
(a) Saída para  $ref > 1m$



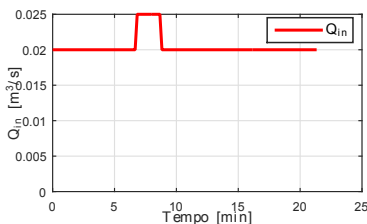
(b) Saída para  $ref < 1m$



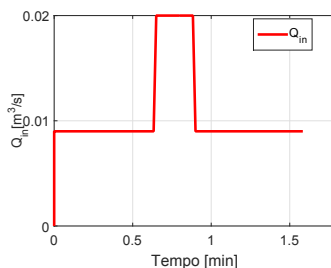
(c) Ação de controle para  $ref > 1m$



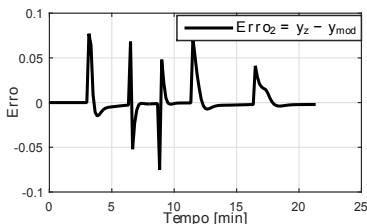
(d) Ação de controle para  $ref < 1m$



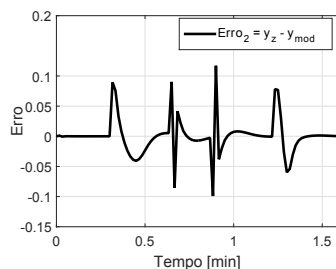
(e) Perturbação para  $ref > 1m$



(f) Perturbação para  $ref < 1m$



(g) Sinais de Erro para  $ref > 1m$



(h) Sinais de Erro para  $ref < 1m$

Fonte: Própria.

Nas Figuras (a) e (b), mostra-se que tanto para referências superiores a 1 m quanto para referências inferiores a 1 m as simulações apresentam bom seguimento de referências e rejeição à perturbação.

Na Tabela 4, são apresentados os tempos de execução obtidos para o algoritmo desta versão.

Tabela 4 – Tempo de execução do algoritmo para a segunda versão do PNMPC.

Execução	T. execução (s)	T. execução (s)
	$y_r > 1$ m	$y_r < 1$ m
1	38,19	31,43
2	38,29	30,45
3	38,43	29,44
4	38,19	29,56
5	38,36	29,46
6	38,51	29,31
7	38,64	29,76
8	38,58	29,90
9	38,75	29,60
10	38,50	29,55
T. Total [s]	384,44	298,46
T. médio [s]	38,44	29,85

### 4.3.3 Terceira versão: PNMPC obtendo só a primeira coluna da matriz $\mathbf{G}_{PNMPC}$ .

Nesta versão o sistema é tratado como se fosse invariante no tempo, mas para isso é preciso verificar o grau de variação que apresentam os elementos das diagonais principais da matriz  $\mathbf{G}_{PNMPC}$ , como foi explicado na seção 4.1.3.

Para verificar se é possível aplicar a terceira versão neste estudo de caso, o primeiro passo é pegar várias matrizes  $\mathbf{G}_{PNMPC}$  em iterações aleatórias, observando o grau da não linearidade do sistema. Na Figura 14, mostram-se os elementos da matriz  $\mathbf{G}_{PNMPC}$  da iteração 25, escolhida aleatoriamente. Nesta matriz, observa-se que o sistema apresenta uma não linearidade suave, já que os elementos das diagonais principais da matriz  $\mathbf{G}_{PNMPC}$  não apresentam variações significativas, característica que foi repetida em matrizes de outras iterações. Dessa

forma é possível aplicar a terceira versão neste estudo de caso.

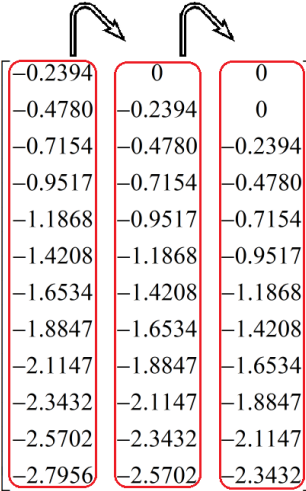
Figura 14 – Matriz  $\mathbf{G}_{PNMPC}$  da iteração 25.

$$\mathbf{G}_{PNMPC} = \begin{bmatrix}
 -0.2394 & 0 & 0 \\
 -0.4780 & -0.2393 & 0 \\
 -0.7154 & -0.4776 & -0.2391 \\
 -0.9517 & -0.7148 & -0.4772 \\
 -1.1868 & -0.9510 & -0.7143 \\
 -1.4208 & -1.1859 & -0.9503 \\
 -1.6534 & -1.4197 & -1.1851 \\
 -1.8847 & -1.6522 & -1.4187 \\
 -2.1147 & -1.8834 & -1.6510 \\
 -2.3432 & -2.1132 & -1.8821 \\
 -2.5702 & -2.3416 & -2.1118 \\
 -2.7956 & -2.5685 & -2.3400
 \end{bmatrix}$$

**Fonte: Própria.**

A seguir, calcula-se a primeira coluna da matriz  $\mathbf{G}_{PNMPC}$ . Então, ela é deslocada até se obter uma matriz triangular inferior, como se mostra na Figura 15. Nesta versão, o sistema é tratado como se fosse invariante no tempo.

Figura 15 – Matriz  $\mathbf{G}_{PNMPC}$  invariante no tempo.



The matrix  $\mathbf{G}_{PNMPC}$  is a 13x3 matrix. The first, second, and third columns are highlighted with red boxes. Two curved arrows point from above to the first and second columns.

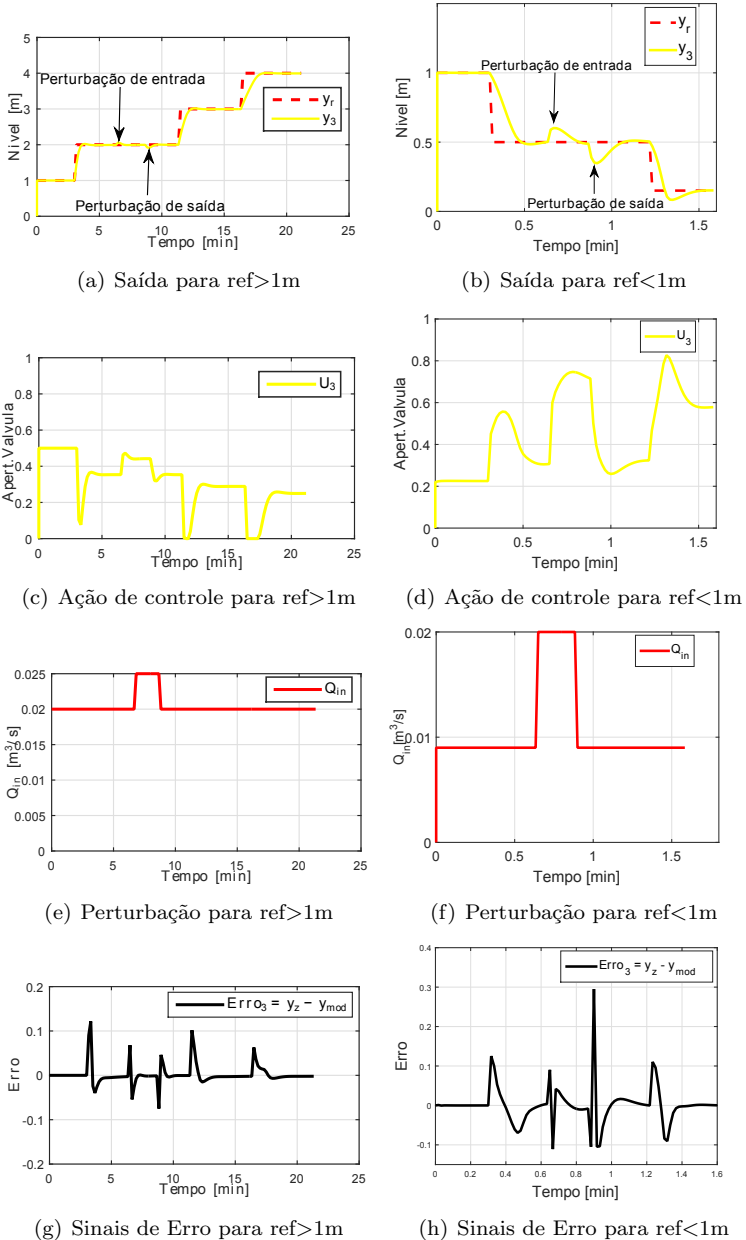
$$\mathbf{G}_{PNMPC} = \begin{bmatrix} -0.2394 & 0 & 0 \\ -0.4780 & -0.2394 & 0 \\ -0.7154 & -0.4780 & -0.2394 \\ -0.9517 & -0.7154 & -0.4780 \\ -1.1868 & -0.9517 & -0.7154 \\ -1.4208 & -1.1868 & -0.9517 \\ -1.6534 & -1.4208 & -1.1868 \\ -1.8847 & -1.6534 & -1.4208 \\ -2.1147 & -1.8847 & -1.6534 \\ -2.3432 & -2.1147 & -1.8847 \\ -2.5702 & -2.3432 & -2.1147 \\ -2.7956 & -2.5702 & -2.3432 \end{bmatrix}$$

**Fonte: Própria.**

Os gráficos das simulações desta versão são apresentados na Figura 16, onde pode-se observar que o desempenho do controlador não foi comprometido, obtendo seguimento de referência e rejeição à perturbação. Na Tabela 5, são apresentados os tempos de execução desta versão.



Figura 16 – Simulação da terceira versão do PNMPC.



Fonte: Própria.

Tabela 5 – Tempo de execução do algoritmo para a terceira versão do PNMPC

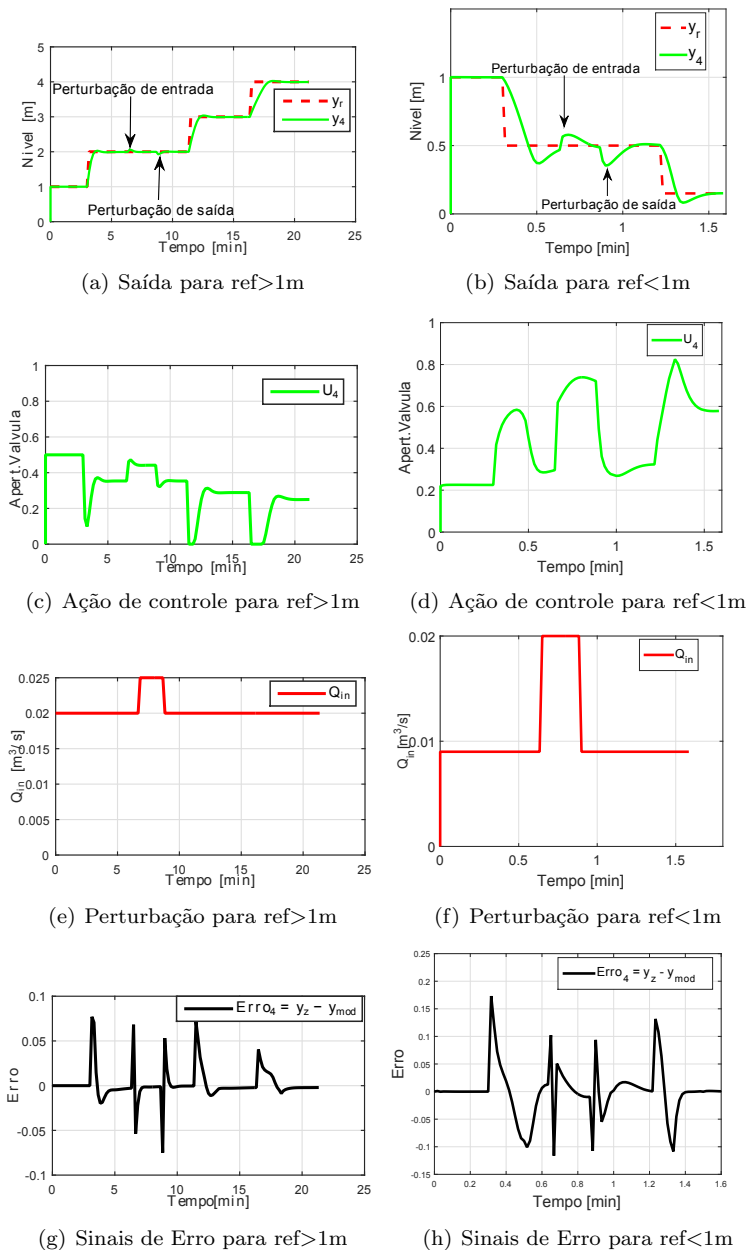
Execução	T. execução (s)	
	$y_r > 1$ m	$y_r < 1$ m
1	40,22	23,62
2	25,66	21,47
3	25,67	20,23
4	25,66	20,46
5	25,88	20,23
6	25,62	20,22
7	25,71	20,25
8	25,77	20,20
9	25,69	20,40
10	25,72	20,31
T. total [s]	257,17	207,39
T. médio [s]	25,72	20,74

#### 4.3.4 Quarta versão: PNMPC obtendo só a primeira coluna da matriz $\mathbf{G}_{PNMPC}$ recalculada de tempos em tempos.

Nesta versão, o cálculo da primeira coluna da matriz  $\mathbf{G}_{PNMPC}$  é realizado de tempos em tempos, ou seja, a cada  $n$  iterações. Neste estudo de caso, considerou-se  $n = 25$ .

Na Figura 17, são apresentadas respostas obtidas com as simulações para esta versão e na Tabela 6, os tempos de execução. Análoga à versão anterior, o desempenho do controlador não foi comprometido, a diferença está no tempo requerido para os cálculos da matriz  $\mathbf{G}_{PNMPC}$ . Como esperado, os tempos de execução são menores que os obtidos com a terceira versão não comprometendo o seguimento da referência e a rejeição da perturbação. No entanto, percebe-se que para referências inferiores a 1 m o controlador apresenta uma maior dificuldade para seguimento das referências em comparação com a terceira versão.

Figura 17 – Simulação da quarta versão do PNMPC.



Fonte: Própria.

Tabela 6 – Tempo de execução do algoritmo para a quarta versão do PNMPC.

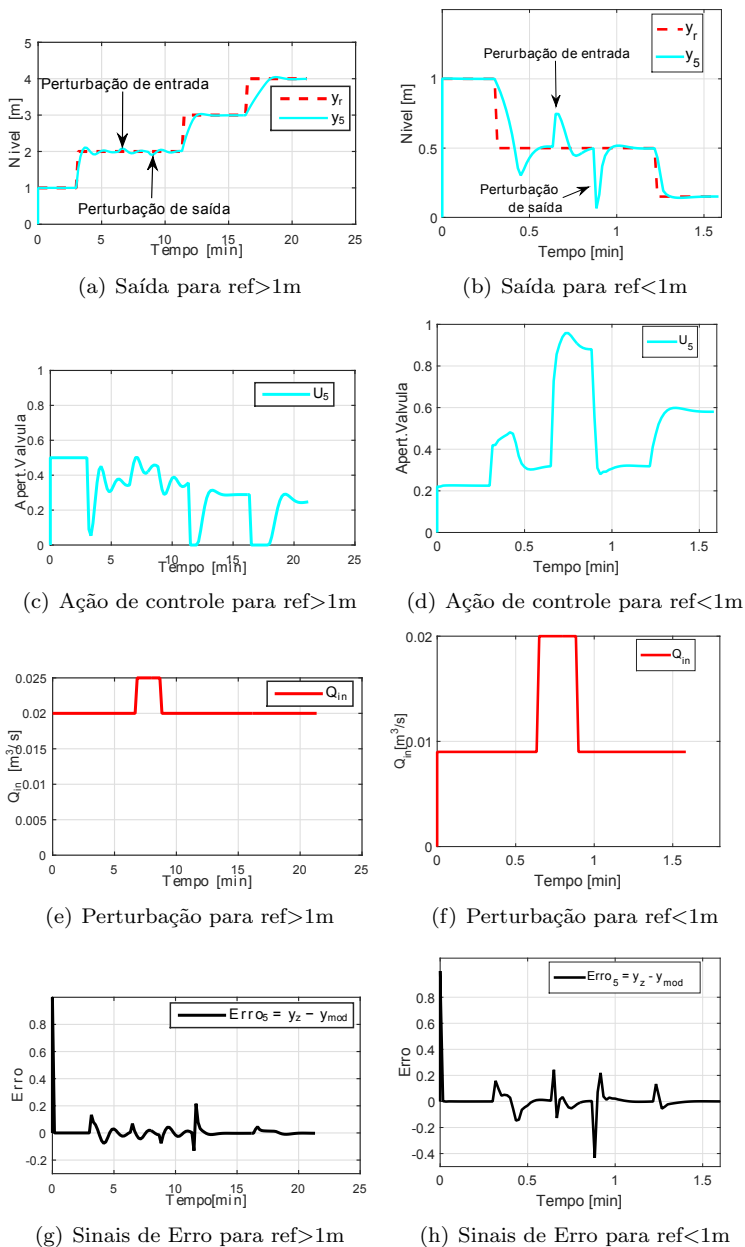
Execução	T. execução (s)	T. execução (s)
	$y_r > 1$ m	$y_r < 1$ m
1	19,37	16,01
2	19,11	15,64
3	19,18	15,60
4	19,16	15,59
5	19,07	15,47
6	19,23	15,59
7	19,21	15,81
8	19,16	15,86
9	19,22	15,84
10	19,07	15,60
T. total [s]	191,80	157,01
T. médio [s]	19,18	15,70

#### 4.3.5 Quinta versão: PNMPC calculando a resposta livre de tempos em tempos, utilizando o DMC-Recursivo.

Nas simulações desta versão, o cálculo da resposta livre é realizado a cada 35 iterações. Os resultados das simulações são apresentados na Figura 18 e na Tabela 7.

Como a resposta livre nesta versão torna-se linear a cada 35 iterações, nota-se que o controlador realiza maior esforço para reagir às perturbações tanto para referências inferiores a 1 m quanto para referências superiores a 1 m.

Figura 18 – Simulação da quinta versão do PNMPC.



Fonte: Própria.

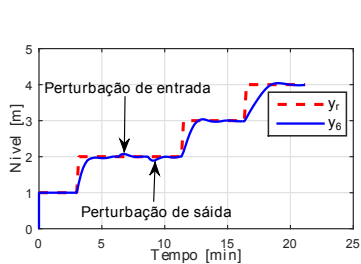
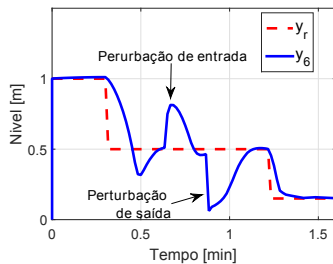
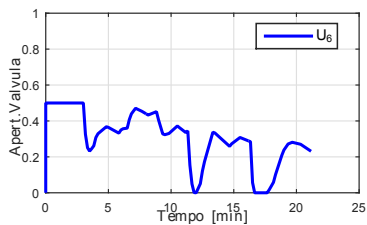
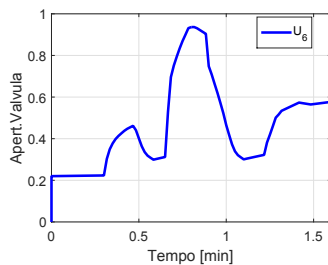
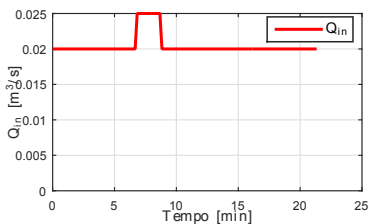
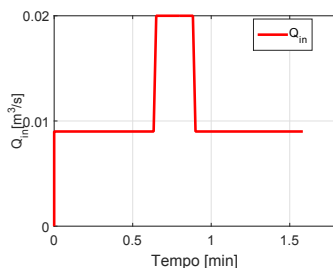
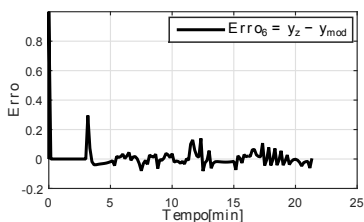
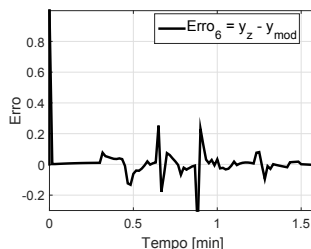
Tabela 7 – Tempo de execução do algoritmo para a quinta versão do PNMPC.

Execução	T. execução (s)	T. execução (s)
	$y_r > 1$ m	$y_r < 1$ m
1	29,44	28,81
2	29,99	28,21
3	30,00	28,02
4	29,23	28,26
5	29,07	28,46
6	29,05	28,37
7	29,02	28,11
8	29,05	28,90
9	29,09	28,83
10	29,49	28,18
T. total [s]	293,43	284,15
T. médio [s]	29,34	28,41

#### 4.3.6 Sexta versão: PNMPC com reavaliação da matriz $\mathbf{G}_{PNMPC}$ e a resposta livre (DMC-Recursivo), conforme o erro de predição.

A sexta versão trabalha com um valor de erro de predição admissível, o erro de predição é o erro existente entre a saída e o modelo. Esta versão foi aplicada no estudo de caso considerando um erro admissível de 0,017, ou seja, a resposta livre de forma recursiva e a matriz  $\mathbf{G}_{PNMPC}$  são calculados quando o erro se encontra fora da faixa admissível de erro. As respostas das simulações para esta versão são apresentadas na Figura 19 e na Tabela 8. Como pode-se observar na figura, o controlador nesta versão, apresenta maior dificuldade na rejeição das perturbações em comparação com a quinta versão.

Figura 19 – Simulação da sexta versão do PNMPC.

(a) Saída para  $ref > 1m$ (b) Saída para  $ref < 1m$ (c) Ação de controle para  $ref > 1m$ (d) Ação de controle para  $ref < 1m$ (e) Perturbação para  $ref > 1m$ (f) Perturbação para  $ref < 1m$ (g) Sinais de Erro para  $ref > 1m$ (h) Sinais de Erro para  $ref < 1m$ 

Fonte: Própria.

Tabela 8 – Tempo de execução do algoritmo para a sexta versão do PNMPC.

