

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CONTROLE MPC MULTIVARIÁVEL COM RESTRIÇÕES
USANDO FUNÇÕES DE LAGUERRE

TARCISIO CARLOS FARIAS PINHEIRO

DM: 10/2018

UFPA/ITEC/PPGEE
CAMPUS UNIVERSITÁRIO DO GUAMÁ
BELÉM – PARÁ – BRASIL
2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

TARCISIO CARLOS FARIAS PINHEIRO

CONTROLE MPC MULTIVARIÁVEL COM RESTRIÇÕES
USANDO FUNÇÕES DE LAGUERRE

DM: 10/2018

UFPA/ITEC/PPGEE
CAMPUS UNIVERSITÁRIO DO GUAMÁ
BELÉM – PARÁ – BRASIL
2018

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

CONTROLE MPC MULTIVARIÁVEL COM RESTRIÇÕES
USANDO FUNÇÕES DE LAGUERRE

TARCISIO CARLOS FARIAS PINHEIRO

Orientado: Prof. Dr. Antonio da Silva Silveira

Dissertação submetida à Banca Examinadora do Programa de Pós-Graduação em Engenharia Elétrica da UFPA para obtenção do Grau de Mestre em Engenharia Elétrica na área de Sistemas de Energia.

UFPA/ITEC/PPGEE
CAMPUS UNIVERSITÁRIO DO GUAMÁ
BELÉM – PARÁ – BRASIL
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema de Bibliotecas da Universidade Federal do Pará
Gerada automaticamente pelo módulo Ficat, mediante os dados fornecidos pelo(a) autor(a)

- P654c Pinheiro, Tarcisio Carlos Farias
CONTROLE MPC MULTIVARIÁVEL COM RESTRIÇÕES USANDO FUNÇÕES DE LAGUERRE /
Tarcisio Carlos Farias Pinheiro. — 2018
138 f. : il. color
- Dissertação (Mestrado) - Programa de Pós-graduação em Engenharia Elétrica (PPGEE), Instituto de
Tecnologia, Universidade Federal do Pará, Belém, 2018.
Orientação: Prof. Dr. Antonio da Silva Silveira
1. Controle Preditivo Baseado em Modelo. 2. Funções de Laguerre. 3. Programação Quadrática de
Hildreth. 4. Controle Multivariável. I. Silveira, Antonio da Silva, *orient.* II. Título
-

CDD 621.3191

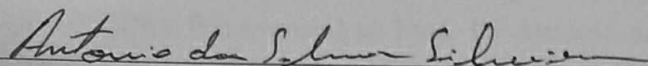
**“CONTROLE MPC MULTIVARIÁVEL COM RESTRIÇÕES USANDO FUNÇÕES
DE LAGUERRE”**

AUTOR: TARCISIO CARLOS FARIAS PINHEIRO

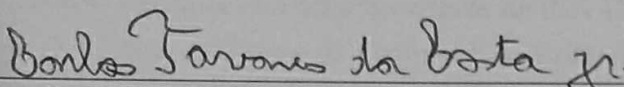
DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO
COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA NA ÁREA DE SISTEMAS DE ENERGIA ELÉTRICA.

APROVADA EM: 01/03/2018

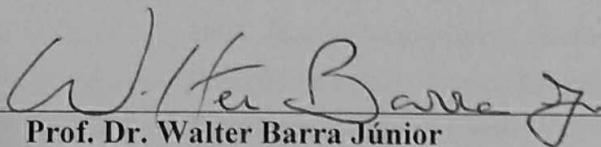
BANCA EXAMINADORA:



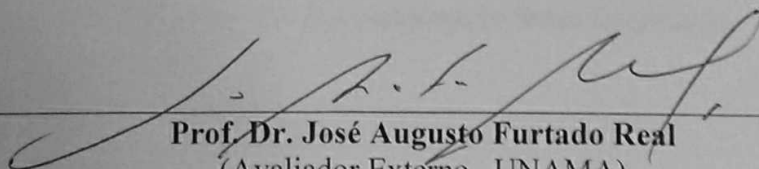
Prof. Dr. Antonio da Silva Silveira
(Orientador – PPGEE/UFPA)



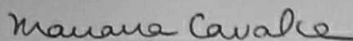
Prof. Dr. Carlos Tavares da Costa Júnior
(Avaliador Interno – PPGEE/UFPA)



Prof. Dr. Walter Barra Júnior
(Avaliador Interno – PPGEE/UFPA)

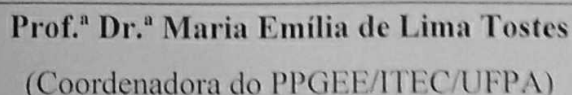


Prof. Dr. José Augusto Furtado Real
(Avaliador Externo – UNAMA)



Prof.ª Dr.ª Mariana Santos Matos Cavalca
(Avaliadora Externa – UDESC)

VISTO:



Prof.ª Dr.ª Maria Emília de Lima Tostes
(Coordenadora do PPGEE/ITEC/UFPA)

AGRADECIMENTOS

Quero agradecer a todos os meus familiares, em especial a minha mãe, Dona Lili, o qual teve papel de suma importância na minha vida através de seu apoio moral, de seus incentivos e cuidados nos momentos difíceis que aparecem em nossas vidas.