Execução	T. execução (s)	T. execução (s)
	$y_r > 1$ m	$y_r < 1$ m
1	27,53	23,40
2	27,70	23,19
3	27,66	24,30
4	27,56	23,03
5	27,45	22,90
6	27,66	23,02
7	27,48	22,82
8	27,70	23,05
9	27,68	22,95
10	27,79	23,00
T. total [s]	276,22	231,66
T. médio [s]	27,62	23,17

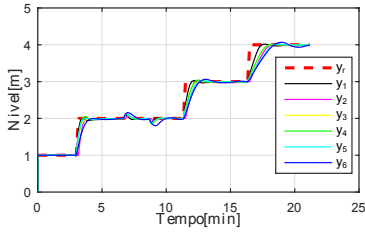
#### 4.4 CONCLUSÃO

Neste capítulo, foram apresentadas seis alternativas de implementar o algoritmo PNMPC. Para avaliar o desempenho dos controladores, estes foram aplicados em um estudo de caso.

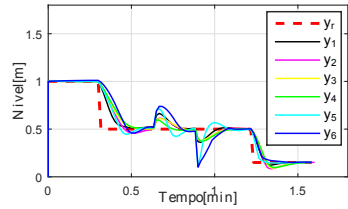
Comparando os resultados obtidos na Figura 20, pode-se observar que os seis algoritmos implementados têm uma resposta muito similar. Apesar de que o algoritmo completo apresenta os melhores resultados e todos conseguem reagir à perturbação adequadamente, a diferença entre eles é o tempo de execução. Isto pode ser atribuído à que a não linearidade do processo é *suave*. Assim, um estudo do comportamento do processo pode levar a uma implementação com ganhos em termos de tempo de processamento sem prejudicar de forma importante o desempenho do controlador. No entanto, percebe-se que na medida em que o modelo representa o processo com menos qualidade a resposta a perturbação fica pior pelo que a escolha entre tempo de execução ou desempenho do controlador depende de cada processo, pode que neste estudo de caso a diferença possa ser aceitável, no entanto, em outro processo a diferença poderia ser impactante.



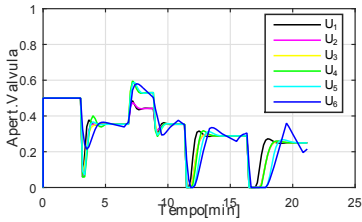
Figura 20 – Comparações das versões do PNMPC: saídas, ações de controle, perturbações e sinais dos erros.



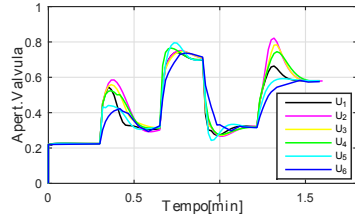
(a) Saídas para  $ref > 1m$



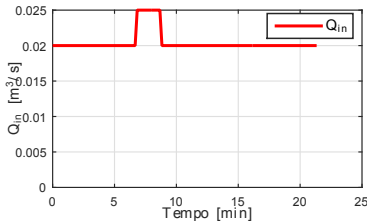
(b) Saídas para  $ref < 1m$



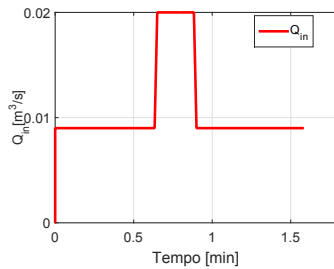
(c) Ações de controle para  $ref > 1m$



(d) Ações de controle para  $ref < 1m$



(e) Perturbação para  $ref > 1m$



(f) Perturbação para  $ref < 1m$

Fonte: Própria.

Como pode-se observar na Tabela 9 e na Tabela 10, considerando as versões 2, 3, 4, 5, 6 é possível obter uma redução do custo computacional se comparado à versão 1, o que era o esperado, já que com cada uma das diferentes abordagens o número de iterações dos ciclos de cálculo, ou seja, da matriz  $\mathbf{G}_{PNMPC}$ , do cálculo da variação de controle ou da resposta livre são reduzidos em comparação com a primeira versão.

Tabela 9 – Tabela comparativa dos tempos de execução para referências superiores a 1 m.

Execução	1ª versão	2ª versão	3ª versão	4ª versão	5ª versão	6ª versão
1	40,22	38,19	25,78	19,37	29,44	27,53
2	39,58	38,29	25,66	19,11	29,99	27,67
3	38,89	38,43	25,67	19,18	30,00	27,66
4	38,95	38,19	25,66	19,16	29,23	27,56
5	39,18	38,36	25,88	19,07	29,07	27,45
6	39,09	38,51	25,62	19,23	29,05	27,66
7	39,12	38,64	25,71	19,21	29,02	27,48
8	39,15	38,58	25,77	19,16	29,05	27,70
9	39,19	38,75	25,69	19,22	29,09	27,68
10	38,98	38,50	25,71	19,07	29,45	27,79
T. total [s]	392,35	384,44	257,17	191,80	293,43	276,22
T. médio [s]	39,23	38,44	25,72	19,18	29,34	27,62

Tabela 10 – Tabela comparativa dos tempos de execução para referências inferiores a 1 m.

Execução	1ª versão	2ª versão	3ª versão	4ª versão	5ª versão	6ª versão
1	32,39	31,43	23,62	16,01	28,81	23,40
2	30,84	30,45	21,47	15,64	28,21	23,19
3	30,77	29,44	20,23	15,60	28,02	24,30
4	31,26	29,56	20,46	15,59	28,26	23,03
5	31,12	29,46	20,23	15,47	28,46	22,90
6	32,51	29,31	20,22	15,59	28,37	23,02
7	31,27	29,76	20,25	15,81	28,11	22,82
8	32,24	29,90	20,20	15,86	28,90	23,05
9	32,02	29,60	20,40	15,84	28,83	22,95
10	32,50	29,55	20,31	15,60	28,18	23,00
T. total [s]	316,92	298,46	207,39	157,01	284,15	231,66
T. médio [s]	31,69	26,85	20,74	15,70	28,41	23,17

Para avaliar o desempenho dos controladores, foram obtidos os índices de desempenho MSE (*Mean Squared Error*) e IAE (*Integral of the Absolute Magnitude of the Error*) para todas as versões do PN MPC para valores de referências superiores a 1 m. Os índices de desempenho

são mostrados na Tabela 11. O que era esperado pode-se confirmar na Tabela 11, a versão que apresentou melhor desempenho é a primeira, a qual possui a resposta livre não linear, a matriz  $\mathbf{G}_{PNMPC}$  calculada em dois passos e fazendo uma média ( $\tilde{\mathbf{G}}_{PNMPC}$ ) a cada iteração, faz com que o sistema seja tratado com maior precisão.

Tabela 11 – Índices de desempenho do controlador das seis versões do PNMPC para referências superiores a 1 m.

I. desempenho	1ª versão	2ª versão	3ª versão	4ª versão	5ª versão	6ª versão
MSE	0,0031	0,0040	0,0063	0,0085	0,0100	0,0096
IAE	2,58	3,40	4,56	4,77	5,89	5,07



## Capítulo 5

---

# CONTROLE PREDITIVO BASEADO EM MODELO COM PASSO DE ESTABILIDADE

---

Neste capítulo, apresenta-se uma outra estratégia para lidar com erros nos ganhos, as constantes de tempos e principalmente erros de atrasos de transporte nos modelos. A retroalimentação é capaz de lidar com erros nos ganhos e nas constantes de tempo. No entanto, os erros no atraso de transporte podem causar oscilações nas variáveis do processo e, em casos extremos, são capazes de instabilizar a planta, mesmo para sistemas estáveis em malha aberta. Uma alternativa para o tratamento dos erros de predição é a aplicação de filtros no erro de predição, com o objetivo de melhorar a estabilidade do sistema sob controle e oferecer robustez na planta frente aos erros de modelo. Esse conceito foi desenvolvido a partir das ideias do Preditor de Smith Filtrado, apresentado em Normey e Camacho (2007) e mostrado na Figura 21.

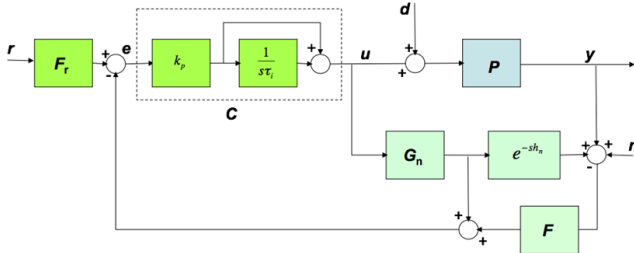
Da Figura 21 se obtém a função de transferência em malha fechada entre a saída e a perturbação é dada pela expressão:

$$\frac{Y(s)}{d(s)} = P(s) \left( 1 - \frac{C(s)P(s)F(s)}{1 + C(s)G_n(s)} \right) \quad (5.1)$$

onde:

$C(s)$	controlador;
$P(s)$	planta real;
$F(s)$	filtro;
$G_n(s)$	modelo;
$Y(s)$	saída da planta;
$d(s)$	perturbação.

Figura 21 – Preditor de Smith Filtrado.



**Fonte:** (NORMEY; CAMACHO, 2007).

O projeto do filtro  $F(S)$  é uma possibilidade de introduzir robustez e rejeição à perturbação, entretanto, não se pode obter robustez e rejeição à perturbação simultaneamente (PLUCENIO et al., 2014).

Um outro trabalho aplicando o filtro no erro de predição, é proposto em Lima, Normey e Plucenio (2014), onde é utilizada a estratégia DMC-Recursivo apresentada no capítulo 2 na seção 2.2.3. É possível filtrar o erro de predição separadamente e adicioná-lo à predição de malha aberta  $y_o$ :

$$\tilde{y}(t+k|t) = \sum_{i=1}^k g_i \Delta(t+k-i) + y_o(t+k|t) + F(z^{-1})(y_m(t) - y_o(t|t)) \quad (5.2)$$

No qual o erro do modelo é:

$$e = y_m(t) - y_o(t|t) \quad (5.3)$$

O efeito do filtro é análogo ao método utilizado no algoritmo DTC-GPC descrito em Normey e Camacho (2007). A idéia de aplicar um filtro no erro de predição para corrigir as predições pode ser aplicada a outros algoritmos MPC, como GPC, DMC e PN MPC, dentre as técnicas de controle preditivo não linear proposto por Plucenio (2010).

Controladores onde o filtro é projetado para robustez apresentam um comportamento lento, não conseguindo rejeitar as perturbações. Uma maneira de mostrar isso é utilizando a expressão para o cálculo da ação de controle nos algoritmos de controle MPCs. Utilizando uma expressão clássica do MPC para o algoritmo DMC (CAMACHO; BOR-

DONS, 2007), pode-se escrever o vetor de incremento de ação de controle  $\Delta \mathbf{u}$  como:

$$\begin{aligned} \Delta \mathbf{u} &= \mathbf{K} (\mathbf{W} - \mathbf{F}) \\ \text{com} \\ \mathbf{K} &= (\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R} \\ \mathbf{F} &= \mathbf{F}_{sc} + \mathbf{E}_f \end{aligned} \quad (5.4)$$

Onde

- $\mathbf{W}$  vetor das referências;
- $\mathbf{G}$  matriz dinâmica ou Jacobiana;
- $\mathbf{R}$  matriz de ponderação do seguimento da referência;
- $\mathbf{Q}$  matriz de ponderação no controle;
- $\mathbf{F}_{sc}$  vetor da resposta livre sem correção;
- $\mathbf{E}_f$  vetor do erro de predição filtrado.

O Erro de predição filtrado pode ser escrito como:

$$\mathbf{E}_f(k) = \beta \mathbf{E}_f(k-1) + (1-\beta) \mathbf{E}(k) \quad (5.5)$$

Onde escolhendo  $\beta$  entre 0 e 1 ( $0 \leq \beta < 1$ ), obtém-se o filtro discreto projetado para garantir estabilidade em casos de erros no modelo de predição para um caso SISO. O erro  $\mathbf{E}(k)$  é a diferença entre o vetor com a variável controlada medida  $\mathbf{Y}(k)$  e o vetor com a variável controlada predita  $\tilde{\mathbf{Y}}(k)$ , mostrado na equação (5.6).

$$\mathbf{E}(k) = \mathbf{Y}(k) - \tilde{\mathbf{Y}}(k) \quad (5.6)$$

Substituindo as equações (5.5) e (5.6) nas expressões da equação (5.4), a ação de controle pode ser escrita como:

$$\Delta \mathbf{u} = \mathbf{K} \left( \mathbf{W} - \mathbf{F}_{sc} - \beta \mathbf{E}_f(k-1) - (1-\beta) \left( \mathbf{Y}(k) - \tilde{\mathbf{Y}}(k) \right) \right) \quad (5.7)$$

Considerando que o sistema atingiu o regime permanente, temos que:

$$\begin{aligned} \mathbf{W} &= \mathbf{Y}(k-1) \\ \mathbf{Y}(k-1) &= \mathbf{F}_{sc} + \mathbf{E} \\ \mathbf{F}_{sc} &= \mathbf{Y}(k-1) - \mathbf{E} \end{aligned} \quad (5.8)$$

Substituindo as expressões da equação (5.8) em (5.7) se tem a

equação (5.9):

$$\Delta \mathbf{u} = \mathbf{K} \left( (1 - \beta) \mathbf{E} - (1 - \beta) \left( \mathbf{Y}(k) - \tilde{\mathbf{Y}}(k) \right) \right) \quad (5.9)$$

Normalmente,  $\mathbf{Y}(k) - \tilde{\mathbf{Y}}(k)$  seria igual a  $\mathbf{E}$  no instante  $k$  em regime permanente, o que levaria a  $\Delta \mathbf{u} = 0$ . Entretanto, supondo que no instante  $k$  ingressa uma perturbação não modelada  $\mathbf{d}(k)$ . Nesse caso tem-se:

$$\begin{aligned} \Delta \mathbf{u} &= \mathbf{K} \left( (1 - \beta) \mathbf{E} - (1 - \beta) \left( \mathbf{Y}(k) + \mathbf{d}(k) - \tilde{\mathbf{Y}}(k) \right) \right) \\ \Delta \mathbf{u} &= \mathbf{K} \left( (1 - \beta) \mathbf{E} - (1 - \beta) (\mathbf{E} + \mathbf{d}(k)) \right) \\ \Delta \mathbf{u} &= \mathbf{K} \left( (1 - \beta) \mathbf{E} - (1 - \beta) \mathbf{E} - (1 - \beta) \mathbf{d}(k) \right) \\ \Delta \mathbf{u} &= -\mathbf{K} (1 - \beta) \mathbf{d}(k) \end{aligned} \quad (5.10)$$

A ação de controle para atuar na perturbação  $\mathbf{d}(k)$  é calculada a partir de  $(1 - \beta) \mathbf{d}(k)$ . Isto torna a resposta da rejeição à perturbação extremamente lenta quando  $\beta$  é grande. Entretanto, isto ocorre quando se tem grandes erros no atraso de transporte utilizado no modelo de predição.

A aplicação de um filtro no erro de predição é uma alternativa que visa modificar o erro de predição, melhorando a robustez frente aos erros do modelo. Esta abordagem tem seus benefícios, mas, um problema inevitável é que tem-se que conhecer a *priori* o erro aproximado do atraso de transporte para obter o melhor benefício com a técnica de filtragem. Além disso, muita filtragem ( $\beta$  grande) no sistema pode resultar numa resposta mais lenta a perturbações.

Com base no resultado da análise do comportamento do filtro frente a perturbações, apresenta-se neste capítulo o método proposto por Plucenio et al. (2014), para o tratamento de erros de predição em plantas estáveis em malha aberta, utilizando o segundo método de estabilidade de Lyapunov. Deriva-se um sinal que é somado à predição, com o objetivo de fornecer estabilidade e bom desempenho do controlador frente a erros do modelo e perturbações inseridas no sistema.

Neste trabalho, apresenta-se a comparação do método aplicando o filtro no erro de predição e o método proposto por Plucenio et al. (2014). Nesta técnica, nenhum filtro é utilizado. Em vez disso, o algoritmo MPC com passo de estabilidade propõe tratar o problema em duas etapas:

1. Etapa de estabilidade; e
2. Etapa de cálculo da ação de controle.



Na etapa de estabilização, um sinal é calculado e adicionado ao vetor do erro de predição, estabilizando a planta frente a erros de modelagem e perturbações não modeladas inseridas no sistema. Na etapa de cálculo da ação de controle, é aplicado o erro modificado sobre o cálculo do sinal de controle.

Esse capítulo está organizado da seguinte forma: na seção 5.1 apresenta-se os conceitos fundamentais que serão necessários para a compreensão da metodologia de cálculo desta técnica. Para ilustrar melhor a aplicação desta técnica é mostrado um exemplo de simulação para um caso SISO. Na seção 5.2 serão apresentadas as simulações obtidas aplicando o DMC-Recursivo com a técnica de filtragem no erro de predição e com a técnica do passo de estabilidade, ambas aplicadas num estudo de caso. O reator químico com duas variáveis manipuladas e duas variáveis controladas, comumente visto nas indústrias petroquímicas. Para demonstrar a eficácia desta técnica, erros de modelagens e perturbações não modeladas serão inseridos no sistema. Finalmente, na seção 5.3 serão apresentadas as respectivas conclusões deste capítulo.

## 5.1 PASSO DE ESTABILIDADE PARA COMPENSAÇÃO DE ERROS NO MODELO DE PREDIÇÃO

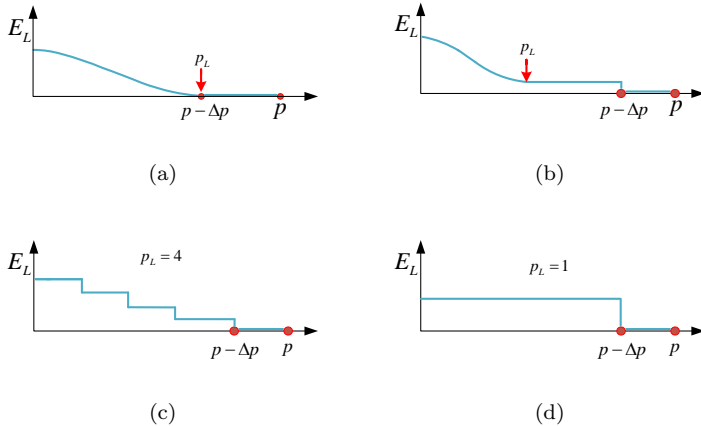
O passo de estabilidade é uma técnica que procura um sinal que somado, ao erro verdadeiro, proporcione robustez e estabilidade em plantas estáveis em malha aberta com erros de modelo. Esta técnica propõe uma solução onde a correção do erro de predição  $\mathbf{E}_L$  é somado ao vetor de erro verdadeiro  $\mathbf{E}$  proporcionando melhor desempenho frente a perturbações não modeladas, .

Uma primeira opção para obter  $\mathbf{E}_L$  é utilizar  $p$  possíveis diferentes elementos. No entanto, isso pode trazer um elevado custo computacional. Assim, apresentam-se outras opções para o projeto do vetor  $\mathbf{E}_L$ , conforme:

- a) aplicando um horizonte de correção de erro  $p_L = p - \Delta p$  e  $\Delta p$  elementos nulos, como se mostra na Figura 22 (a);
- b) aplicando um horizonte de correção de erro  $p_L$  e repetindo o último elemento do horizonte para todos os  $p - p_L$  elementos seguintes, como se mostra na Figura 22 (b);
- c) usando o conceito de *blocking*, onde  $\mathbf{E}_L$  é preenchido com  $n_L$  grupo de diferentes valores possíveis, como se mostra na Figura 22 (c);

- d) preenchendo o vetor de tamanho  $p - \Delta p$  com um único valor de  $\mathbf{E}_L$  e  $\Delta p$  elementos nulos, como se mostra na Figura 22 (d).

Figura 22 – Formas de implementar o vetor  $E_L$ .



Fonte: (PLUCENIO et al., 2014).

Para evitar que o requisito de estabilidade perturbe o seguimento da referência quando se tem  $\mathbf{E}_L$  definido em todos os elementos  $p$ , é importante ter alguns dos últimos componentes de  $\mathbf{E}_L$  nulos. Dessa forma, faz-se com que a correção total convirja para aquela utilizada pelo DMC-Recursivo no final do horizonte de previsão.

Qualquer que seja a opção escolhida dentre as mencionadas anteriormente, o vetor final de estabilização da correção do erro é expresso por:

$$\mathbf{E}_s = \mathbf{E} + \mathbf{M}\mathbf{E}_L \quad (5.11)$$

onde:

$\mathbf{E}_L$ : é o vetor com todas correções do erro de Lyapunov das variáveis controladas  $[E_L^1 \ E_L^2 \ \dots \ E_L^p]^T$ ;

$\mathbf{E}$ : é o vetor com todos os erros verdadeiros das variáveis controladas  $[E^1 \ E^2 \ \dots \ E^p]^T$ ;

$\mathbf{E}_s$ : é o vetor com todos os erros modificados;

**M**: é uma matriz com dimensão  $p \times p$ . É composto de uns e zeros, de tal maneira que implementa as estratégias mostradas na Figura 22.

Para a obtenção do vetor  $\mathbf{E}_L$ , aplica-se o segundo método de Lyapunov para estabilidade. Esse método baseia-se na analogia de uma função de Lyapunov  $V(x)$  com energia ou potencial. Se  $x = 0$  é um ponto de equilíbrio, então se existe um  $V(x)$  tal que:

$$\begin{aligned} V(x) &> 0 && \text{para qualquer } x \neq 0 && \text{e} \\ V(x) &= 0 && \text{só para } x = 0 \end{aligned} \quad (5.12)$$

Então, se:

$$\frac{dV(x)}{dt} \leq 0 \quad (5.13)$$

$x = 0$  é um ponto de equilíbrio estável no sentido de Lyapunov. Se  $\frac{dV(x)}{dt} \leq 0$  e  $\frac{dV(x)}{dt} = 0$  somente quando  $x = 0$  o sistema é assintoticamente estável. Este é um resultado muito bem aceito e a base para muitos estudos de estabilidade (MONTEIRO, 2006).

Para plantas estáveis em malha aberta se tem que o sistema está em equilíbrio quando  $\Delta \mathbf{u} = 0$ . A instabilidade pode ser causada pela ação de controle devido a erros do modelo de predição, é natural considerar como uma função de Lyapunov:

$$L = \frac{1}{2} \Delta \mathbf{u}^T \Delta \mathbf{u} \quad (5.14)$$

Observe que  $L$  é uma função Lyapunov candidata, válida desde que  $\Delta \mathbf{u} = 0$  seja a condição de equilíbrio:

$$\begin{aligned} L(0) &= 0 && \text{e} \\ L &> 0 && \text{para qualquer } \Delta \mathbf{u} \neq 0 \end{aligned} \quad (5.15)$$

Considerando um caso irrestrito, onde:

$$\Delta \mathbf{u} = \mathbf{K}(\mathbf{W} - \mathbf{F})$$

$$\mathbf{K} = (\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R} \quad (5.16)$$

$$\mathbf{F} = \mathbf{F}_{sc} + \mathbf{E} + \mathbf{M} \mathbf{E}_L$$

Na equação (5.16),  $\mathbf{W}$  representa o vetor de trajetórias desejadas dentro do horizonte de predição,  $\mathbf{F}$  é a resposta livre corrigida no horizonte de predição,  $\mathbf{R}$  e  $\mathbf{Q}$  são as matrizes de ponderação do *set-point*

e de supressão de movimento,  $\mathbf{F}_{sc}$  é a resposta livre sem correção ao longo do horizonte de predição  $p$ ,  $\mathbf{E}$  é o vetor do erro de predição no instante atual,  $\mathbf{E}_L$  é o vetor do erro de estabilidade de Lyapunov, que é calculado para corrigir o erro de predição.

A fim de assegurar estabilidade, deve-se cumprir  $\frac{\partial L}{\partial t} \leq 0$ . Em que  $\frac{\partial L}{\partial t}$  pode ser calculado como:

$$\frac{\partial L}{\partial t} = \frac{\partial L}{\partial \Delta \mathbf{u}} \frac{\partial \Delta \mathbf{u}}{\partial \mathbf{E}_L} \frac{\partial \mathbf{E}_L}{\partial t} \quad (5.17)$$

ou

$$\frac{\partial L}{\partial t} = \Delta \mathbf{u}^T (-\mathbf{KM}) \frac{\partial \mathbf{E}_L}{\partial t} \quad (5.18)$$

Fazendo substituições:

$$\frac{\partial L}{\partial t} = -(\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{ME}_L)^T \mathbf{K}^T \mathbf{KM} \frac{\partial \mathbf{E}_L}{\partial t} \quad (5.19)$$

O termo  $\frac{\partial \mathbf{E}_L}{\partial t}$  pode ser reescrita como  $\mathbf{E}_L^k - \mathbf{E}_L^{k-1}$ .

Assim, a expressão final para a equação (5.19) torna-se:

$$\begin{aligned} \frac{\partial L}{\partial t} &= \frac{1}{2} \mathbf{E}_L^T \Phi_L \mathbf{E}_L - \frac{1}{2} \Theta_L^T \mathbf{E}_L + \frac{1}{2} \mathbf{D} \\ \Phi_L &= \mathbf{M}^T \mathbf{K}^T \mathbf{KM} \\ \mathbf{A} &= \mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} \\ \mathbf{B} &= \mathbf{M}^T \mathbf{K}^T \mathbf{K} \\ \Theta_L &= \Phi_L \mathbf{E}_L^{k-1} + \mathbf{BA} \\ \mathbf{D} &= \mathbf{A}^T \mathbf{B}^T \mathbf{E}_L^{k-1} \end{aligned} \quad (5.20)$$

Tendo a expressão para  $\frac{\partial L}{\partial t}$ , pode-se formular o seguinte problema de otimização:

$$\begin{aligned} \min_{\mathbf{E}_L} J \\ \text{sujeito a:} \end{aligned} \quad (5.21)$$

$$J = \frac{1}{2} \mathbf{E}_L^T \Phi_L \mathbf{E}_L - \frac{1}{2} \Theta_L^T \mathbf{E}_L + \frac{1}{2} f_{PD}(\mathbf{E}_L)$$

Os dois primeiros termos da equação (5.21) representam a ex-

pressão  $\frac{\partial L}{\partial t}$ . A idéia é que o valor encontrado de  $\mathbf{E}_L$  obrigue o termo  $\frac{\partial L}{\partial t} \leq 0$ . Porém, isto deve ter um limite, e essa é a razão da presença do termo  $f_{DP}(\mathbf{E}_L)$ . Uma função positiva definida sobre  $\mathbf{E}_L$  deve ser concebida com o objetivo de se contrapor ao termo negativo de  $J$ . Além disso, a função deve incluir ponderações para ajustar uma melhor sintonia do controlador. A primeira função candidata é:

$$\mathbf{E}_L^T \mathbf{M}^T \mathbf{S}_L \mathbf{M} \mathbf{E}_L \quad (5.22)$$

Os valores da diagonal da matriz de ponderação  $\mathbf{S}_L$  atuam com o objetivo de suprimir  $\mathbf{E}_L$ , a fim de evitar que o valor  $\Delta L$  seja negativo. No entanto, o papel desta função é limitado a suprimir os valores negativos em  $J$  e não contribui para qualquer outro atributo do sistema sob controle.

Outra função candidata definida positiva é:

$$\left( \tilde{\mathbf{Y}} - \mathbf{W} \right)^T \mathbf{S}_L \left( \tilde{\mathbf{Y}} - \mathbf{W} \right) \quad (5.23)$$

onde

$$\tilde{\mathbf{Y}} = \mathbf{F}_{sc} + \mathbf{E} + \mathbf{M} \mathbf{E}_L + \mathbf{G} \mathbf{K} (\mathbf{W} - (\mathbf{F}_{sc} + \mathbf{E} + \mathbf{M} \mathbf{E}_L)) \quad (5.24)$$

O objetivo desta função é suprimir os valores negativos de  $J$ , estabelecendo uma relação entre o principal objetivo, que é evitar que  $\frac{\partial L}{\partial t}$  seja negativo, garantindo o seguimento da referência. Os elementos positivos da matriz diagonal  $\mathbf{S}_L$  são ajustados a fim de obter  $\mathbf{E}_L$  que proporcione estabilidade e seguimento da referência para o sistema sob controle.

Finalmente, o problema de otimização quadrática para obter  $\mathbf{E}_L$

é formulado como:

$$\begin{aligned}
 & \underset{\mathbf{E}_L}{\min} J \\
 & \text{sujeito a :} \\
 & J = \frac{1}{2} \mathbf{E}_L^T \Phi_L \mathbf{E}_L - \frac{1}{2} \Theta_L^T \mathbf{E}_L \\
 & \Phi_L = \mathbf{M}^T (\mathbf{K}^T \mathbf{K} + \mathbf{Z}^T \mathbf{S}_L \mathbf{Z}) \mathbf{M} \\
 & \mathbf{Z} = \mathbf{G} \mathbf{K} - \mathbf{I}_{p \times p} \\
 & \Theta_L^T = \Phi_L \mathbf{E}_L^{\mathbf{k}-1} + \mathbf{B} \mathbf{A} + \mathbf{M}^T \mathbf{Z}^T \mathbf{S}_L \mathbf{Z} \mathbf{A}
 \end{aligned} \tag{5.25}$$

Para o caso sem restrições, o valor do vetor  $\mathbf{E}_L$  que minimiza  $J$  é:

$$\mathbf{E}_L = 0,5 \Phi_L^{-1} \Theta_L \tag{5.26}$$

A matriz  $\Phi_L$  será definida positiva, desde que a matriz de ponderação diagonal  $\mathbf{S}_L$  seja escolhida com valores positivos.

Um aspecto interessante desta metodologia é que restrições em  $\Delta \mathbf{u}$  e  $\mathbf{u}$  podem ser atendidas na etapa de estabilidade. Introduzindo restrições no sistema, os seguintes termos se podem escrever como:

$$\begin{aligned}
 \Delta \mathbf{u} \leq \mathbf{B} \quad \text{é equivalente a :} \\
 \mathbf{K} (\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{M} \mathbf{E}_L) \leq \mathbf{B}
 \end{aligned} \tag{5.27}$$

ou :

$$-\mathbf{K} \mathbf{M} \mathbf{E}_L \leq \mathbf{B} - \mathbf{K} (\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E})$$

O vetor  $\mathbf{u}$  pode ser escrito como:  $\mathbf{u} = \mathbf{u}_{k-1} \mathbf{1}_{m_h \times 1} + \mathbf{T}_I \Delta \mathbf{u}$ , onde  $\mathbf{T}_I$  é a matriz Toeplitz integradora, e  $\mathbf{1}_{m_h \times 1}$  é um vetor com cada componente igual a 1 para um caso SISO como é mostrado a seguir:

$\mathbf{u} \leq \mathbf{C}$  é equivalente a :

$$\mathbf{u}_{k-1} \mathbf{1}_{m_h \times 1} + \mathbf{T}_I \mathbf{K} (\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{M}\mathbf{E}_L) \leq \mathbf{C}$$

ou : (5.28)

$$\mathbf{T}_I \mathbf{K} \mathbf{M} \mathbf{E}_L \leq \mathbf{C} - \mathbf{u}_{k-1} \mathbf{1}_{m_h \times 1} - \mathbf{T}_I \mathbf{K} \mathbf{A}$$

$$\mathbf{A} = \mathbf{W} - \mathbf{F}_{sc} - \mathbf{E}$$

Ao introduzir estas restrições na programação quadrática, dada pela equação (5.25), o valor de  $\mathbf{E}_L$  obtido pode ser usado na equação (5.29) para derivar  $\Delta \mathbf{u}$ .

$$\Delta \mathbf{u} = \mathbf{K} (\mathbf{W} - \mathbf{F}) \tag{5.29}$$

$$\mathbf{F} = \mathbf{F}_{sc} + \mathbf{E} + \mathbf{M}\mathbf{E}_L$$

O vetor  $\Delta \mathbf{u}$  obtido proverá estabilidade ao sistema respeitando as restrições impostas em  $\Delta \mathbf{u}$  e  $\mathbf{u}$ .

Para ilustrar a aplicação desta técnica e analisar o caso de erros de modelagem no atraso, é empregado um exemplo apresentado em Normey e Camacho (2007) e mostrado na Figura 23. No exemplo tem-se um sistema de aquecimento de água onde  $\mathbf{w}$  é a resistência encarregada de aquecer a água dentro do tanque e na saída do tanque tem-se uma tubulação com um sensor de temperatura instalado a uma certa distância do tanque. Esta distância gera um atraso de transporte  $L$  medido em minutos, sendo que  $L = 6 \text{ min}$ .

A função de transferência do sistema de primeira ordem com atraso é apresentada na equação (5.30).

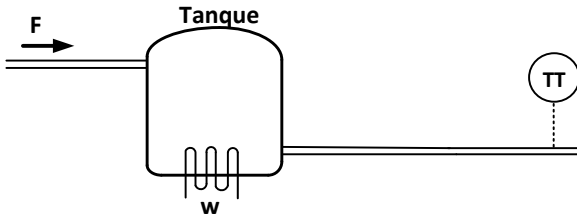
$$P_n(s) = \frac{e^{-Ls}}{2s + 1} \tag{5.30}$$

Levando em conta o atraso e a dinâmica do processo, considera-se um tempo de amostragem  $Ts = 0,2 \text{ min}$ . Dessa forma, a função de transferência discreta é mostrada na equação (5.31).

$$P_n(z) = \frac{0,095z^{-d_n}}{z - 0,905} \tag{5.31}$$

Para controlar esse processo, parametriza-se um controlador DMC-

Figura 23 – Esquema do tanque de aquecimento de água.



**Fonte:** Modificado de (NORMEY; CAMACHO, 2007).

Recursivo com horizonte de predição  $p = 80$ , horizonte de controle  $m_h = 10$ , horizonte de modelo  $N = 100$ , ponderação no seguimento da referência  $\gamma = 0,01$  e ponderação no esforço de controle  $\lambda = 1$ .

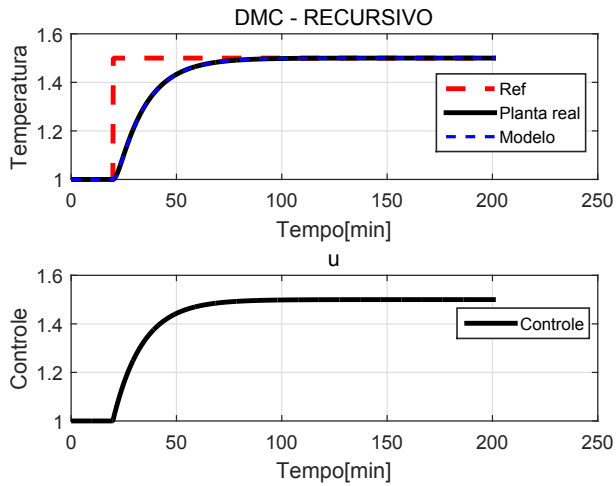
Inicialmente implementou-se o DMC-Recursivo para controlar o tanque de aquecimento de água considerando  $d_n = 0$  tanto na planta real quanto no modelo, onde o modelo é a saída predita, como se mostra na Figura 24.

Em seguida é implementado o DMC-Recursivo considerando  $d_n = 30$  tanto na planta quanto no modelo, como se mostra na Figura 25.

Os dados das saídas preditas ( $y_{pred}$ ) da Figura 24 e da Figura 25 foram armazenadas e apresentadas na Figura 26. Nessa figura percebe-se a diferença existente entre a saída predita sem o atraso e a saída predita com o atraso. Com esses dados, foi calculada a diferença entre ambas saídas ( $diff = y_{pred}$  sem atraso  $- y_{pred}$  com atraso) mostrada na Figura 26. O  $diff$ , é o sinal com a qual pode-se corrigir a diferença entre a predição sem o atraso e a predição com o atraso, sendo que é o sinal com o qual poderia-se corrigir erros de modelagens no atraso.

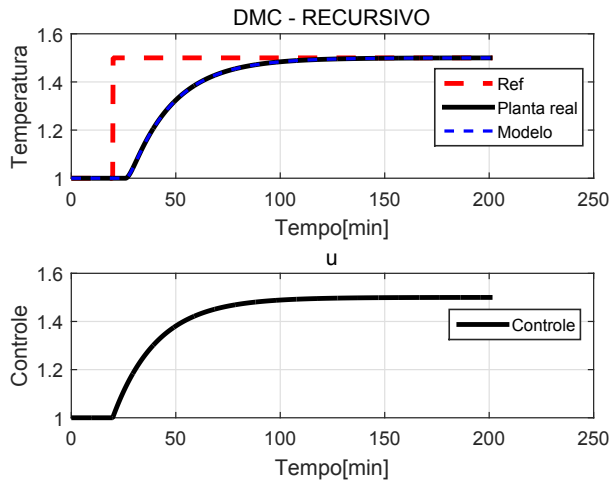


Figura 24 – Simulação do sistema sem atraso e sem incertezas no modelo.



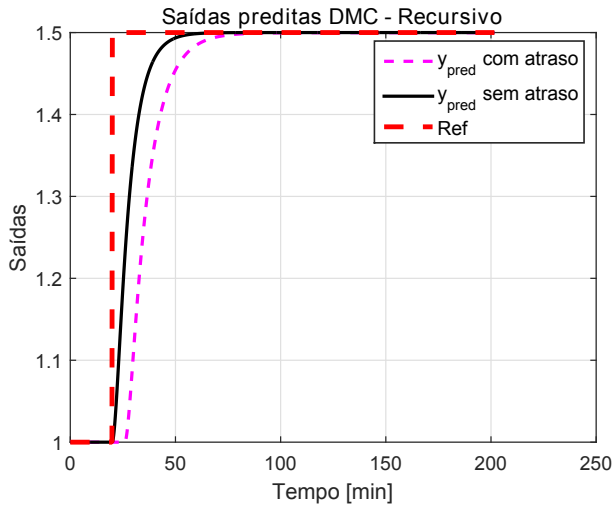
Fonte: própria.

Figura 25 – Simulação do sistema com atraso e sem incertezas no modelo.



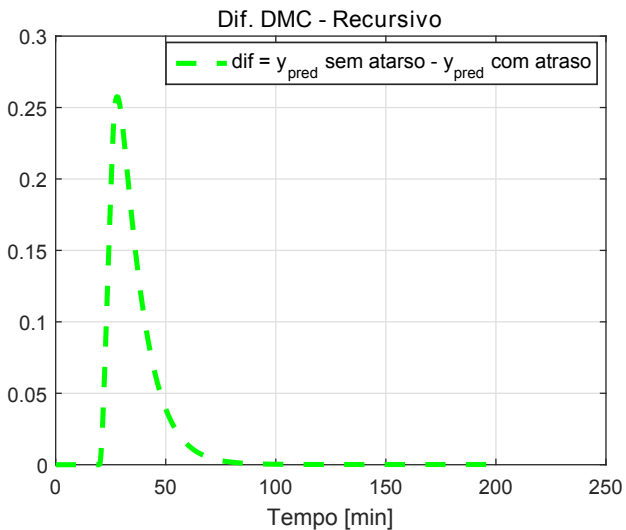
Fonte: própria.

Figura 26 – Saídas preditas com o atraso e sem o atraso.



Fonte: própria.

Figura 27 – Diferença entre as saídas predita com o atraso e sem o atraso.

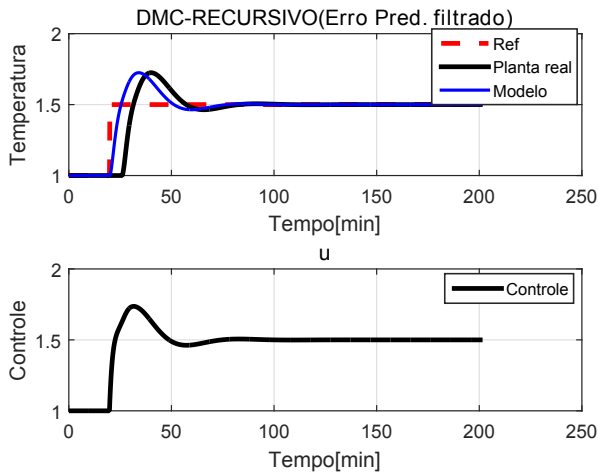


Fonte: própria.

A seguir, são aplicadas as técnicas DMC-Recursivo com filtro no erro de predição e DMC-Recursivo com passo de estabilidade. Nas simulações foi considerado que a planta real não possui atraso e que o modelo possui um atraso de 30.

Na Figura 28, mostra-se a resposta obtida na simulação aplicando a técnica DMC-Recursivo com filtro no erro de predição. Nessa figura, apresenta-se a saída da planta, a saída do modelo e o sinal de controle. Os parâmetros de sintonia usados nesse controlador são: horizonte de predição  $p = 80$ , horizonte de controle  $m_h = 10$ , horizonte de modelo  $N = 100$ , ponderação no seguimento da referência  $\gamma = 0,001$ , ponderação no esforço de controle  $\lambda = 50$  e filtro no erro de predição  $\beta = 0,95$ .

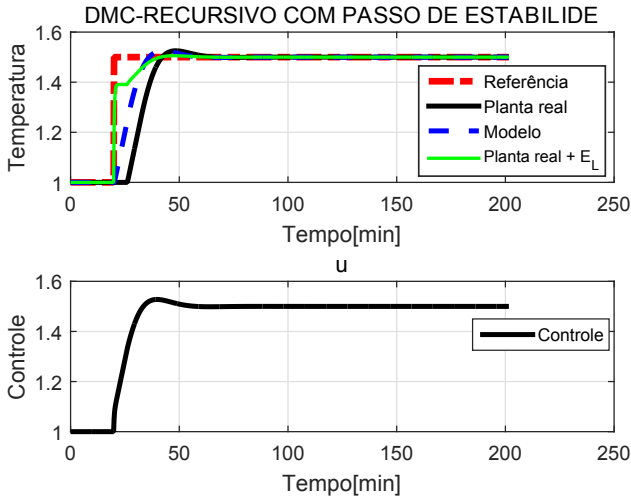
Figura 28 – DMC-Recursivo com filtro no erro de predição aplicando uma incerteza de 6 min ( $d_n = 30$ ).



Fonte: própria.

Na Figura 29, mostra-se a resposta obtida aplicando a técnica DMC-Recursivo com passo de estabilidade. Nessa figura, apresenta-se a saída da planta, a saída do modelo e o sinal de controle. Os parâmetros de sintonia usados nesse controlador são: horizonte de predição  $p = 80$ , horizonte de controle  $m_h = 10$ , horizonte de modelo  $N = 100$ , ponderação no seguimento da referência  $\gamma = 0,01$ , ponderação no esforço de controle  $\lambda = 40$ , ponderação de supressão para  $E_L, P_L = 3800$  e ponderação de supressão para  $E_L, S_L = 1$ .

Figura 29 – DMC-Recursivo com passo de estabilidade aplicando uma incerteza de 6 min ( $d_n = 30$ ).



Fonte: própria.

Na Figura 28 e na Figura 29, observa-se que as duas técnicas conseguem seguimento de referência frente a incerteza inserida no modelo. No entanto, pode-se perceber que a técnica do DMC-Recursivo com passo de estabilidade consegue alcançar o regime permanente com maior rapidez comparada com a técnica DMC-Recursivo com filtro no erro de predição.

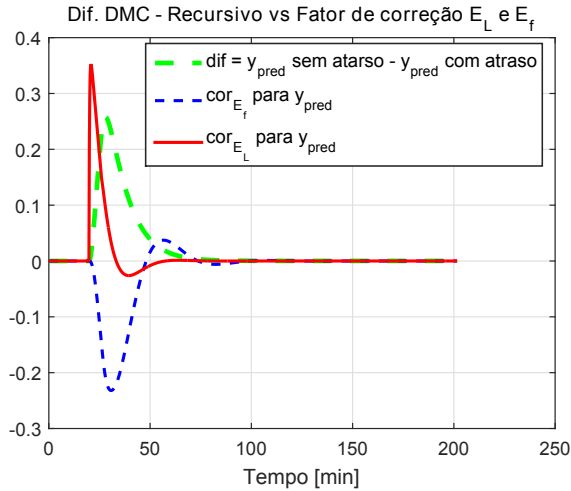
Por último, na Figura 30 são comparados os fatores de correção do DMC-Recursivo com filtro no erro de predição ( $E_f$ ) e do DMC-Recursivo com passo de estabilidade ( $E_L$ ) com o fator de correção obtido com o DMC-Recursivo ( $dif$ ). Nessa figura pode-se observar que a sinal que mais se assemelha ao  $dif$  é o fator de correção obtido aplicando a técnica passo de estabilidade.

Como normalmente o erro verdadeiro é somado à predição, a correção é obtida fazendo a diferença entre a planta real e o modelo de predição apresentado na equação (5.32)

$$\mathbf{E}(k) = \mathbf{Y}(k) - \tilde{\mathbf{Y}}(k) \quad (5.32)$$

Na realidade, o que acontece ao aplicar a técnica com passo de estabilidade é que ela calcula uma correção adicional  $\mathbf{E}_L$ , que somada ao

Figura 30 – Fatores de correção do DMC-Recursivo com passo de estabilidade, DMC-Recursivo com filtro no erro de predição e *dif*.



Fonte: própria.

erro de predição verdadeiro  $\mathbf{E}$  fornece estabilidade no sistema. Assim, pode-se imaginar que a técnica passo de estabilidade “modifica” a saída da planta  $\mathbf{Y}'(k)$  como se mostra na equação (5.33).

$$\mathbf{Y}'(k) = \mathbf{Y}(k) + \mathbf{E}_L(k) \quad (5.33)$$

A resposta desta modificação é representada com a linha verde na Figura 29. Se não houvesse  $\mathbf{E}_L$  o sistema iria oscilar, pois ao tentar realizar a correção aplicando o erro verdadeiro o sistema aplicaria uma ação de controle maior achando que não atingiu a referência.  $\mathbf{E}_L$  “engana” o algoritmo ao fazer com que a planta se pareça com o modelo de predição sem atraso. Considerando as equações (5.32) e (5.33), tem-se:

$$\begin{aligned} \mathbf{Y}'(k) - \tilde{\mathbf{Y}}(k) &= \mathbf{Y}(k) + \mathbf{E}_L - \tilde{\mathbf{Y}}(k) \\ \mathbf{Y}'(k) - \tilde{\mathbf{Y}}(k) &= \mathbf{E}_L(k) + \mathbf{E}(k) \end{aligned}$$

Pelo que, o erro que vai ser somado à predição não é só o erro verdadeiro, agora é a soma do erro verdadeiro como o  $\mathbf{E}_L$ , como mostrado na equação (5.34).

$$\mathbf{Y}'(k) = \tilde{\mathbf{Y}}(k) + \mathbf{E}_L(k) + \mathbf{E}(k) \quad (5.34)$$

## 5.2 ESTUDO DE CASO

Para demonstrar a eficácia da estratégia aplicando o passo de estabilidade proposto nesse trabalho, considera-se o controle de um reator de tanque agitado, comumente encontrado na indústria petroquímica.

Nesta seção, os resultados de simulação são apresentados. As comparações da técnica de filtragem e da técnica aplicando o passo de estabilidade são implementadas no controlador com estratégia DMC-Recursivo. As respostas das simulações são comparadas utilizando o índice de desempenho do controlador e pela verificação da resposta de simulação aplicados no reator de tanque agitado.

Dois controladores DMCs-Recursivos são configurados. O primeiro controlador aplicando a técnica de filtragem no erro de predição mostrado na Tabela 12, e o outro controlador aplicando a técnica passo de estabilidade mostrado na Tabela 13. Ambos implementados no reator de tanque agitado apresentado em Normey e Camacho (2007), e mostrado na Figura 31.

Tabela 12 – Parâmetros de sintonia do controlador DMC recursivo com filtragem no erro de predição

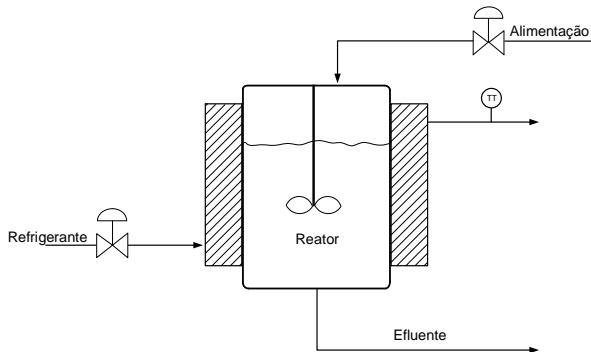
Parâmetros	Definição	Valor
$p_1$	horizonte de predição para $y_1(k)$	50
$p_2$	horizonte de predição para $y_2(k)$	40
$N_1$	horizonte de modelo para $y_1(k)$	70
$N_2$	horizonte de modelo para $y_2(k)$	70
$m_{h_1}$	horizonte de controle para $u_1(k)$	10
$m_{h_2}$	horizonte de controle para $u_2(k)$	10
$\lambda_1$	ponderação no controle	0,2
$\lambda_2$	ponderação no controle	1
$\gamma_1$	ponderação no seguimento da referência	0,01
$\gamma_2$	ponderação no seguimento da referência	0,1
$\beta_1$	filtro no erro de predição	0,75
$\beta_2$	filtro no erro de predição	0,75

O modelo é descrito na seguinte matriz de transferência (as constantes de tempo são expressos em minutos).

Tabela 13 – Parâmetros de sintonia do controlador DMC-Recursivo com passo de estabilidade.

Parâmetros	Definição	Valor
$p_1$	horizonte de predição para $y_1(t)$	50
$p_2$	horizonte de predição para $y_2(t)$	40
$N_1$	horizonte de modelo para $y_1(t)$	70
$N_2$	horizonte de modelo para $y_2(t)$	70
$m_{h_1}$	horizonte de controle para $u_1(t)$	10
$m_{h_2}$	horizonte de controle para $u_2(t)$	10
$\lambda_1$	ponderação no controle	0,01
$\lambda_2$	ponderação no controle	2
$\gamma_1$	ponderação no seguimento da referência	0,01
$\gamma_2$	ponderação no seguimento da referência	0,1
$P_{L_1}$	Ponderação de supressão para $E_L$	0,1
$P_{L_2}$	Ponderação de supressão para $E_L$	0,1
$S_{L_1}$	Ponderação de supressão para $E_L$	1000
$S_{L_2}$	Ponderação de supressão para $E_L$	1000

Figura 31 – Esquema do Reator de tanque agitado.



Fonte: (NORMEY; CAMACHO, 2007)

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{e^{-0,24s}}{1+0,7s} & \frac{5e^{-0,24s}}{1+0,3s} \\ \frac{e^{-0,09s}}{1+0,5s} & \frac{2e^{-0,09s}}{1+0,4s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix} \quad (5.35)$$

As variáveis manipuladas  $U_1(s)$  e  $U_2(s)$  da equação (5.35) representam na Figura 31 a vazão de alimentação e a vazão do fluido refrigerante no revestimento. As variáveis controladas  $Y_1(s)$  e  $Y_2(s)$  são a concentração do efluente e a temperatura do reator, respectivamente. Como observado na matriz (5.35), existe influência entre as variáveis controladas e manipuladas, já que existe acoplamento no sistema.

Utilizando um tempo de amostragem  $T_s = 0,03$  min, o modelo discretizado é descrito pela expressão (5.36).

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \frac{0,0420z^{-1}}{1-0,9580z^{-1}}z^{-8} & \frac{0,4758z^{-1}}{1-0,9048z^{-1}}z^{-8} \\ \frac{0,0582z^{-1}}{1-0,9418z^{-1}}z^{-3} & \frac{0,1445z^{-1}}{1-0,9277z^{-1}}z^{-3} \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} \quad (5.36)$$

Os tempos mortos neste caso são causadas por efeitos de medição. O tempo morto na temperatura de saída  $y_2(k)$  é o tempo necessário para transportar o fluido para o sensor. O tempo morto na concentração  $y_1(k)$  é causada pelo efeito de transporte e o tempo requerido pelo analisador. Os tempos mortos mínimos são:  $d_1 = 8$  (para as funções de transferência da saída 1, FT11 e FT12) e  $d_2 = 3$  (para as funções de transferência da saída 2, FT21 e FT22).

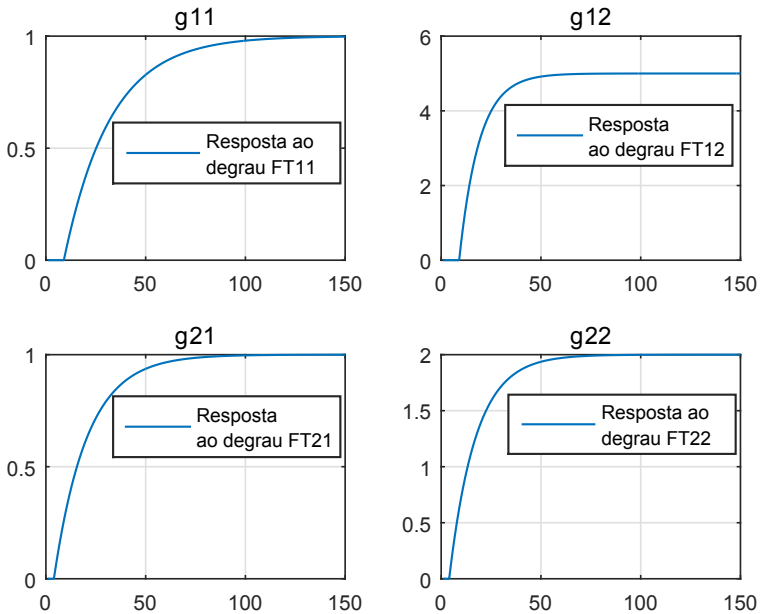
Implementando a técnica do passo de estabilidade notou-se que considerando só a matriz de ponderação  $\mathbf{S}_L$ , o valor desta tinha que ser muito pequeno, já que  $\Phi_L$  da equação (5.25) resultava em valores muito pequenos. Então, foi considerada a incorporação de uma outra matriz de ponderação denominada  $\mathbf{P}_L$ , em  $\Phi_L$  e  $\mathbf{B}$ , como se mostra na equação (5.37):

$$\begin{aligned} \mathbf{B} &= \mathbf{M}^T \mathbf{K}^T \mathbf{P}_L \mathbf{K} \\ \Phi_L &= \mathbf{M}^T (\mathbf{K}^T \mathbf{P}_L \mathbf{K} + \mathbf{Z}^T \mathbf{S}_L \mathbf{Z}) \mathbf{M} \end{aligned} \quad (5.37)$$



A Figura 32 mostra as respostas ao degrau com seus respectivos atrasos de transporte para cada função de transferência. Onde,  $g_{11}$  é a resposta ao degrau da função de transferência que relaciona  $y_1$  com  $u_1$ ,  $g_{12}$  é a resposta ao degrau da função de transferência que relaciona  $y_1$  com  $u_2$ ,  $g_{21}$  é a resposta ao degrau da função de transferência que relaciona  $y_2$  com  $u_1$  e  $g_{22}$  é a resposta ao degrau da função de transferência que relaciona  $y_2$  com  $u_2$ ,

Figura 32 – Resposta ao degrau das funções de transferência.

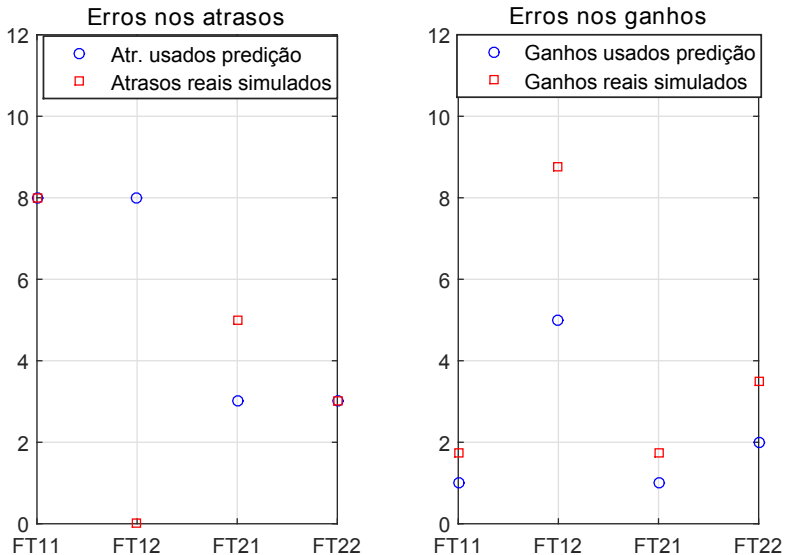


**Fonte: própria.**

Na Figura 33, mostra-se as incertezas entre a planta real e o modelo de predição aplicadas a cada uma das funções de transferência para o atraso de transporte e o ganho. Na figura da esquerda são apresentados os atrasos de transporte utilizados no modelo de predição e os utilizados para simular a planta real, à direita, mostram-se os ganhos da planta real e do modelo de predição. Como observado nessa figura, a distância existente entre o quadrado e o círculo representa a incerteza inserida no sistema. As incertezas no atraso de transporte são: para FT11 0%, FT12 100%, FT21 80% e FT22 0%. A incerteza aplicada nos ganhos para cada uma das funções de transferência é de

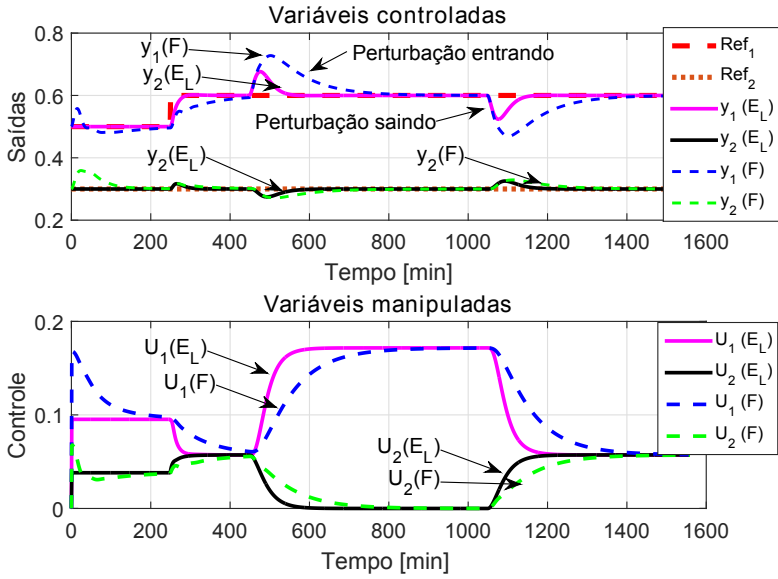
75%.

Figura 33 – Incertezas ineridas no atraso de transporte e no ganho.

**Fonte: própria.**

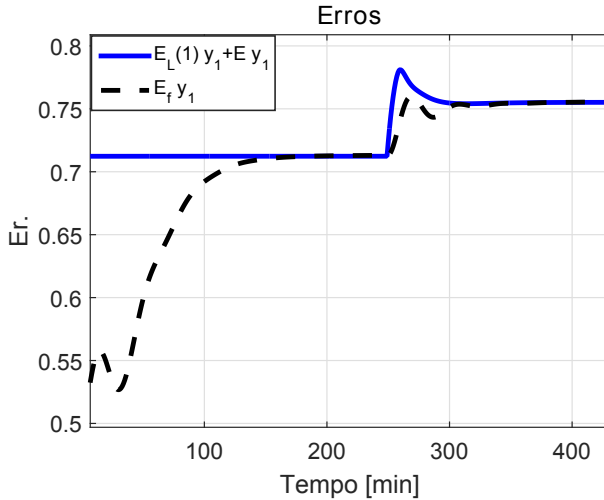
Na Figura 34 apresentam-se os resultados obtidos com a aplicação da técnica passo de estabilidade, com erros entre a planta real e o modelo de predição conforme mostrados na Figura 33. Na Figura 34, uma perturbação não modelada é introduzida no minuto 450, mantendo-se até o minuto 1050. Observa-se que aplicando a técnica com o passo de estabilidade obtém-se um melhor seguimento de referência, comparado com a utilização de filtragem no erro de predição, quando existem erros de modelo tanto para o ganho como para o atraso de transporte. Por outro lado, a resposta na rejeição da perturbação é mais rápida aplicando o passo de estabilidade, já que o uso de muito filtragem no sistema causa lentidão na rejeição da perturbação conforme já mostrado na introdução do presente capítulo.

Figura 34 – Resultados de simulação



Na Figura 35 apresenta uma comparação dos erros de predição obtidos com a aplicação de filtro e com a técnica passo de estabilidade. Apresentam-se o erro de predição filtrado para a variável  $y_1(k)$  e o primeiro componente do vetor  $\mathbf{E}_L$ ,  $\mathbf{E}_L(1)$ , somado com o erro verdadeiro de  $y_1(k)$ . O comportamento do sinal do erro aplicando o passo de estabilidade para a variável controlada  $y_1(k)$ , é muito mais agressivo do que o sinal do erro filtrado, de modo que compensa o erro de tempo morto no instante da mudança da referência. Como esperado, os erros no sistema em regime permanente coincidem.

As Figuras 36, 37, 38 e 39 mostram em detalhes as respostas obtidas aplicando DMC-Recursivo com passo de estabilidade e DMC-Recursivo com filtro no erro de predição nos intervalos onde são inseridas as perturbações. Com o objetivo de medir a velocidade de rejeição da perturbação, foi considerado um tempo de acomodação de 5% representado com as linhas tracejadas de cor preto. Na Figura 36, no minuto 450 tem-se uma perturbação entrando, observa-se que aplicando filtro no erro de predição na saída  $y_1(k)$ , foi preciso de 201 min para reagir à perturbação. No entanto, aplicando o passo de estabilidade foi preciso de 62 min, é dizer, aplicando a técnica passo de estabilidade obteve-

Figura 35 – Comparação dos erros  $E_f$  frente a  $E_L$ 

se uma resposta três vezes mais rápida em comparação com a técnica aplicando o filtro no erro de predição. Algo similar acontece na Figura 37, no minuto 1050 a perturbação sai do sistema. Nessa figura pode-se observar que a resposta aplicando o passo de estabilidade é duas vezes mais rápida que a resposta obtida aplicando o filtro no erro de predição.

Nas Figuras 38 e 39, observa-se o efeito causado pelo acoplamento existente no sistema. Nessas figuras são mostradas a resposta da saída  $y_2(k)$  que é afetada pelas perturbações inseridas em  $y_1(k)$ . Também observa-se que a resposta do sistema é mais rápida aplicando o passo de estabilidade em comparação com o filtro no erro de predição.

Figura 36 – Resultados de simulação  
Variável controlada

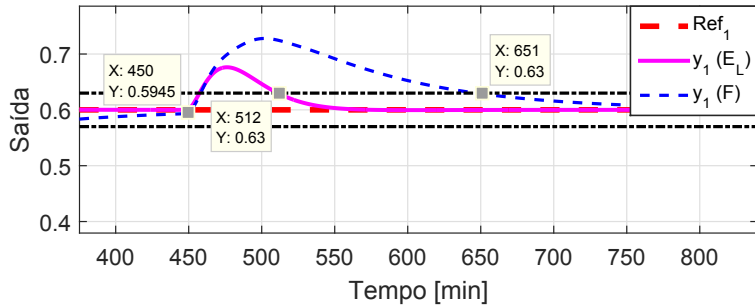


Figura 37 – Resultados de simulação  
Variável controlada

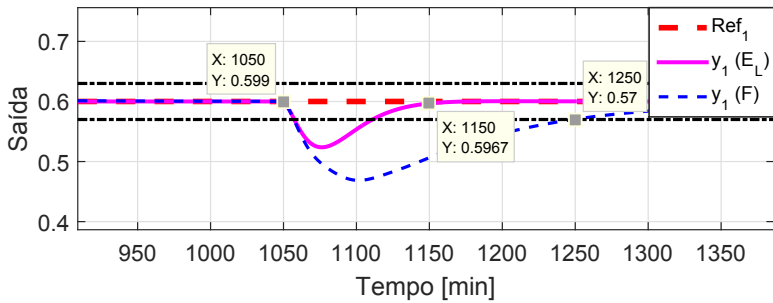


Figura 38 – Resultados de simulação  
Variável controlada

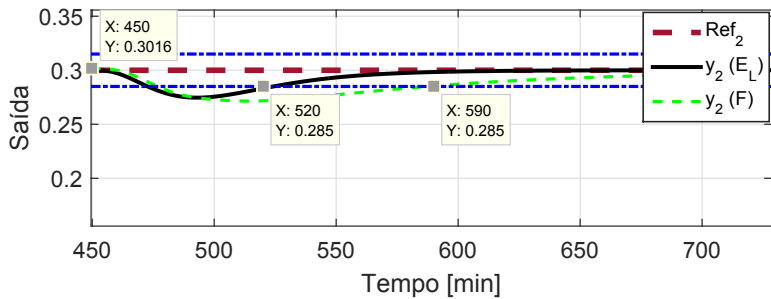
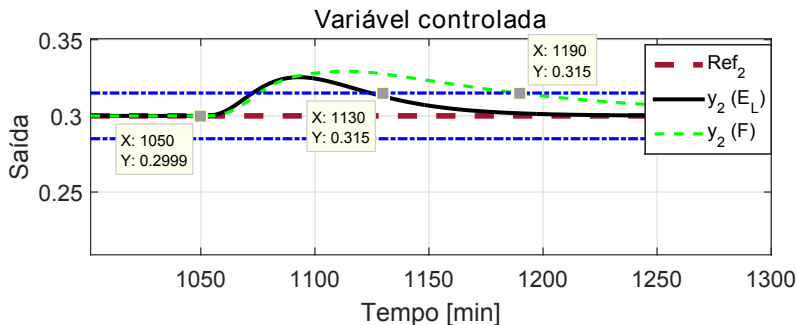


Figura 39 – Resultados de simulação



### 5.3 CONCLUSÕES

Neste capítulo, apresentou-se outra opção para lidar com erros de modelagem em plantas estáveis em malha aberta, que é baseado no segundo método de Lyapunov para a estabilidade. O passo de estabilidade modifica o erro de predição futuro de forma otimizada, com o objetivo de proporcionar estabilidade no sistema. Com o intuito de prover estabilidade, o algoritmo encontra a correção necessária à predição.

Conforme mostra a Figura 34 o tempo necessário para a técnica passo de estabilidade trazer  $y_1(k)$  à referência, é  $1/3$  do tempo utilizado pela técnica de filtragem quando uma perturbação não modelada perturba o sistema. A técnica passo de estabilidade apresentou um desempenho superior em comparação com o método de filtragem do erro de predição. Isso foi verificado tanto no seguimento de referência como na rejeição das perturbações não modeladas conforme se constata com os índices de desempenho dos controladores MSE apresentados na Tabela 14.

Tabela 14 – Índice de desempenho do controlador.

Estratégia	$MSE_1$	$MSE_2$
DMC <sub>F</sub>	1020,00	284,90
DMC <sub>E<sub>L</sub></sub>	590,66	140,31

Este é um resultado interessante, pois em muitos processos principalmente nas áreas de petróleo e gás, a maioria das variáveis afetadas por perturbações não modeladas tem um impacto econômico impor-

tante que afetam diretamente no desempenho do processo. Quanto mais tempo se encontram fora das faixas aceitáveis de operação, maior é a perda econômica.





## Capítulo 6

---

# CONSIDERAÇÕES FINAIS

---

Esse último capítulo conclui a dissertação, sintetizando as contribuições desse trabalho e discutindo os resultados obtidos. A partir dos resultados alcançados, um conjunto de trabalhos pode ser considerado. Assim, sugestões e direções para trabalhos futuros também são indicadas.

Para alcançar os objetivos propostos nesse trabalho, foi realizado, em primeiro lugar, um levantamento das pesquisas realizadas sobre os algoritmos de controle preditivo linear e não linear. Como resultados das pesquisas, este trabalho foi dividido em duas partes. Uma primeira parte envolve o estudo da técnica PNMPC e na segunda parte, foi realizado o estudo de técnicas usadas para sistemas que apresentem incertezas no modelo de previsão, principalmente no atraso de transporte, onde é apresentado o MPC com passo de estabilidade.

A técnica PNMPC foi escolhida como uma boa opção para o tratamento de sistemas não lineares por possuir uma boa relação desempenho, velocidade e simplicidade na implementação. Assim, foram introduzidos os conceitos de controladores preditivos com enfoque dedicado aos controladores PNMPC.

Apresentaram-se neste trabalho, as contribuições às estratégias dos controladores PNMPC. Foram propostas seis alternativas diferentes para a aplicação do algoritmo PNMPC. Esses algoritmos, são modificações da técnica completa proposta por Plucenio (2010). Os algoritmos propostos podem ser empregados como guia para o tratamento de sistemas não lineares. Cada um destes algoritmos permite tratar diferentes graus de não linearidade dos sistemas, permitindo escolher entre desempenho ou tempo de execução. Esta proposta oferece uma ampla gama de opções para o tratamento de sistemas não lineares aplicando a

estratégia PNMPC. Estes algoritmos, permitem tratar processos com dinâmica não linear da seguinte formas: como totalmente não lineares (resposta livre não linear e resposta forçada linearizada a cada iteração), como totalmente lineares (resposta livre linear e resposta forçada linearizada a cada  $n$  iterações), ou no caso onde a linearidade é suave, a resposta livre é tratada como não linear e a resposta forçada como linear.

Para avaliar os seis controladores, foi escolhido um estudo de caso com dinâmica não linear. Os seis algoritmos dos controladores PNMPC propostos apresentaram bom desempenho, sendo que o melhor desempenho é do controlador PNMPC completo (versão 1). Por outro lado, esta versão apresenta o maior custo computacional.

Na segunda parte do trabalho, foram estudadas técnicas que tratam incertezas no modelo de predição, principalmente no atraso de transporte, características que podem levar um processo à instabilidade.

Nos sistemas que apresentam incertezas no modelo de predição, principalmente no atraso de transporte, uma opção é o uso do filtro no erro de predição, partindo da idéia do preditor de Smith filtrado. Uma outra alternativa foi estudada nesta dissertação, o MPC com passo de estabilidade, apresentada em Plucenio et al. (2014). Nesse trabalho, foi realizada a comparação destas duas técnicas. Para isso, foi escolhido como estudo de caso um processo comumente encontrado na indústria petroquímica. Neste estudo de caso, foram consideradas incertezas no ganho, e no atraso de transporte entre a planta real e o modelo de predição. Os índices de desempenho obtidos nos resultados mostram que o MPC com passo de estabilidade apresenta melhor desempenho. O resultados obtidos nas simulações mostram que o MPC com passo de estabilidade apresenta maior rapidez na rejeição das perturbações não modeladas inseridas no sistema. Na implementação do MPC com passo de estabilidade nenhum filtro é aplicado.

Ainda como resultados do trabalho foram aprovados e publicados três artigos: dois no Congresso Brasileiro de Automática (CBA2016), realizado o 3 a 7 de outubro de 2016, em Vitória Espírito Santo intitulados "*Practical Non-linear Model Predictive Control PNMPC: Algorithm implementations*" (PICON; PLUCENIO, 2016b) e "*MODELING AND CONTROL OF AN OIL PLATFORM GAS COMPRESSION STATION*" (PLUCENIO; VETTORAZO; PICON, 2016) e um no 12<sup>th</sup> *IEE/IAS International Conference on Industry Applications* (INDUSCON 2016) realizado o 20 a 23 de novembro de 2016, em Curitiba intitulado "DMC de forma recursiva aplicando o passo de estabili-

dade” (PICON; PLUCENIO, 2016a).

Com tudo isso, pode-se concluir que os objetivos deste trabalho foram atingidos satisfatoriamente.

Como trabalhos futuros pode-se citar:

- avaliar as versões do PNMPC para outros processos que apresentam não linearidades em sua dinâmica;
- avaliar as versões do PNMPC para processos multivariáveis;
- estudo de robustez na implementação do método passo de estabilidade;
- estudo de métodos de sintonia para o controlador MPC com passo de estabilidade.



## REFERÊNCIAS

- ALLGOWER, F.; FINDEISEN, R. Nonlinear model predictive control: From theory to application. **Institute for Systems Theory in Engineering**, 2004.
- ANDRADE, G. et al. Controle preditivo não linear aplicado em sistemas de geração de energia solar. **CBA**, p. 8, Setembro 2012.
- BLOEMEN, H.; BOOM, T. Van den; VERBRUGGEN, H. Model-based predictive control for hammerstein systems. **Proceedings of the 39th IEEE Conference on Decision and Control**, 2000.
- BORDONS, C. **Control Predictivo: metodología, tecnología y nuevas perspectivas**. 2000.
- BRAVO, C.; NORMEY, J. Controle de plantas não lineares utilizando control preditivo linear baseado em modelos locais. **SBA: Controle e Automação Sociedade Brasileira de Automatica**, 2009.
- CAMACHO, E.; BORDONS, C. **Model Predictive Control in Process Industry**. 1995.
- CAMACHO, E.; BORDONS, C. **Model Predictive Control**. Second edition. 2007.
- CANNON, M. Efficient nonlinear model predictive control algorithms. **Annual Reviews in Control**, 2004.
- CASTILLA, M. et al. Control predictivo no lineal del confort térmico y calidad de aire. **Comité Español de Automática de la IFAC**, 2014.
- CLARKE, D.; MOHTADI, C. Generalized predictive control. part ii: extension and interpretations. **AUTOMATICA**, 1987.
- CLARKE, D.; MOHTADI, C. Properties of generalized predictive control. part i: The basic algorithm. **AUTOMATICA**, 1987.
- CUTLER, C.; REMARKER, B. Dynamic matrix control-a computer control algorithm. **In Automatic Control, Conference, San Francisco**, 1980.

FINDEISEN, R.; ALLGOWER, F. An introduction to nonlinear model predictive control. **21st Benelux Meeting on Systems and Control**, 2002.

GRUBER, J.; BORDONS, C. Control predictivo no lineal basado en modelos de voltterra. aplicacion a una planta piloto. **Revista Iberoamericana de Automatica e Informatica Industrial**, 2007.

KEYSER, R. D. A gentle introduction to model based predictive control. **PADI2 International Conference on Control Engineering and Signal Processing**, 1998.

KEYSER, R. D. The nepsac nonlinear predictive controller in a real life experiment. **1th International Conference on Intelligent Engineering Systems**, 2007.

LAWRYNCZUK, M. **Computationally Efficient Model Predictive Control Algorithms**. 2014.

LIMA, D.; NORMEY, J.; PLUCENIO, A. Improving robustness and disturbance rejection performance with industrial mpc. **Congresso Brasileiro de Automatica**, 2014.

LUCIA, S.; RUMSCHINSKI, P.; KRENER, A. Improved design of nonlinear model predictive controllers. **5th IFAC Conference on Nonlinear Model Predictive Control NMPC - Seville, Spain**, 2015.

MACIEJOWSKI, J. **Predictive Control with Constraints**. : Prentice Hall, 2002.

MAYNE, D. Control of constrained dynamic systems. **European Journal of Control**, 2001.

MONTEIRO, L. **Sistemas Dinâmicos**. 2006.

MORAES, A. D. et al. Controle preditivo para controle de nível do tanque de uma planta piloto industrial. **Simpósio Brasileiro de Automação Inteligente**, Outubro 2013.

MORARI, M.; LEE, J. Model predictive control: past, present and future. **elsevier**, 1999.

NELLES, O. Nonlinear system identification,. **Spinger**, 2001.

NORMEY, J. Gpc adaptativo basado em el predictor de smith para procesos con retardo variable. **SBAI**, 2001.

NORMEY, J.; CAMACHO, E. **Control of Dead time processes.** : Springer, 2007.

NORMEY, J.; LIMA, D.; SANTOS, T. Robustness of nonlinear mpc for dead-time processes. **5th IFAC Conference on Nonlinear Model Predictive Control NMPC - Seville, Spain**, 2015.

PICON, B.; PLUCENIO, A. Dmc de forma recursiva aplicando o passo de estabilidade. **12<sup>th</sup> IEEE/IAS International Conference on Industry Applications INDUSCON**, 2016.

PICON, B.; PLUCENIO, A. Practical non-linear model predictive control pnmpc: Algorithm implementations. **Congresso Brasileiro de Automatica CBA**, 2016.

PLUCENIO, A. **Desenvolvimento de Técnicas de Controle não linear para Elevação de Fluidos Multifásicos.** Dissertação (Dissertação de Doutorado-Engenharia de Automae Sistemas) — Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis”, 2010.

PLUCENIO, A. et al. Model predictive control algorithm with stability step. **Congresso Brasileiro de Automática**, Setembro 2014.

PLUCENIO, A. et al. Controle preditivo não linear na indústria do petróleo e gás. **PDPETRO**, Outubro 2007.

PLUCENIO, A. et al. A practical approach to predictive control for nonlinear processes. In: **IFAC**. 2007.

PLUCENIO, A.; VETTORAZO, C.; PICON, B. Modeling and control of an oil platform gas compression station. **Congresso Brasileiro de Automatica CBA**, 2016.

POMAR, M.; NORMEY, J.; CAMPONOGARA, E. Non linear model based predictive controller of a buck boost converter. **Proc. of the XVI IFAC World Congress**, 2005.

PRADA, C. D. El futuro del control de processos. **Revista Iberoamericana de Automatica e Informatica Industrial**, 2004.

QIN, S.; BADGWELL, T. A survey of industrial model predictive control technology. **Control Engineering Practice**, 2003.

RICHALET, J.; RAULT, A.; TESTUD, J. Model predictive heuristic control: Application to industrial process. **AUTOMATICA**, 1978.

SANTOS, J. **Controle Preditivo nao Linear para Sistemas de Hammerstein**. Tese (Doutorado) — UFSC, 2005.

SKRJANC, I.; MATKO, D. **Advances in Model-Based Predictive Control, chapter Fuzzy Predictive Controller with Adaptive Gain**. : Oxford University Press, 1994.

SMITH, O. Closer control of loops with dead-time. **Chem. Eng. Progress**, 1957.

TANG, Y.; KEYSER, R. D. **Advances in Model-Based Predictive Control, chapter Neural Network Based Predictive Control**. : 1994, 1994.

THOME, F. **Controle Preditivo Rubusto de Sistemas Não Lineares com Atraso**. Dissertação (Mestrado) — UFSC, 2013.



## APÊNDICE A – Apêndice



PRACTICAL NON-LINEAR MODEL PREDICTIVE CONTROL PNMP: ALGORITHM IMPLEMENTATIONS

BRYANT PICON YANG\*, AGUSTINHO PLUCENIO†

\**Dpto. de Automação e Sistemas*  
*Universidade Federal de Santa Catarina*  
*88040-900*  
*Florianópolis, SC, Brasil*

†*Firenze-Consultoria em Engenharia e Informatica*  
*Av. Lédio João Martins, 201-203 - 88102-000 KOBASOL São José, SC, Brasil*

Emails: bryant.bruce@posgrad.ufsc.br, brplucenio@firenzeconsult.com,

**Abstract**— Most dynamic processes exhibit nonlinear relationships between controlled and manipulated variables. Depending on the application, the MPC algorithm is often chosen between a non-linear or a linear type. This paper presents different implementations of the PNMP-Practical Non-linear Model Predictive Control algorithm. It is shown that the PNMP algorithm may be written to handle the non-linearity at different levels. For the sake of simplicity, the algorithms are implemented to control the level of a conical tank. The main purpose of the controller is to manipulate the position of the outlet valve in order to follow the reference. The performance and the computational cost of six different implementations of PNMP controller are shown. The results of the controllers are compared and analyzed by the system response verification, the algorithm runtime and driver performance.

**Keywords**— Nonlinear predictive control, Computational cost, Conical tank, Level control, SISO system

**Resumo**— A maioria dos processos dinâmicos exibem relações não lineares entre variáveis controladas e manipuladas. Dependendo da aplicação, o algoritmo MPC é muitas vezes escolhido entre um tipo linear ou não linear. Este trabalho apresenta varias maneiras de implementar o algoritmo PNMP. Mostra-se que o algoritmo PNMP pode ser escrito para lidar com a não-linearidade em diferentes níveis. Por uma questão de simplicidade, os algoritmos são implementados para controlar o nível de um tanque cônico. A principal finalidade do controlador é manipular a abertura da válvula de saída, a fim de seguir a referência. O desempenho e o custo computacional de seis implementações diferentes de controlador PNMP são exibidos. Os resultados dos controladores são comparados e analisados pelo sistema de verificação de resposta, o tempo de execução do algoritmo e o desempenho do atuador.

**Palavras-chave**— Controle preditivo não linear, Custo computacional, Tanque cônico, Controle de nível, Sistema SISO

## 1 Introduction

The Model Predictive Control (MPC) is one of the most advanced modern control techniques, being probably the one that has been the most successful in industrial applications especially in the oil industry (García, 1989).

The history of advanced control begins in the early 1960's with the work of Kalman as well as co-author Ricatti and others, (Qina and Badgwell, 2003). But it was only after a set of papers presented in the late 1970's that MPC controllers became interesting for industrial applications

In 1978, Richalet described the applications of "Control Models Predictive Heuristic" in 1979 Shell Engineers Cutler and Ramaker as well as Pratt and Gillette outlined "Dynamic Control Matrix" (DMC) and reported applications for a catalytic tab fluid. In both algorithms, an explicit dynamic model of the plant has been used to predict the effect of future control actions on the output, hence the name "Model Predictive Control".

The main causes of this success is due to the ability of the MPC to control a wide range of processes, from those having a relatively simple dy-

namic to others more complex, for example: Systems with delay, multivariables, unstable, as well non-linear dynamic among others. The MPC also allows to obtain solutions, notwithstanding the existence of constraints in the variables (Normey and Camacho, 2007).

The majority of the real systems have non-linear dynamics, however, when the process operational range is small, the dynamics can be satisfactorily approximated by linear models. This approach has motivated many MPC techniques using linear models such as: Dinamix Matrix Control-DMC, (C.R. Cutler and B.L. Ramaker, 1988); Model Algorithm Control-MAC, (Richalet, Rault, Testud and Papon, 1976); Generalized Predictive Control-GPC, (Clarke, Mothadi and Tuffs, 1987)) (Plucenio et al., 2007).

The main advantage of linear MPC is the low cost to obtain linear models in the development of the project compared to non-linear approaches. Other advantages are lesser difficulties to solve the associated optimization problems. On the other hand, when the processes have a very strong non-linear dynamic or when the operation range is wide, then it might be necessary to consider

the non-linear model in the control algorithm, to maintain the desired performance for the closed-loop system.

When choosing a predictive control algorithm, the initial decision is to choose between two algorithm systems, either the linear or the non-linear systems. In many cases we have good non-linear process models. Unfortunately it is common practice to get a linear representation of the model in a single point of operation and use this representation to obtain the dynamic matrix coefficients and the free response. The original PNMPC technique uses a real-time linearization of the input-output variables. This linearization is performed in a two step process and it obtains almost a second-order approximation. Depending on the non-linearity process to be controlled, there are several ways to implement the PNMPC algorithm. The algorithm choice will depend on the available processing time and quality of response to be obtained. The PNMPC can be implemented as a linear controller. Due to the way it treats the prediction error, the linear version of PNMPC will be similar to a GPC.

This paper is organized as follows: section 2 presents the description of the mathematical model of the conical tank. The theoretical principles of non-linear control strategy are presented in section 3. The description of the six alternative versions of the implementation of the PNMPC are shown in section 4. And section 5 presents the results of implementations described in section 4.

## 2 Mathematical Model Description of the Conical Tank

To demonstrate the application of the Practical MPC algorithm for non-linear systems (PNMPC), a process with a non-linear dynamics was considered. Furthermore and in order to simplify the demonstration, the level control in a conical tank (SISO system) was also used. The process is shown in Figure 1, where the manipulated variable is the output valve. The input flow will act as a disturbance in the system. The mass balance

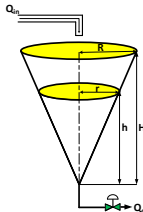


Figure 1: conical tank

inside the tank is given by the equation:

$$Q_{in} - Q_o = A \frac{dh}{dt} \quad (1)$$

where:

$Q_{in}$	input flow;
$Q_o$	output flow;
$R$	maximum radius of the conical tank;
$r$	radius of the conical tank at steady state;
$H$	maximum height of the conical tank;
$h$	height of the conical tank at steady state.

Whereas the cross-section area of the conical tank is given by:

$$A = \pi \left[ r - (R - r) \left( \frac{h}{H} \right) \right]^2 \quad (2)$$

and the output flow is represented by the equation:

$$Q_o = C_v U_c \sqrt{h} \quad (3)$$

where  $U_c$  is the control action of the outlet valve and  $C_v$  is a constant for the valve.

Substituting equations (2) and (3) in (1), we have the following model of the conical tank:

$$\frac{dh}{dt} = \frac{Q_{in} - C_v U_c \sqrt{h}}{\pi \left[ r - (R - r) \left( \frac{h}{H} \right) \right]^2} \quad (4)$$

## 3 PNMPC control

The PNMPC technique was developed by (Plucenio, 2010) and is based on algorithms that use the vector representation of predictions over the horizon  $p$ ,  $\tilde{\mathbf{Y}}$ , as a vector function with  $m$  changes in the control action  $\Delta \mathbf{u}$ , according to the equation (5). The PNMPC is a technique that offers an alternative to solve the problem of linear optimization using conventional optimizers as QP quadratic programming or linear programming solution.

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{G} \Delta \mathbf{u} \quad (5)$$

The PNMPC differs from other MPC techniques mainly because it uses linearized models of the system trajectory. It is assumed that the predictions  $\tilde{\mathbf{Y}}$  depend only on past entries  $\overleftarrow{\mathbf{u}}$ , past outputs  $\overleftarrow{\mathbf{y}}$  and the future control increments  $\Delta \mathbf{u}$ .

$$\tilde{\mathbf{Y}} = f(\overleftarrow{\mathbf{y}}, \overleftarrow{\mathbf{u}}, \Delta \mathbf{u}) \quad (6)$$

The vector of predictions is rewritten as:

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{G}_{PNMPC} \Delta \mathbf{u} \quad (7)$$

where

$$\mathbf{F} = f(\overleftarrow{\mathbf{y}}, \overleftarrow{\mathbf{u}}) \quad (8)$$

$$\mathbf{G}_{PNMPC} = \frac{\partial \tilde{\mathbf{Y}}}{\partial \Delta \mathbf{u}} \quad (9)$$

The matrix  $\mathbf{G}_{PNMPC}$  is the Jacobian of  $\tilde{\mathbf{Y}}$  relative to the control increments and is obtained numerically as the vector of free response  $\mathbf{F}$ . A prediction algorithm computes the vector  $\tilde{\mathbf{Y}}$  with

the  $p$  predictions when it provides the values of the past inputs and outputs, and the vector with the  $m$  future input increments  $\Delta \mathbf{u}$ .

If we call the set of past inputs  $\overleftarrow{\mathbf{u}}$  and the current and past output  $\overleftarrow{y}$ , predictions can be rewritten in the following expression.

$$\begin{aligned} \tilde{y}(k+1) &= f(\overleftarrow{y}, \overleftarrow{\mathbf{u}}, \Delta \mathbf{u}(k)) \\ \tilde{y}(k+2) &= f(\overleftarrow{y}, \overleftarrow{\mathbf{u}}, \Delta \mathbf{u}(k), \Delta \mathbf{u}(k+1)) \\ &\vdots \\ \tilde{y}(k+p) &= f(\overleftarrow{y}, \overleftarrow{\mathbf{u}}, \Delta \mathbf{u}(k), \dots, \Delta \mathbf{u}(k+m-1)) \end{aligned} \quad (10)$$

The following compact representation is obtained:

$$\tilde{\mathbf{Y}} = \mathbf{F} + \mathbf{G}_{PNMPC} \Delta \mathbf{u} \quad (11)$$

Whereas that  $\mathbf{F}$  is the vector of predictions that would be obtained for  $\Delta \mathbf{u} = 0$  and  $\mathbf{G}_{PNMPC}$  is the Jacobian of the predicted outputs with respect to the vector of increments of control signal  $\Delta \mathbf{u}$ .  $\mathbf{G}_{PNMPC}$  is a generalized dynamic matrix that can be used for linear systems and non-linear systems, provided they are continuous and differentiable. The expression (11) for a SISO system can be re-written as:

$$\begin{bmatrix} \tilde{y}(k+1) \\ \tilde{y}(k+2) \\ \vdots \\ \tilde{y}(k+p) \end{bmatrix} = \begin{bmatrix} f_1(\overleftarrow{y}, \overleftarrow{\mathbf{u}}) \\ f_2(\overleftarrow{y}, \overleftarrow{\mathbf{u}}) \\ \vdots \\ f_p(\overleftarrow{y}, \overleftarrow{\mathbf{u}}) \end{bmatrix} + \begin{bmatrix} 0 & \dots & 0 \\ \frac{\partial \tilde{y}(k+1)}{\partial \Delta u(k)} & \dots & 0 \\ \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+2)}{\partial \Delta u(k+1)} & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k)} & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+1)} & \dots & \frac{\partial \tilde{y}(k+p)}{\partial \Delta u(k+m-1)} \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+m-1) \end{bmatrix} \quad (12)$$

The lower triangular form of the Jacobian matrix is due to the causality of the system and absence of direct coupling between the input and output. This means  $\frac{\partial \tilde{y}(k+j)}{\partial \Delta u(k+i)} = 0$  for  $i \geq j$ . The partial derivative  $\frac{\partial \tilde{y}_k}{\partial \Delta u(k)}$  it is defined as:

$$\frac{\partial \tilde{y}_{k+j}}{\partial \Delta u(k)} = \lim_{\Delta u_k \rightarrow 0} \frac{\tilde{y}_{k+j}(u_{k-1} + \Delta u_k) - \tilde{y}_{k+j}(u_{k-1})}{\Delta u_k} \quad (13)$$

For a SISO system, for example, on the first instant of sampling, the following procedure for obtaining the numerical value of  $\mathbf{F}$  and  $\mathbf{G}_{PNMPC}$  is performed:

1. Get the vector  $\tilde{\mathbf{Y}}_p^0$  (dimension  $p \times 1$ ) executing the prediction model with the past inputs and outputs and  $\Delta \mathbf{u} = [0 \ 0 \ \dots \ 0]^T$ .  $\mathbf{F} = \tilde{\mathbf{Y}}_p^0$ .
2. Computes the first column of the matrix  $\mathbf{G}_{PNMPC}$ . Obtaining the vector  $\tilde{\mathbf{Y}}_p^1$  (dimension  $p \times 1$ ) executing the model with the past inputs and outputs and considering  $\Delta \mathbf{u} = [\epsilon \ 0 \ \dots \ 0]^T$ , where  $\epsilon$  is a very small value,  $\frac{u_{k-1}}{1000}$ , for example.

$$\mathbf{G}_{PNMPC}(:, 1) = \frac{\tilde{\mathbf{Y}}_p^1 - \tilde{\mathbf{Y}}_p^0}{\epsilon}$$

3. Computes the second column of the matrix  $\mathbf{G}_{PNMPC}$ . Obtain the vector  $\tilde{\mathbf{Y}}_p^2$  (dimension  $p \times 1$ ) executing the model with the past inputs and outputs and considering  $\Delta \mathbf{u} = [0 \ \epsilon \ \dots \ 0]^T$ .

$$\mathbf{G}_{PNMPC}(:, 2) = \frac{\tilde{\mathbf{Y}}_p^2 - \tilde{\mathbf{Y}}_p^0}{\epsilon}$$

4. Continues with the calculation of the others columns of the matrix  $\mathbf{G}_{PNMPC}$  to the last column, where the vector  $\tilde{\mathbf{Y}}_p^m$  is obtained executing the model with the past inputs and outputs  $\Delta \mathbf{u} = [0 \ 0 \ \dots \ \epsilon]^T$ .  $\mathbf{G}_{PNMPC}(:, m) = \frac{\tilde{\mathbf{Y}}_p^m - \tilde{\mathbf{Y}}_p^0}{\epsilon}$ .

For the application of PNMPC technique in multivariable systems it is necessary to concatenate the prediction vectors for each variable  $\tilde{\mathbf{Y}}$  and the control increments  $\Delta \mathbf{u}$ . Thus, a system with  $ni$  entries and  $no$  outputs have the following vectors:

$$\begin{aligned} \tilde{\mathbf{Y}} &= [\tilde{\mathbf{Y}}_1 \ \tilde{\mathbf{Y}}_2 \ \dots \ \tilde{\mathbf{Y}}_{pno}]^T \\ \Delta \mathbf{u} &= [\Delta u_1 \ \Delta u_2 \ \dots \ \Delta u_{mni}]^T \end{aligned} \quad (14)$$

To obtain  $\mathbf{F}$  and  $\mathbf{G}_{PNMPC}$  the procedure is then as above. The difference is that now the matrix  $\mathbf{G}_{PNMPC}$  have  $ni \times no$  computed blocks as in the SISO case for each pair of input and output. Obviously, as in linear MPC, weak relationships output-input may be disregarded to avoid noise propagation. The control action is obtained by minimizing the cost function  $J$ . Using equation (7) as a way of expressing the predictions, the quadratic objective function from section 2.2.2 can be re-written as follows:

$$\begin{aligned} \min_{\Delta \mathbf{u}} J \\ \text{subject to} \\ J &= (\tilde{\mathbf{Y}} - \mathbf{W})^T \mathbf{R} (\tilde{\mathbf{Y}} - \mathbf{W}) + \Delta \mathbf{u}^T \mathbf{Q} \Delta \mathbf{u}, \\ \tilde{\mathbf{Y}} &= \mathbf{F} + \mathbf{G}_{PNMPC} \end{aligned} \quad (15)$$

where  $\mathbf{W}$  represents the vector of future reference,  $\mathbf{R}$  the error weighting matrix and  $\mathbf{Q}$  the control weighting matrix. The minimization of the cost function for the case without constraints can be obtained by equating to zero the gradient of the cost function and in the case with constraints the solutions is obtained by solving a QP.

Thus, in terms of optimization, the PNMPC has a complexity similar to the DMC or the GPC algorithm. The predictive control techniques applied to linear systems use a mechanism that consists in correct the predictions with the error between the measured value of the controlled variable and its predicted value at time  $k$ . This technique applied to linear systems guaranteed zero error in steady state, because the increase in the control effort required for error correction at the instant  $k$  will be the same in future instants due to the linear relationship between the input and output, so the gain is constant. The application of this procedure to the PNMPC technique applied to non-linear systems does not guarantee null error in steady state, because the gain varies with the operating point of the system. The procedure to correct the predictions presented in (Plucenio, 2010), consists in adding to each prediction the integral of the filtered prediction error. Figure 2 shows how the correction factor for each controlled variable can be obtained. A correction factor

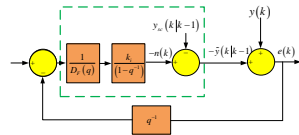


Figure 2: Block diagram of the correction factor

is added to the prediction so that the error between the measured and the corrected value of prediction to

zero. The correction factor is obtained filtering and integrating the error as shown in Figure 2. Considering the controlled variable  $y(t)$ , if the level error has a linear approximation, the transfer function that relates the error and the difference between the measured and the predicted values of the variable without correction can be obtained as shown in (16). It must be noted that the error  $e(z)$  is defined as the difference between the measured value and the corrected predicted value, whereas  $(y(k) - \hat{y}_{sc}(k))$  corresponds to the error between the measured variable and the predicted value without correction.

$$\frac{e(z)}{(y(k) - \hat{y}_{sc}(k))} = \frac{z^2 - (1 + f_d)z + f_d}{z^2 - (1 + f_d - k_i)z + f_d} \quad (16)$$

Assuming the closed loop system is stable, we can apply the Final Value Theorem to verify that the presence of  $(1 - z^{-1})$  in the numerator of the transfer function (16) guarantees null error in steady state for an input of type step. The values of  $k_i$  and  $f_d$  are chosen by comparing the denominator of the transfer function (16) with the desired polynomial  $p_d(z) = z^2 - 2az + a^2$ . The value of  $a$  has an impact on the speed at which the error responds to disturbances. After the value of  $a$  is determined,  $f_d$  and  $k_i$  can be obtained through equations (17) and (18).

$$f_d = a^2 \quad (17)$$

$$k_i = 1 + a^2 - 2a \quad (18)$$

#### 4 Implementation of PNMPCC: Alternatives

The PNMPCC presented in the previous section can be implemented in different ways. The goal of this paper is to study the performance and the computational cost of the PNMPCC controller, by using the following: six different versions of the PNMPCC implementation, as well as simplifications of the complete original technique to reduce the processing time. The way of calculating the predictions can be modified to convert the PNMPCC into a linear MPC controller. Six versions of the PNMPCC will be applied to the conical tank. The goal is to control the level of the tank through outlet valve manipulation.

##### 4.1 1st version: PNMPCC complete

Following the PNMPCC theory from (Plucenio, 2010), the first PNMPCC proposed in this paper considers the non-linear free response and calculation of the matrix  $\mathbf{G}$  in two steps. The first procedure of the algorithm is to calculate at a given iteration the variation of initial control ( $\Delta \mathbf{u}_{ini}$ ), which is calculated according to the existing matrix  $\mathbf{G}$ , that is, the matrix  $\mathbf{G}_{previous}$ . Obtaining the initial control variation, the future control actions are calculated according to the equation:

$$\mathbf{U}_{m \times 1} = \mathbf{H}_{m \times 1} \mathbf{u}(k-1) + \mathbf{T}_{m \times m}^T \Delta \mathbf{u}_{ini_{m \times 1}} \quad (19)$$

where:

$m$	control horizon;
$\mathbf{H}$	vector of ones;
$\mathbf{u}(k-1)$	previous control action;
$\mathbf{T}^T$	integrating Toeplitz matrix;
$\Delta \mathbf{u}_{ini}$	initial control variation.

With the control action calculated, an initial predicted output  $\hat{\mathbf{Y}}_p^0$  is obtained. A control increment  $\varepsilon$  is added to the control vector  $\mathbf{U}$  and a new output prediction ( $\hat{\mathbf{Y}}$ ) is obtained. This value  $\varepsilon$  will scroll through the  $\Delta \mathbf{u}$  vector in order to calculate the matrix  $\mathbf{G}_{PNMPCC}$  as described in (Plucenio, 2010) and shown in equation (20).

$$\mathbf{G}_{PNMPCC}(:, m) = \frac{\hat{\mathbf{Y}}_p^m - \hat{\mathbf{Y}}_p^0}{\varepsilon} \quad (20)$$

In fact, the calculation of the derivative of this new control signal  $\mathbf{U}$ , serves to compensate the non-linearity of the system. Having the matrix  $\mathbf{G}(k)$  at the current time and the matrix  $\mathbf{G}(k-1)$  calculated in the previous instant  $\mathbf{G}_{previous}$ , it is possible calculate in two steps the intermediate matrix  $\bar{\mathbf{G}}_{PNMPCC}$  by the following equation:

$$\bar{\mathbf{G}}_{PNMPCC} = 0.5\mathbf{G}(k-1) + 0.5\mathbf{G}(k) \quad (21)$$

The algorithm implemented for this version is detailed in the flow diagram in Figure 3:

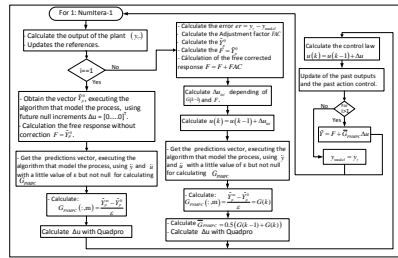


Figure 3: PNMPCC complete algorithm flow diagram

##### 4.2 2nd version: PNMPCC without calculation $\Delta \mathbf{u}_{ini}$

For processes with a strong non-linear characteristic it is recommended to choose the complete PNMPCC technique (first version), but if the process presents a smooth non-linearity it may be possible to reduce the computational cost by eliminating the calculation of  $\Delta \mathbf{u}_{ini}$ . In this version it is proposed the calculation of the  $\mathbf{G}_{PNMPCC}$  matrix in one step. Therefore, it is not necessary to calculate  $\Delta \mathbf{u}_{ini}$ . In order to calculate the control action needed to obtain  $\hat{\mathbf{Y}}_p^0$  the value of the control variation of the previous iteration is used. This  $\hat{\mathbf{Y}}_p^0$  is then calculating the matrix  $\mathbf{G}_{PNMPCC}$ , as shown in section three. Once the calculation of  $\Delta \mathbf{u}_{ini}$  is eliminated, the precision with which the PNMPCC calculates the  $\mathbf{G}_{PNMPCC}$  decreases, this can be explained as the  $\Delta \mathbf{u}_{ini}$  has a direct influence in the calculation of the  $\hat{\mathbf{Y}}_p^0$  as it is directly involved in the calculation of the matrix  $\mathbf{G}_{PNMPCC}$ .

##### 4.3 3rd version: PNMPCC getting only the first column of $\mathbf{G}$

As already described in the literature, the matrix  $\mathbf{G}$  the one that is responsible to save the values of the dynamics process (Camacho and C., 2007) and

the  $\mathbf{G}_{PNMPC}$  is the non-linear version of the dynamic matrix used in the linear predictive control (Plucenio, 2010). One of the possibilities to reduce the computational cost is to calculate only the first column of the matrix  $\mathbf{G}$  by the normal procedure of PNMPC and mount the matrix  $\mathbf{G}_{PNMPC}$  considering the system time-invariant, this procedure is commonly performed in linear systems (De Keyser, 1998). Thus the non-linear optimization problem can be transformed into a sequence of linear optimization problems. For this it is necessary to check how much the diagonal elements of the matrix  $\mathbf{G}$  vary. In the case where the variations are not significant, one can opt for the technique used in linear systems; only calculate the first column of the matrix  $\mathbf{G}$  and displaced until a lower triangular matrix is obtained, that means treat the non-linear process as if it were linear. In the case of a non-linear process, it can be seen that the diagonal elements of the matrix  $\mathbf{G}$  present significant variations between the elements of the diagonal, however, in the case of a linear process, the diagonals elements of the matrix  $\mathbf{G}$  do not present significant variations. To check the degree of the non-linearity in the study case, Figure 4(a) shows the variations in the diagonals elements of the matrix  $\mathbf{G}$  calculated by the Jacobian matrix ( $\mathbf{G}_{PNMPC}$ ). As this matrix is recalculated on

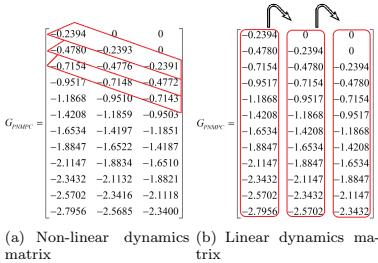


Figure 4: Dynamic matrix  $\mathbf{G}_{PNMPC}$

each iteration, the data shown in Figure 4(a) represent only a particular step, the iteration  $k = 35$ . As shown, the diagonal elements of the matrix  $\mathbf{G}_{PNMPC}$  present variations, but these variations are small, so they could be disregarded. Applying the third version of the PNMPC algorithm proposed in this paper, where only the first column of the matrix  $\mathbf{G}$  is calculated, so that the new matrix  $\mathbf{G}$  for iteration  $k = 35$  shown in Figure 4(b). In the Figure 4(b) the first column of the matrix  $\mathbf{G}$  is shifted  $m$  times, to form a lower triangular matrix, where  $m$  is the horizon control. With this modification, the computational cost for the calculation of  $\mathbf{G}$  decreases reasonably.

#### 4.4 4th version: PNMPC similar to 3rd version, but now the matrix $\mathbf{G}$ is recalculated only from time to time.

This version is a very interesting and flexible option. The process is treated as if it were linear and only the first column of  $\mathbf{G}$  is calculated. However, the matrix  $\mathbf{G}$  is not obtained at each iteration, it is recalculated every  $n$  sampling times. An operator with knowledge

of the process and the non-linearity degree can adjust the value of  $n$  in each application. In the case study of this paper, as shown in the previous version, the non-linearity degree of the process is not significant, so the recalculation of the matrix  $\mathbf{G}$  does not need to be performed at each iteration. For the testes was used a recalculation every 25 samples. The choice of this time was made taking care not to lose the dynamics of the process.

#### 4.5 5th version: PNMPC looking for free response time to time, using the Cutler idea (Recursive DMC) to the other instants

This version seeks to reduce the computational cost in processes where the non-linearity is not significant, inserting in the PNMPC technique the calculation of free response as is done in the recursive DMC proposed by Cutler. To (Cutler and Ramaker 1988), as shown in (Plucenio et al., 2014), the open-loop predictions for time  $t+k$ , having information in  $t$  and  $t-1$  are shown in the equations (22) and (23) respectively:

$$y_0(t+k|t) = \sum_{i=k+1}^{\infty} g_i \Delta u(t+k-i) \quad (22)$$

$$y_0(t+k|t-1) = \sum_{i=k+2}^{\infty} g_i \Delta u(t+k-i) \quad (23)$$

The difference between the open-loop prediction in  $t+k$  having  $t$ , and in  $t+k$  having  $t-1$ , it is only the new control action which is not known in  $\Delta u(t-1)$ . Subtracting the equations (22) and (23) has the equation (24):

$$y_0(t+k|t) - y_0(t+k|t-1) = g_{k+1} \Delta u(t-1) + \sum_{i=k+2}^{\infty} g_i \Delta u(t+k-i) - \sum_{i=k+2}^{\infty} g_i \Delta u(t+k-i) \quad (24)$$

which leads to the equation (25):

$$y_0(t+k|t) = g_{k+1} \Delta u(t-1) + y_0(t+k|t-1) \quad (25)$$

The recursive DMC calculation is executed as follows: Keep in memory a vector with  $N$  elements:  $\mathbf{Y}_{free} = [y_0(t|t-1), \dots, y_0(t+N-1|t-1)]^T$ . Its elements are the free responses given the known control actions at the moment  $t-1$ . When starting the DMC controller at the moment  $t_0$ , and as the system is in steady state, it is considered that the open-loop predictions are constant and equal to a value of  $y(t_0)$ . When starting the DMC algorithm at the moment  $t$ , the vector of the free response must be updated because the change of control action  $\Delta u(t-1)$  it is now known. The vector  $\mathbf{Y}_{free}$  of the equation (26) is updated.

$$\mathbf{Y}_{free} = \mathbf{Y}_{free} + \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} \Delta u(t-1) \quad (26)$$

After getting the new control signal  $u(t)$ , it is necessary to 'displace' the values within the vector  $\mathbf{Y}_{free}$ . At time  $t+1$  the free responses in  $t+1$  until  $t+N$  is obtained based on the data until  $t$ , where  $N$  is called

the model horizon. So the first element is discarded because it is the prediction concerning to instant  $t$  that has passed. The problem is that making this displacement, the last element ( $y(t+N|t)$ ) is unknown. Therefore, in the stable case ( $y_0(t+N|t) \cong y_0(t+N-1|t)$ ), the new vector  $\mathbf{Y}_{free}$  will be given by the equation (27):

$$\mathbf{Y}_{free} = \begin{bmatrix} y_0(t+1|t) \\ y_0(t+2|t) \\ \vdots \\ y_0(t+N-1|t) \\ y_0(t+N-1|t) \end{bmatrix} \quad (27)$$

The condition  $y_0(t+N|t) \cong y_0(t+N-1|t)$  is shown as:

$$\begin{aligned} y_0(t+N-1|t) &= \sum_{i=N}^{\infty} g_i \Delta u(t+N-1-i) \\ y_0(t+N|t) &= \sum_{i=N+1}^{\infty} g_i \Delta u(t+N-i) = \\ &= \sum_{i=N}^{\infty} g_{i+1} \Delta u(t+N-1-i) \\ y_0(t+N|t) - y_0(t+N-1|t) &= \\ &= \sum_{i=N}^{\infty} (g_{i+1} - g_i) \Delta u(t+N-1-i) \end{aligned}$$

As  $g_{i+1} - g_i \cong 0, \forall i > N$ , it has  $y_0(t+N|t) \cong y_0(t+N-1|t)$ , as was able to demonstrate.

Thus, the calculation of free response considered in the DMC Recursive is given by the equation:

$$\mathbf{f} = \begin{bmatrix} y_0(t+1|t) \\ \vdots \\ y_0(t+p|t) \end{bmatrix} + \mathbf{1}_{p \times 1} (y(t) - y_0(t|t)) \quad (28)$$

Where the equation of the model error is:

$$e = y(t) - y_0(t|t) \quad (29)$$

As shown in (Plucenio et al., 2014), the representation of predictions considering the calculation of the free response to the DMC Recursive is shown in the equation (30):

$$\tilde{\mathbf{Y}} = \mathbf{f} + \mathbf{G} \Delta \mathbf{u} \quad (30)$$

where:

- $\tilde{\mathbf{Y}}$  predicted output with dimension  $p \times 1$ ;
- $\mathbf{f}$  free corrected response with dimension  $p \times 1$ ;
- $\mathbf{G}$  Jacobian matrix of  $\tilde{\mathbf{Y}}$  with dimension  $p \times m$ ;
- $\Delta \mathbf{u}$  vector with dimension  $m \times 1$  containing the increments of the control action.

Using this form of implementation, the necessary calculations to obtain the free response become more simple, not requiring the storage of past control increments. This procedure only changes the way to calculate the free response, the rest of the control algorithm is not modified.

#### 4.6 6th version: PNMPCC reassessing the matrix $\mathbf{G}$ and free response according to the prediction error

The idea of this version is calculate the absolute value of the prediction error, with the objective to fix a range tolerable error in which it is considered that the values obtained are acceptable, and it is not necessary to recalculate the  $\mathbf{G}$  matrix and the free response for each iteration.

Table 1: Tuning parameters for the PNMPCC controllers

Param	Definition	Value
$p$	prediction horizon	12
$m$	control horizon	3
$\lambda$	control-weighting factor	3
$\gamma$	reference tracking weighting factor	1
$\alpha$	reference filter	0.25
$\epsilon$	value for variation in control	0.01
$a$	filter parameter for prediction error	0.15

Table 2: Comparative table of the execution times for the PNMPCC versions

Execution	1st version	2nd version	3rd version	4th version	5th version	6th version
1	40.2191	38.1917	25.7825	19.3745	29.4440	27.5302
2	35.5854	38.2922	25.6580	19.1153	29.3883	27.0980
3	38.8533	38.4277	25.6750	19.1802	29.5960	27.6553
4	38.9484	38.1932	25.6655	19.1637	29.2337	27.5607
5	39.1778	38.3610	25.8829	19.0723	29.0735	27.4466
6	39.0862	38.5138	25.6190	19.2308	29.0478	27.6613
7	39.1187	38.6370	25.7077	19.2133	29.0250	27.4830
8	39.1515	38.5807	25.7749	19.1650	29.0478	27.7040
9	39.1868	38.7472	25.6069	19.2164	29.0875	27.6850
10	38.9844	38.4075	25.7159	19.0604	29.4870	27.7875
Total time [s]	392.3513	384.4420	257.1723	191.8008	293.4296	276.2206
Average [s]	39.2351	38.4442	25.7172	19.1800	29.3429	27.6220

## 5 Results

This section presents the results obtained in the different versions implemented with the PNMPCC technique. As the main objective is to evaluate the computational cost involved in the implementation of the PNMPCC technique, it will be shown the runtime obtained in the calculation of the algorithm for each version of the PNMPCC applied in the case study on the non-linear process of the level control of the conical tank. All simulations were performed in MatLab 2013, on a PC with A8 quad-core processor, AMD with 1.9 GHz and 6GB of RAM, with Windows 8. To first obtain an accurate approximation of the model, the differential equation describing the model was integrated using Ode45 function of MatLab. The tuning parameters used for the PNMPCC controllers are the same for all versions for comparison purposes, these values are shown in Table 1. The sample time for all versions is 10 second for reference greater than 1 meter and 1 second for reference less than 1 meter. Table 2 shows runtime the values for each version of the PNMPCC technique algorithm implemented in section 4.

To estimate the execution time of each algorithm, each version was executed 10 times and the final time was calculated as the average of 10 executions. The time is measured in seconds and the results are shown in Table 2.

Figure 5 shows the results for two cases. The first for reference greater than 1 meter (Figure 5(a), 5(c) and 5(e)) and the second test for reference less than 1 meter (Figures 5(b), 5(d) and 5(f)).

## 6 Conclusions

This paper presented six different versions of the PNMPCC technique. With little modifications of the complete technique, either in the calculation of the free response or in assembling the matrix  $\mathbf{G}_{PNMPCC}$ , the processing time is reduced. With these modifications the PNMPCC can be simplified and be converted into a linear MPC controller. As shown in the results in



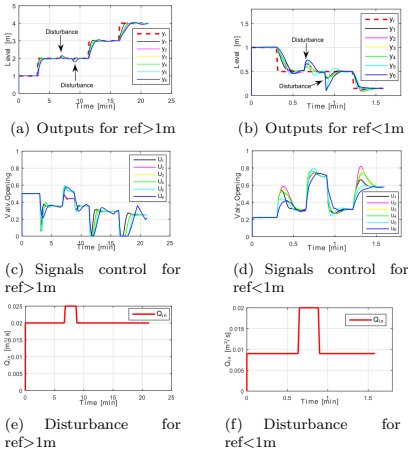


Figure 5: Outputs references, control signals, disturbance

Table 3: Performance index of the PNMPC versions

Performance index	$V_1$	$V_2$	$V_3$	$V_4$	$V_5$	$V_6$
MSE	0.0031	0.0040	0.0063	0.0085	0.0100	0.0096
IAE	2.58	3.40	4.56	4.77	5.89	5.07

Figures 5, the six versions implemented can follow reference and reject disturbances in the both tests (reference greater than 1 meter and reference less than 1 meter), however, the PNMPC can also reach the references and reject disturbances more quickly. As it is shown in Table 2 the computational cost is higher due to the complexity of calculation the PNMPC requires. On the other hand, versions 3, 4, 5 and 6 which were part of the non-linearity process are soft; these controllers do manage to follow the reference and reject disturbance, however, more slowly as compared to the complete PNMPC, which has a lower computational cost.

Figure 6 shows that, for significant height variations, the six versions have a good reference tracking, where the complete PNMPC reacts faster than the linear controllers remembering that the computational cost is greater. Comparing the results, it can be observed that the six implemented algorithms have a very similar response, although the complete PNMPC algorithm provides the best results, where all can react properly to the disturbance, and where, the main difference between them being the runtime. This can be attributed to the "soft" non-linearity of the process. Thus, a study of the behavior process can lead to an implementation with gains in terms of processing time without a significantly compromise of the performance. Table 3 shows the comparison of the performance index of MSE (mean squared error) and IAE (integral of the absolute magnitude of the error) for PNMPC versions.

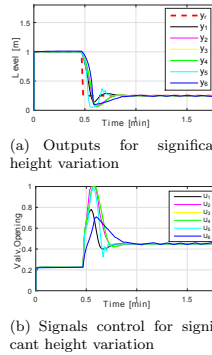


Figure 6: Outputs and signal control for significant height variation

### Acknowledgments

The authors wish to extend his gratitude for the financial support he received from PRH-34 The Agency National of Petroleum (ANP), oil, Gas, and Biofuels.

### References

- Camacho and C., B. (2007). *Model Predictive Control*, segunda edn.
- De Keyser, R. (1998). A gentle introduction to model based predictive control, *PADI2 International Conference on Control Engineering and Signal Processing*.
- García, Carlos E.; Prett, D. M. (1989). Model predictive control: Theory and practice-a survey, *International Federation of Automation Control* **25**(3): 14.
- Normey, J. E. and Camacho, E. F. (2007). *Control of Dead time processes*.
- Plucenio, A. (2010). *Desenvolvimento de técnicas de controle não linear para elevação de fluidos multifásicos*, Dissertação de doutorado-engenharia de automação e sistemas, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- Plucenio, A., Campos, M., Vasconcelos, L. and De Gomes, M. (2014). Model predictive control algorithm with stability step, *Congresso Brasileiro de Automática*.
- Plucenio, A., J., N., D., P. and A., B. (2007). Controle preditivo não linear na indústria do petróleo e gás, *PDPETRO*.
- Qina, S. J. and Badgwell, T. A. (2003). A survey of industrial model predictive control technology, *Control Engineering Practice* **11**: 733-764.

# DMC de forma recursiva aplicando o passo de estabilidade

Bryant Bruce Picon Yang  
 Dto. de Automação e Sistemas  
 Universidade Federal de Santa Catarina  
 Florianópolis, SC - Brasil 88040-900  
 Email: bryant.bruce@posgrad.ufsc.br

Agustinho Plucenio  
 Firenze-Consultoria em Engenharia e informática  
 São José, SC - Brasil 88102-000  
 Email: plucenio@firenzeconsult.com

**Resumo**— Grande parte dos processos existentes na indústria de petróleo e gás utiliza controladores preditivos. A crescente utilização dessas estratégias pode ser atribuída ao fato de elas facilitarem a declaração de problemas de controle relativamente complexos, como os casos de controle de processos multivariáveis, processos integradores, processos instáveis, processos não lineares e processos com atraso de transporte. Este trabalho apresenta uma abordagem que é orientada ao controle de plantas monovariáveis e multivariáveis, com múltiplas incertezas no modelo incluindo ganho, constante de tempo e atraso de transporte, que representam os casos mais comuns na indústria de processos. É implementado o algoritmo MPC com passo de estabilidade, a ação de controle é calculada em dois passos, garantindo estabilidade e robustez na planta, provocando uma melhora na rejeição a perturbações não modeladas. Os resultados são comparados com o algoritmo DMC de forma recursiva, aplicando a filtragem do erro de predição.

## I. INTRODUÇÃO

A técnica de Controle Preditivo Baseado em Modelo (Model Based Predictive Control, MBPC ou Model Predictive Control, MPC), é uma das técnicas de controle moderno mais avançadas. Em sua forma mais geral, aceita qualquer tipo de modelo, funções objetivo ou restrições [1]. Um dos motivos para o êxito desta técnica é a maneira na qual os problemas de controle são formulados no domínio do tempo [2], permitindo que os algoritmos sejam muito mais simples de compreender.

As diferenças entre os diversos algoritmos existentes devem-se basicamente à forma de escolher os modelos de predição para o processo, a forma como as perturbações e os erros de predição são tratados, o tipo de função objetivo e o procedimento para o cálculo do sinal de controle [3]. O modelo de predição é o elemento mais importante dentro do projeto do controlador. Sua importância se deve ao fato de que ele deve ser capaz de representar adequadamente a dinâmica do processo para permitir o cálculo das predições da saída, ser intuitivo e ao mesmo tempo permitir uma análise teórica do sistema.

Em geral, o modelo utilizado para o projeto do controlador MPC é separado em duas partes: O modelo do processo propriamente dito e o modelo das perturbações;

ambos necessários para o cálculo das predições, daí o nome Modelo de Controle Preditivo [7].

O interesse pelos controladores MPC aplicados em controle de processos industriais remete a um conjunto de artigos apresentados no final de 1970. A apresentação do Model Predictive Heuristic Control (MPHC), por [4], e a do Dynamic Matrix Control (DMC), por [5] engenheiros da Shell Oil Co, entre outros artigos, foram marcos iniciais no uso de MPC na indústria. Posteriormente, a capacidade do MPC de alcançar sistemas de controle altamente eficientes, determinando uma condição favorável de operação mesmo em situações adversas, vem mostrando um rápido progresso em pesquisas acadêmicas e aplicações industriais [6].

Um componente comum no comportamento dos processos industriais, principalmente nos processos da indústria de petróleo e gás, é o atraso de transporte. Na prática, tem-se muitos processos industriais onde o atraso é causado pelo transporte de material, energias ou pelo acúmulo de informação no sistema. Nesse caso, o sistema não responde imediatamente. Ele leva algum tempo para exibir em sua saída os efeitos de uma entrada [8].

Alguns trabalhos apresentaram estratégias interessantes para contornar o problema do atraso de transporte e obter melhor desempenho dos controladores. A maioria dessas estratégias são baseadas nas ideias do preditor de Smith [9].

Em [10] foi proposto um trabalho aplicando o algoritmo DMC que é amplamente utilizado nas áreas industriais, um filtro é incluído no modelo de predição do controlador para conseguir rejeição da perturbação e melhorar a robustez, especialmente quando são considerados múltiplos atrasos de transporte e erros no modelo.

O algoritmo DMC é modificado para um DMC de forma recursiva, implementado segundo [5], e é aplicado um filtro no erro de predição partindo da ideia do preditor de Smith.

Segundo [5], as predições de malha aberta em  $t+k$ , dado as informações em  $t$  e  $t+1$  são mostradas nas equações (1) e (2) respectivamente:

$$y_0(t+k|t) = \sum_{j=k+1}^{\infty} g_j \Delta u(t+k-j) \quad (1)$$

$$y_0(t+k|t-1) = \sum_{j=k+2}^{\infty} g_j \Delta u(t+k-j) \quad (2)$$

Onde a diferença entre a predição de malha aberta em  $t+k$  dado  $t$ , e em  $t+k$  dado  $t-1$ , é apenas a nova ação de controle que não é conhecida em  $\Delta \mathbf{u}(t-1)$ . Subtraindo as equações (1) e (2) tem-se a equação (3):

$$\begin{aligned} y_0(t+k|t) - y_0(t+k|t-1) = \\ g_{k+1} \Delta u(t-1) + \sum_{j=k+2}^{\infty} g_j \Delta u(t+k-j) + \\ - \sum_{j=k+2}^{\infty} g_j \Delta u(t+k-j) \end{aligned} \quad (3)$$

então tem-se a equação (4):

$$y_0(t+k|t) = g_{k+1} \Delta u(t-1) + y_0(t+k|t-1) \quad (4)$$

O cálculo do DMC de forma recursiva é feito da seguinte forma: Mantém-se em memória um vetor com  $N$  elementos,  $\mathbf{Y}_{livre} = [y_0(t|t-1), \dots, y_0(t+N-1|t-1)]^T$ , onde  $N$  é chamado de horizonte de modelo. Seus elementos são as respostas livres dadas as ações de controle conhecidas até o instante  $t-1$ . Ao iniciar o controlador DMC no instante  $t_0$ , dado que o sistema encontra-se em regime permanente, considera-se que as predições de malha aberta sejam constantes e iguais ao valor de  $y(t_0)$ .

Ao iniciar a execução do algoritmo DMC no instante  $t$ , é preciso atualizar o vetor da resposta livre, pois o incremento da ação de controle  $\Delta \mathbf{u}(t-1)$  agora é conhecido. O vetor  $\mathbf{Y}_{livre}$  da equação (5) é atualizado.

$$\mathbf{Y}_{livre} = \mathbf{Y}_{livre} + \begin{bmatrix} g_1 \\ g_2 \\ \vdots \\ g_N \end{bmatrix} \Delta \mathbf{u}(t-1) \quad (5)$$

Após obter o novo sinal de controle  $u(t)$ , é preciso deslocar os valores dentro do vetor  $\mathbf{Y}_{livre}$ . Isto porque no instante posterior  $t+1$ , serão requeridas as respostas livres de  $t+1$  até  $t+N$  baseadas nos dados até  $t$ . Então o primeiro elemento é descartado, pois é a predição relativa ao instante  $t$  que já passou. O problema é que, ao se fazer este deslocamento, o último elemento, no caso a predição em  $y(t+N|t)$ , é desconhecida. Porém, se o sistema for estável, pode-se considerar  $y_0(t+N|t) \cong y_0(t+N-1|t)$ . Assim, o novo vetor  $\mathbf{Y}_{livre}$  será dado pela equação (6):

$$\mathbf{Y}_{livre} = \begin{bmatrix} y_0(t+1|t) \\ y_0(t+2|t) \\ \vdots \\ y_0(t+N-1|t) \\ y_0(t+N-1|t) \end{bmatrix} \quad (6)$$

A condição  $y_0(t+N|t) \cong y_0(t+N-1|t)$  é mostrada como:

$$\begin{aligned} y_0(t+N-1|t) &= \sum_{j=N}^{\infty} g_j \Delta u(t+N-1-j) \\ y_0(t+N|t) &= \sum_{j=N+1}^{\infty} g_j \Delta u(t+N-j) = \\ &= \sum_{j=N}^{\infty} g_{j+1} \Delta u(t+N-1-j) \\ &= y_0(t+N-1|t) - \\ &= \sum_{j=N}^{\infty} (g_{j+1} - g_j) \Delta u(t+N-1-j) \end{aligned}$$

Como  $g_{j+1} - g_j \cong 0, \forall i > N$ , tem-se que  $y_0(t+N|t) \cong y_0(t+N-1|t)$ , como se queria demonstrar.

Assim, o cálculo da resposta livre considerada no DMC de forma recursiva, é dado pela equação:

$$\mathbf{f} = \begin{bmatrix} y_0(t+1|t) \\ \vdots \\ y_0(t+p|t) \end{bmatrix} + \mathbf{1}_{p \times 1} (y(t) - y_0(t|t)) \quad (7)$$

No algoritmo DMC o erro entre a saída medida e a saída predita no instante  $k$  é usado para corrigir as predições de cada estado de todos os elementos dentro do horizonte de predição  $p$ . Onde a equação do erro do modelo é:

$$e = y(t) - y_0(t|t) \quad (8)$$

Como é mostrado em [13], a representação das predições considerando o cálculo da resposta livre com o DMC de forma recursiva, é obtida com a equação (9):

$$\tilde{\mathbf{Y}} = \mathbf{f} + \mathbf{1}_{p \times 1} e + \mathbf{G} \Delta \mathbf{u} \quad (9)$$

onde:

- $\tilde{\mathbf{Y}}$  saída predita de dimensão  $p \times 1$ ;
- $\mathbf{f}$  vetor da resposta livre de dimensão  $p \times 1$ ;
- $\mathbf{G}$  matriz jacobiana de  $\tilde{\mathbf{Y}}$  de dimensão  $p \times m$ ;
- $\Delta \mathbf{u}$  vetor de dimensão  $m \times 1$  que contém os incrementos na ação de controle.

Onde  $p$  é o horizonte de predição e  $m$  é o horizonte de controle.

Utilizando esta forma de implementação, os cálculos necessários para obter a resposta livre tornam-se mais simples, não sendo necessário o armazenamento dos incrementos de controle passados. Com este procedimento, só é alterado o modo de calcular a resposta livre, o restante do algoritmo de controle não é modificado.

Como o erro de predição é explícito no DMC de forma recursiva, é possível inserir um filtro de forma simples:

$$\begin{aligned} \tilde{y}(t+k|t) &= \sum_{j=1}^k g_j \Delta u(t+k-j) + \\ &= y_0(t+k|t) + F(z^{-1})(y(t) - y_0(t|t)) \end{aligned} \quad (10)$$

É possível filtrar o erro de predição separadamente e adicioná-lo à predição de malha aberta  $y_o$ .

O filtro discreto fica na forma:

$$e_f(k) = \alpha e_f(k-1) + (1-\alpha)e(k). \quad (11)$$

Com  $0 \leq \alpha < 1$  para o erro filtrado inserido na predição. A equação (9) pode ser reescrita como:

$$\tilde{\mathbf{Y}} = \mathbf{f} + \mathbf{1}_{p \times 1} e_f + \mathbf{G} \Delta \mathbf{u} \quad (12)$$

O efeito do filtro é análogo ao método utilizado no algoritmo DTC-GPC descrito em [11]. A idéia de aplicar um filtro no erro de predição para corrigir as predições, pode ser aplicado a outros algoritmos MPC, como por exemplo: GPC, DMC e PNMPCC dentre as técnicas de controle preditivo não linear proposto por [12], que vem apresentando bom desempenho e vasta possibilidade de aplicação, que se torna interessante quando avalia-se simplicidade.

Existem varias pesquisas sobre algoritmos MPC para modelos de predição com incerteza, nos quais se propõem estratégias para tratar este problema e uma revisão não é o propósito neste artigo.

O uso de filtragem no erro de predição é muito interessante para o controle de plantas com erros de atraso de transporte no modelo. Este método também lida com outros erros, mas os erros de tempo morto são os mais críticos, já que podem facilmente instabilizar processos.

Em geral, o filtro não afeta a resposta do sistema quando o modelo é perfeito, mas sim quando existe erro no atraso de transporte, no ganho e nas constantes de tempo entre a planta real e o modelo de predição. Além disso, muita filtragem no sistema pode resultar numa resposta mais lenta a perturbações.

Neste trabalho, apresenta-se a comparação do método aplicando o filtro no erro de predição e o método proposto por [12], que é aplicado a plantas estáveis em malha aberta com erros no modelo de predição. No método proposto em [12] nenhum filtro é utilizado. Em vez disso, o algoritmo MPC com passo de estabilidade propõe tratar o problema em dois passos:

- 1) Passo de estabilidade
- 2) Passo de cálculo de controle

No passo de estabilização, um sinal será calculado e adicionado ao vetor do erro de predição, estabilizando a planta. O passo de cálculo de controle usará o erro modificado sobre o cálculo do sinal de controle.

O artigo é organizado da seguinte forma. Na Seção II o algoritmo para obter o sinal de correção de erro será derivado. Na Seção III serão mostradas as simulações obtidas aplicando o DMC de forma recursiva com a técnica de filtragem no erro de predição e com a técnica do passo de estabilidade, ambas aplicadas num reator químico com duas variáveis manipuladas e duas variáveis controladas, comumente visto nas indústrias petroquímicas. Para demonstrar a eficácia desta técnica, erros de modelagens serão inseridos. Finalmente, na Seção IV serão apresentadas as respectivas conclusões do trabalho.

## II. DMC DE FORMA RECURSIVA COM PASSO DE ESTABILIDADE PARA COMPENSAÇÃO DE ERROS NO MODELO DE PREDIÇÃO

Uma das possibilidades de tratar os erros de predição é através da aplicação de filtros no erro de predição. Essa estratégia tem a finalidade de melhorar a robustez da planta frente a erros de modelo.

Como mencionado na seção anterior, no algoritmo DMC proposto em [5], a correção de erro aplicados às predições foi assumida constante e igual ao erro actual. A técnica de filtragem descrita na seção 1, aplica um filtro no erro de predição partindo da ideia do preditor de Smith.

Uma outra proposta é a descrita em [13], onde se busca um sinal que, somado ao erro verdadeiro, proporcione robustez e estabilidade em plantas com erros de modelo e um melhor desempenho frente a perturbações. Esta solução é a derivada do erro de correção  $\mathbf{E}_L$  que é somado ao vetor de erro  $\mathbf{E}$ .

Uma primeira opção para obter  $\mathbf{E}_L$  é utilizar  $p$  possíveis diferentes elementos. No entanto, isso pode trazer um elevado custo computacional. Assim, apresentamos-se outras opções para o projeto do vetor  $\mathbf{E}_L$  conforme [13]:

- a: aplicando um horizonte de correção de erro  $p_L = p - \Delta p$  e  $\Delta p$  elementos nulos;
- b: aplicando um horizonte de correção de erro  $p_L$  e repetindo o último elemento do horizonte para todos os  $p - p_L$  elementos seguintes;
- c: usando o conceito de blocking, onde  $\mathbf{E}_L$  é preenchido com  $n_L$  grupo de diferentes valores possíveis;
- d: preenchendo o vetor de tamanho  $p - \Delta p$  com um único valor de  $\mathbf{E}_L$  e  $\Delta p$  elementos nulos.

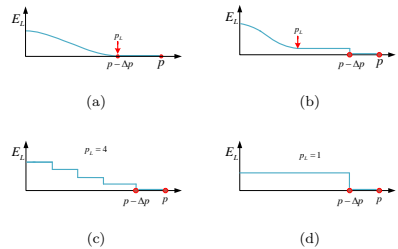


Fig. 1. Formas de implementar  $\mathbf{E}_L$

Para evitar que o requisito de estabilidade perturbe o seguimento da referência quando se tem  $\mathbf{E}_L$  definido em todos os elementos  $p$ , é importante ter alguns dos últimos componentes de  $\mathbf{E}_L$  nulos. Isso tem o significado de dizer ao algoritmo que a correção total deve convergir para aquela utilizada pelo DMC de Cutler no final do horizonte de predição.

Qualquer que seja a opção escolhida dentre as mencionadas anteriormente, o vetor final de estabilização da correção do erro é expresso por:

$$\mathbf{E}_s = \mathbf{E} + \mathbf{M} * \mathbf{E}_L \quad (13)$$

onde:

$\mathbf{E}_L$ : é o vetor com todas correções do erro de Lyapunov das variáveis controladas  $[E_L^1 \ E_L^2 \ \dots \ E_L^m]^T$ ,

$\mathbf{E}$ : é o vetor com todos os erros verdadeiros das variáveis controladas  $[E^1 \ E^2 \ \dots \ E^m]^T$ ,

$\mathbf{E}_s$ : é o vetor com todos os erros modificados,  
 $\mathbf{M}$ : é uma matriz com dimensão  $\sum_m^1 p \times \sum_m^1 p$ . É composto de uns e zeros, de tal maneira que implementa as estratégias mostradas na Figura 1.

Uma outra opção para obter o vetor  $\mathbf{E}_L$ , é aplicar o segundo método de Lyapunov para estabilidade. Esse método se baseia na analogia de uma função de Lyapunov  $V(x)$  com energia ou potencial  $\mathbf{E}_L$  [14]. Se  $x$  é um ponto que tem um equilíbrio em 0, então se existe um  $V(x)$  tal que:

$$\begin{aligned} V(x) &> 0 \quad \text{para qualquer } x \neq 0 \quad \text{e} \\ V(x) &= 0 \quad \text{só para } x = 0 \end{aligned} \quad (14)$$

Então, se:

$$\frac{dV(x)}{dt} \leq 0 \quad (15)$$

$x$  é estável no sentido de Lyapunov. Se  $\frac{dV(x)}{dt} \leq 0$  e  $\frac{dV(x)}{dt} = 0$ . Só em caso de  $x = 0$  o sistema é assintoticamente estável. Este é um resultado muito bem aceito e a base para muitos estudos de estabilidade [13].

Para plantas estáveis em malha aberta, onde a instabilidade é causada pela ação de controle devido a erros do modelo de predição, é natural considerar como uma função de Lyapunov:

$$L = \frac{1}{2} \Delta u^T \Delta u \quad (16)$$

Observe que  $L$  é uma função Lyapunov candidata, válida desde que  $\Delta u = 0$  seja a condição de equilíbrio para:

$$\begin{aligned} L(0) &= 0 \quad \text{e} \\ L &> 0 \quad \text{para qualquer } \Delta u \neq 0 \end{aligned} \quad (17)$$

Considerando um caso irrestrito, onde:

$$\Delta \mathbf{u} = \mathbf{K} (\mathbf{W} - \mathbf{F})$$

$$\mathbf{K} = (\mathbf{G}^T \mathbf{R} \mathbf{G} + \mathbf{Q})^{-1} \mathbf{G}^T \mathbf{R} \quad (18)$$

$$\mathbf{F} = \mathbf{F}_{sc} + \mathbf{E} + \mathbf{M} \mathbf{E}_L$$

Na equação (18),  $\mathbf{W}$  representa o vetor de trajetórias desejadas dentro do horizonte de predição,  $\mathbf{F}$  é a resposta livre corrigida no horizonte de predição,  $\mathbf{R}$  e  $\mathbf{Q}$  são as

matrizes de ponderação do set-point e de supressão de movimento,  $\mathbf{F}_{sc}$  é a resposta livre sem correção ao longo do horizonte de predição  $p$ ,  $\mathbf{E}$  é o vetor do erro de predição no instante atual,  $\mathbf{E}_L$  é o vetor do erro de estabilidade de Lyapunov, que é calculado para corrigir o erro de predição.

A fim de assegurar estabilidade, deve-se cumprir  $\frac{\partial L}{\partial t} \leq 0$ .  $\frac{\partial L}{\partial t}$  pode ser calculado como:

$$\frac{\partial L}{\partial t} = \frac{\partial L}{\partial \Delta \mathbf{u}} \frac{\partial \Delta \mathbf{u}}{\partial t} \frac{\partial \mathbf{E}_L}{\partial t}, \text{ ou} \quad (19)$$

$$\frac{\partial L}{\partial t} = \Delta \mathbf{u}^T (-\mathbf{K} \mathbf{M}) \frac{\partial \mathbf{E}_L}{\partial t}$$

Fazendo substituições:

$$\frac{\partial L}{\partial t} = -(\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{M} \mathbf{E}_L)^T \mathbf{K}^T \mathbf{K} \mathbf{M} \frac{\partial \mathbf{E}_L}{\partial t} \quad (20)$$

O termo  $\frac{\partial \mathbf{E}_L}{\partial t}$  pode ser reescrita como  $\mathbf{E}_L^k - \mathbf{E}_L^{k-1}$ .

Assim, a expressão final para a equação (20) torna-se:

$$\frac{\partial L}{\partial t} = \frac{1}{2} \mathbf{E}_L^T \Phi_L \mathbf{E}_L - \frac{1}{2} \Theta_L^T \mathbf{E}_L + \frac{1}{2} \mathbf{D}$$

$$\Phi_L = \mathbf{M}^T \mathbf{K}^T \mathbf{K} \mathbf{M}$$

$$\mathbf{A} = \mathbf{W} - \mathbf{F}_{sc} - \mathbf{E}$$

$$\mathbf{B} = \mathbf{M}^T \mathbf{K}^T \mathbf{K}$$

$$\Theta_L = \Phi_L \mathbf{E}_L^{k-1} + \mathbf{B} \mathbf{A}$$

$$\mathbf{D} = \mathbf{A}^T \mathbf{B}^T \mathbf{E}_L^{k-1}$$

Tendo a expressão para  $\frac{\partial L}{\partial t}$ , pode-se formular o seguinte problema de otimização:

$$\begin{aligned} \min_{\mathbf{E}_L} J \\ \text{sujeito a :} \end{aligned} \quad (22)$$

$$J = \frac{1}{2} \mathbf{E}_L^T \Phi_L \mathbf{E}_L - \frac{1}{2} \Theta_L^T \mathbf{E}_L + \frac{1}{2} f_{DP}(\mathbf{E}_L)$$

Os dois primeiros termos da equação (22) representam a expressão  $\frac{\partial L}{\partial t}$ , exceto para o termo  $\mathbf{D}$  não relacionado com  $\mathbf{E}_L$ . A ideia é que o valor encontrado de  $\mathbf{E}_L$  obrigue o termo  $\frac{\partial L}{\partial t} \leq 0$  a ficar negativo. Porém, isto deve ter um limite, e essa é a razão da presença do termo  $f_{DP}(\mathbf{E}_L)$ . Uma função positiva definida sobre  $\mathbf{E}_L$  deve ser concebida com o objetivo de se contrapor ao termo negativo de  $J$ . Além disso, a função deve incluir ponderações para ajustar uma melhor sintonia do controlador. A primeira função candidata é  $\mathbf{E}_L^T \mathbf{M}^T \mathbf{S}_L \mathbf{M} \mathbf{E}_L$ . Os valores da diagonal da matriz de ponderação  $\mathbf{S}_L$  atuam com o objetivo de suprimir  $\mathbf{E}_L$ , a fim de evitar que o valor  $\frac{\partial L}{\partial t}$  seja negativo. No entanto, o papel desta função é limitado a suprimir os valores negativos em  $J$  e não contribui para qualquer outro atributo do sistema sob controle.

Outra função candidata definida positiva é  $(\tilde{\mathbf{Y}} - \mathbf{W})^T \mathbf{S}_L (\tilde{\mathbf{Y}} - \mathbf{W})$ . Onde  $\tilde{\mathbf{Y}} = \mathbf{F}_{sc} + \mathbf{E} +$

$\mathbf{ME}_L + \mathbf{GK}(\mathbf{W} - (\mathbf{F}_{sc} + \mathbf{E} + \mathbf{ME}_L))$ . O objetivo desta função é suprimir os valores negativos de  $J$ , estabelecendo uma relação entre o principal objetivo, que é evitar  $\frac{\partial J}{\partial \mathbf{E}_L}$  seja negativo, garantindo o seguimento da referência. Os elementos positivos da matriz diagonal  $\mathbf{S}_L$  são ajustados a fim de obter  $\mathbf{E}_L$  que proporcione estabilidade e seguimento da referência para o sistema sob controle.

Finalmente, o problema de otimização quadrática para obter  $\mathbf{E}_L$  é formulado como:

$$\begin{aligned} & \min_{\mathbf{E}_L} J \\ & \text{sujeito a:} \\ & J = \frac{1}{2} \mathbf{E}_L^T \Phi \mathbf{E}_L - \frac{1}{2} \Theta^T \mathbf{E}_L \end{aligned} \quad (23)$$

$$\Phi = \mathbf{M}^T (\mathbf{K}^T \mathbf{K} + \mathbf{Z}^T \mathbf{S}_L \mathbf{Z}) \mathbf{M}$$

$$\mathbf{Z} = \mathbf{GK} - \mathbf{I}_{\sum_{i=1}^p p \times \sum_{i=1}^n p}$$

$$\Theta^T = \Phi \mathbf{E}_L \mathbf{E}_L^{k-1} + \mathbf{BA} + \mathbf{M}^T \mathbf{Z}^T \mathbf{S}_L \mathbf{ZA}$$

Para o caso sem restrições, o valor do vetor  $\mathbf{E}_L$  que minimiza  $J$  é:

$$\mathbf{E}_L = 0.5 \Phi^{-1} \Theta \quad (24)$$

A matriz  $\Phi$  será definida positiva, desde que a matriz de ponderação diagonal  $\mathbf{S}_L$  seja escolhida com valores positivos.

Um aspecto interessante desta metodologia é que restrições em  $\Delta \mathbf{u}$  e  $\mathbf{u}$  podem ser atendidas na etapa de estabilidade. Introduzindo restrições no sistema, os seguintes termos se podem escrever como:

$$\Delta \mathbf{u} \leq \mathbf{B} \quad \text{é equivalente a:}$$

$$\mathbf{K}(\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{ME}_L) \leq \mathbf{B} \quad (25)$$

ou :

$$-\mathbf{KME}_L \leq \mathbf{B} - \mathbf{K}(\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E})$$

O vetor  $\mathbf{u}$  pode ser escrito como:  $\mathbf{u} = \mathbf{u}_{k-1} \mathbf{1}_{m \times 1} + \mathbf{T}_I \Delta \mathbf{u}$ , onde  $\mathbf{T}_I$  é a matriz Toeplitz integradora, e  $\mathbf{1}_{m \times 1}$  é um vetor de uns para um caso SISO pelo que se torna simple de implementar como é mostrado:

$$\mathbf{u} \leq \mathbf{C} \quad \text{é equivalente a:}$$

$$\mathbf{u}_{k-1} \mathbf{1}_{m \times 1} + \mathbf{T}_I \mathbf{K}(\mathbf{W} - \mathbf{F}_{sc} - \mathbf{E} - \mathbf{ME}_L) \leq \mathbf{C}$$

$$\text{ou:} \quad (26)$$

$$\mathbf{T}_I \mathbf{KME}_L \leq \mathbf{C} - \mathbf{u}_{k-1} \mathbf{1}_{m \times 1} - \mathbf{T}_I \mathbf{KA}$$

$$\mathbf{A} = \mathbf{W} - \mathbf{F}_{sc} - \mathbf{E}$$

Ao introduzir estas restrições na programação quadrática, dada pela equação (23), o valor de  $\mathbf{E}_L$  obtido pode ser usado na equação (27) para derivar  $\Delta \mathbf{u}$ .

$$\Delta \mathbf{u} = \mathbf{K}(\mathbf{W} - \mathbf{F}) \quad (27)$$

$$\mathbf{F} = \mathbf{F}_{sc} + \mathbf{E} + \mathbf{ME}_L$$

O vetor  $\Delta \mathbf{u}$  obtido provará estabilidade ao sistema respeitando as restrições impostas em  $\Delta \mathbf{u}$  e  $\mathbf{u}$ . O resultado é interessante, uma vez que resolve uma dificuldade que é a de garantir a estabilidade com restrições.

### III. SIMULAÇÕES E RESULTADOS

Nesta seção, os resultados de simulação são apresentados aplicando as técnicas discutidas nas seções anteriores. O algoritmo DMC de forma recursiva será implementado num reator de tanque agitado, como mostrado na Figura 2 e descrito pelo modelo com a seguinte matriz de transferência (as constantes de tempo são expressos em minutos) [11]:

$$\begin{bmatrix} Y_1(s) \\ Y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{e^{-0.24s}}{1+0.7s} & \frac{5e^{-0.24s}}{1+0.3s} \\ \frac{e^{-0.09s}}{1+0.5s} & \frac{2e^{-0.09s}}{1+0.4s} \end{bmatrix} \begin{bmatrix} U_1(s) \\ U_2(s) \end{bmatrix}$$

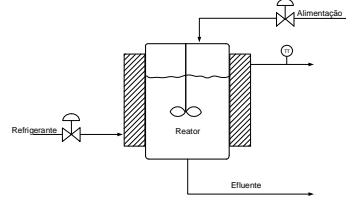


Fig. 2. Reator de tanque agitado

As variáveis manipuladas  $U_1(s)$  e  $U_2(s)$  são respectivamente a vazão de alimentação e a vazão do fluido refrigerante no revestimento. As variáveis controladas  $Y_1(s)$  e  $Y_2(s)$  são a concentração do efluente e a temperatura do reator, respectivamente. Como se pode observar na matriz, existe influência entre as variáveis controladas, já que o sistema é acoplado.

Utilizando um tempo de amostragem de  $T_s = 0,03\text{min}$  o modelo discretizado é mostrado na seguinte matriz com funções de transferência discretas:

$$\begin{bmatrix} y_1(t) \\ y_2(t) \end{bmatrix} = \begin{bmatrix} \frac{0.0420z^{-1}}{1-0.9580z^{-1}} z^{-8} & \frac{0.4758z^{-1}}{1-0.9048z^{-1}} z^{-8} \\ \frac{0.0582z^{-1}}{1-0.9418z^{-1}} z^{-3} & \frac{0.1445z^{-1}}{1-0.9277z^{-1}} z^{-3} \end{bmatrix} \begin{bmatrix} u_1(t) \\ u_2(t) \end{bmatrix}$$

Os tempos mortos neste caso são causadas por efeitos de medição. O tempo morto na temperatura de saída  $y_2(t)$  é o tempo necessário para transportar o fluido para o sensor.

O tempo morto na concentração  $y_1(t)$  é causada pelo efeito de transporte e o tempo requerido pelo analisador. Os tempos mortos mínimos são:  $d_1 = 8$  e  $d_2 = 3$ .

Implementando o DMC de forma recursiva com a técnica de passo de estabilidade, foi percebido que considerando só a matriz de ponderação  $\mathbf{S}_L$ , o valor desta tinha que ser muito pequeno, com o que  $\Phi$  da equação (23), resultava em valores muito pequenos. Tentando melhorar isso, foi considerada a incorporação de uma outra matriz de ponderação denominada  $\mathbf{P}_L$ , em  $\Phi$  e  $\mathbf{B}$  como se mostra na equação (28):

$$\begin{aligned} \mathbf{B} &= \mathbf{M}^T \mathbf{K}^T \mathbf{P}_L \mathbf{K} \\ \Phi &= \mathbf{M}^T (\mathbf{K}^T \mathbf{P}_L \mathbf{K} + \mathbf{Z}^T \mathbf{S}_L \mathbf{Z}) \mathbf{M} \end{aligned} \quad (28)$$

A Figura 3 mostra as respostas ao degrau com seus respectivos atrasos de transporte para cada função de transferência.

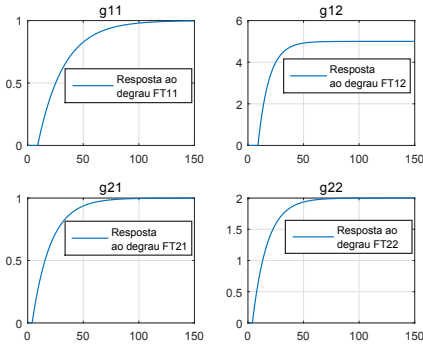


Fig. 3. Resposta ao degrau das funções de transferência

A Figura 4 mostra os erros no atraso de transporte e no ganho, entre a planta real e o modelo de predição. Como pode-se observar nesta figura, no primeiro teste, não foram considerados erros de modelo, no atraso de transporte nem no ganho. As resposta do sistema sem erros, é mostrado na figura 5. Pode-se observar que aplicando as duas técnicas, se obtém um bom seguimento da referência e rejeição das perturbações não modeladas. Mas se percebe que a utilização do filtro no erro de predição deixa o resposta lenta em comparação como a resposta aplicando a idéia do passo de estabilidade.

Todas as funções de transferência são sistemas de primeira ordem com atraso de transporte. Um outro teste é mostrado na figura 5, à esquerda os atrasos de transporte utilizados no modelo de predição e os utilizados para simular a planta real. Na direita mostra a relação dos erros no ganho entre a simulação da planta real e os erros nos ganhos do modelo de predição. Como observado nesta figura, a distância existente entre o quadrado e o círculo representa a incerteza introduzida no sistema.

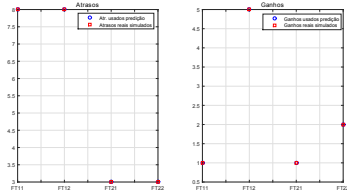


Fig. 4. Erros usados na simulação teste I

Os horizontes de predição  $p_1$  e  $p_2$  para  $y_1(t)$  e  $y_2(t)$  são 50 e 40, respectivamente. O horizonte de controle  $m_1$  e  $m_2$  é de 10. O vetor  $\mathbf{E}_L$  é implementado usando a estrutura  $a)$  da Figura 1, com um horizonte de erro  $\mathbf{p}_L = 5$ .

As ponderações dos pesos no controle e no seguimento da referência, utilizando o DMC de forma recursiva e aplicando o filtro no erro de predição são:  $\lambda_1 = 2$ ,  $\lambda_2 = 10$  e  $\delta_1 = 0,01$ ,  $\delta_2 = 0,1$  e um valor de  $\alpha_1$  e  $\alpha_2$  de 0,75 para o filtro no erro de predição da equação (11).

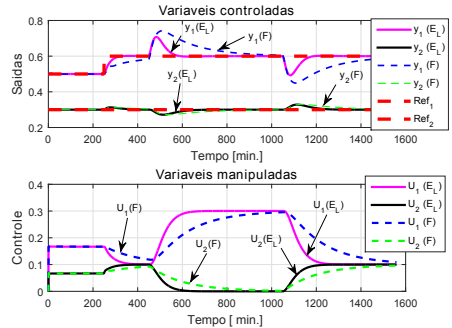


Fig. 5. Resultado da simulação teste I

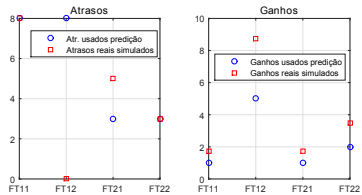


Fig. 6. Erros usados na simulação teste II

A abordagem do passo de estabilidade garante a estabilidade no sistema. As ponderações dos pesos no controle e no seguimento da referência no DMC de forma recursiva

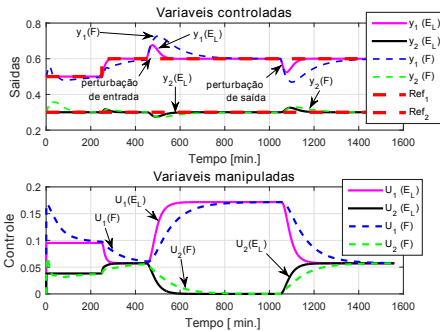


Fig. 7. Resultado da simulação teste II

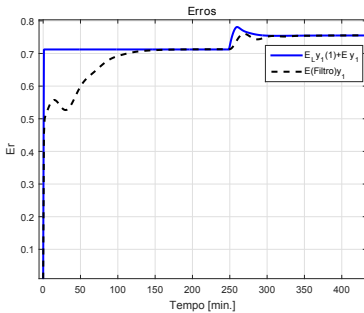


Fig. 8. Comparação dos erros

aplicando o passo de estabilidade, são:  $\lambda_1 = 0,01$ ,  $\lambda_2 = 2$  e  $\delta_1 = 0,01$ ,  $\delta_2 = 0,1$ .

Na Figura 7 se mostra a resposta do sistema frente aos erros apresentados na Figura 6. Na Figura 7, uma perturbação não modelada é introduzida no minuto 450, mantendo-se até o minuto 1050. Como pode se observar, a técnica de passo de estabilidade mostra um melhor seguimento da referência, comparada com a utilização de filtragem no erro de predição, quando existem erros de modelo tanto para o ganho como para o atraso de transporte. Por outro lado, a resposta na rejeição da perturbação é mais rápida aplicando o passo de estabilidade, já que o uso de muito filtragem no sistema pode causar lentidão na rejeição da perturbação.

A Figura 8, mostra a comparação dos erros entre as duas técnicas apresentadas, que é o erro filtrado para a variável controlada  $y_1(t)$  e o primeiro componente obtido adicionando o erro verdadeiro de  $y_1(t)$  com o primeiro componente do erro de Lyapunov  $E_L(1)$ . O comportamento do erro de Lyapunov para a variável controlada

TABELA I. Erro quadrático médio

Estratégia	EQM1	EQM2
DMC <sub>F</sub>	$1,02 \times 10^3$	284,90
DMC <sub>E<sub>L</sub></sub>	590,66	140,31

$y_1(t)$ , é muito mais agressivo do que o erro filtrado, de modo que compensa o erro de tempo morto no momento da mudança da referência. Como esperado, os erros no sistema em estado estacionário coincidem.

#### IV. CONCLUSÕES

Com este trabalho tem-se outra opção para lidar com erros de modelagem em plantas estáveis de malha aberta, que é baseado no segundo método de Lyapunov para a estabilidade.

O passo de estabilidade modifica o erro de predição futuro de forma otimizada, com o objetivo de proporcionar estabilidade no sistema. Com o intuito de prover estabilidade o algoritmo encontra a correção necessária à predição.

O tempo que leva para a técnica passo de estabilidade aplicado no reator de tanque agitado para trazer  $y_1(t)$  à referência, é cerca da metade do tempo que é gasto com a técnica de filtragem, como se mostra na Figura 7, pelo que esta técnica mostrou um desempenho superior em comparação com o método de filtragem no erro de predição, tanto no seguimento da referência como na rejeição das perturbações não modeladas, o que da para perceber na tabela I, onde se detalha o cálculos dos erros quadráticos médios para cada estratégia.

Este é um resultado interessante, pois em muitas plantas principalmente nas áreas de petróleo e gás, a maioria das variáveis afetadas por perturbações não modeladas tem um impacto econômico importante e afetam diretamente no desempenho do processo. Quanto mais tempo se encontram fora das faixas aceitáveis de operação, maior é a perda econômica.

#### AGRADECIMENTOS

Os autores deste trabalho agradecem pelo apoio financeiro recebido do programa de formação de recursos humanos PHR-34 da Agência Nacional do Petróleo, Gás e Biocombustíveis.

#### REFERÊNCIAS

- [1] Camacho, E. and Bordons, C. (2007). *Model Predictive Control*, segunda edn. Springer.
- [2] Bordons, C. *Control Predictivo: metodología, tecnología y nuevas perspectivas*, 2000.
- [3] Plucenio, A. et al. *Control preditivo não linear na indústria do petróleo e gás*. PDPETRO, Outubro 2007.
- [4] Richalet, J. and Rault, A. and Testud, J. (1978), *Model Predictive Heuristic Control: Application to industrial process*. AUTOMATICA.
- [5] Cutler, C. and Remarker, B. (1980), *Dynamic Matrix Control-A Computer Control Algorithm*. In Automatic Control, Conference, San Francisco.
- [6] Qina, S. and Badgwell, T. (2003). *A survey of industrial model predictive control technology*, Control Engineering Practice.



- [7] García, C. and Prett, D. (1989). *Model predictive control: Theory and practice-a survey*, International Federation of Automation Control.
- [8] Normey, J. (2001). *GPC adaptativo basado en el predictor de Smith para procesos con retardo variable*, Congresso Brasileiro de Automática.
- [9] Smith, O. (1957). *Closer Control of Loops with Dead-Time*, Chem. Eng. Progress.
- [10] Lima, D. and Normey, J. and Plucenio, A. et al. (2014). *Improving Robustness and Disturbance Rejection Performance with Industrial MPC*, Congresso Brasileiro de Automática.
- [11] Normey, J. and Camacho, E. (2007). *Control of Dead time Processes*. Springer.
- [12] Plucenio, A. (2010). *Desenvolvimento de técnicas de controle não linear para elevação de fluidos multifásicos*, Dissertação de Doutorado-Engenharia de Automação e Sistemas, Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis.
- [13] Plucenio, A. and Campos, M. et al. (2014). *Model Predictive Control Algorithm with Stability Step*, Congresso Brasileiro de Automática.
- [14] Monteiro, L. (2006). *Systemas Dinâmicos*. Livraria da Física.