Aos professores do curso de Graduação em Engenharia de Controle e Automação do Instituto de Ensino Superior da Amazônia (IESAM). Em especial para o Prof. Máx Ricardo Pantoja da Trindade e ao Prof. Dr José Augusto Furtado Real, pela dedicação em ensinar os princípios, bases e fundamentos da teoria de controle de sistemas.

Aos professores do Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará (UFPA). Em especial ao Prof. Dr Antonio da Silva Silveira pela orientação, disponibilidade, paciência e pela amizade e acima de tudo por ter acreditado em meu potencial.

A FAPESPA (Fundação Amazônica de Amparo a Estudo e Pesquisa) que financiou integralmente as pesquisas relacionadas a este trabalho através do ICAAF 049/2016. Ao Programa de Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará pelo acesso aos meios essenciais para o desenvolvimento desta dissertação.

A todos os meus amigos de pesquisa do Laboratório de Controle e Sistemas (LACOS): Leiliane Borges Cunha, Mauro Gomes da Silva, Bruno Gomes Dutra, Anderson de França Silva, Maryson da Silva Araújo, Luís Augusto Mesquita de Castro, Carlos Eduardo Durans Nogueira, Felipe Martins Resende de Almeida, pela amizade, pelos momentos de descontração, pelos conselhos e pelas conversas muito produtivas, inclusive na fila do RU (Restaurante Universitário) e nos lanches no Ver-o-Pesinho durante a concepção dessa dissertação.

“Não creio que haja uma emoção, mais intensa para um inventor do que ver suas criações funcionando. Essa emoção faz você esquecer de comer, de dormir, de tudo.”
(Nikola Tesla)

RESUMO

Neste trabalho apresenta-se o projeto de um controlador preditivo multivariável baseado em modelo, com restrições, usando as Funções de Laguerre, tendo o intuito de demonstrar os benefícios e facilidades da aplicação deste tipo de controlador em sistemas MIMO (Multiple-Input and Multiple-Output) com restrições. O controlador proposto apresenta a vantagem de diminuir a carga computacional utilizada para resolver o problema de otimização envolvido no projeto, isto porque utiliza uma rede de filtros de funções ortonormais de Laguerre para obter a trajetória futura do sinal de controle dentro de um horizonte de predição, além de melhorar o compromisso entre a viabilidade do sinal de controle e o desempenho de malha fechada do sistema para os casos com restrições, no qual as funções de Laguerre são utilizadas em conjunto com a Programação Quadrática de Hildreth para encontrar a solução ótima do sinal de controle com restrições. Este controlador apresenta grandes vantagens se comparado com o controle preditivo baseado em modelo em sua abordagem clássica, em que os operadores de avanço de tempo são utilizados para prever a trajetória futura do sinal de controle, o que leva à soluções, em alguns casos, pouco satisfatórias, e a uma alta carga computacional para casos onde o sinal de controle requer um longo horizonte de predição e um alto desempenho em malha fechada. Este trabalho também relata testes experimentais com um manipulador robótico configurado como um sistema MIMO com três entradas e três saídas e testes simulados com a coluna de destilação binária de Wood e Berry que é um sistema MIMO com duas entradas e duas saídas, contendo atrasos de transporte. Os testes têm como objetivo comparar os resultados do controlador apresentado com o controlador que usa a abordagem tradicional e com isso demonstrar as vantagens do método usando as funções de Laguerre e sua eficiência para sistemas MIMO.

Palavras-chave: Controle Preditivo Baseado em Modelo, Funções de Laguerre, Programação Quadrática de Hildreth, Controle Multivariável.

ABSTRACT

This work presents a constrained multivariable model predictive controller using Laguerre Functions. This controller uses a set of orthonormal Laguerre networks for representation of the control trajectory within a control horizon. In order to demonstrate the advantages of applying this type of controller in MIMO (Multiple-Input and Multiple-Output) systems, the Laguerre Functions are used to decrease the computational load used to calculate the optimal control. In addition, It improves the compromise between control signal viability and closed-loop performance of the system. The Laguerre Functions are also used in conjunction with Hildreth's Quadratic Programming to find the optimal solution for the case where the control signal is constrained. The proposed controller presents advantages when compared to the classical model predictive control approach, where forward shift operators are used to predict the future trajectory of the control signal, leading to unsatisfactory solutions and a high computational load for cases where the control signal demands a long prediction horizon and a high closed-loop performance. It is also reported the practical testes with a robotic manipulator configured as a MIMO system with three inputs and three outputs and tests simulated with the Wood and Berry binary distillation column which is a MIMO system with two inputs and two outputs, also containing transport time delays. The tests aim to compare the controller results presented with the traditional predictive control approach and thereby demonstrate the advantages of the method using the Laguerre functions and their efficiency for MIMO systems.

Key-words: Model predictive control, Laguerre Functions, Hildreth's Quadratic Programming, Multivariable control.

LISTA DE ILUSTRAÇÕES

Figura 1 – Estrutura do Controlador MPC.	28
Figura 2 – Estratégia do Controlador MPC.	28
Figura 3 – Resposta de $G_{11}(z)$ para a variação de N_p	43
Figura 4 – Resposta de $G_{11}(z)$ para a variação de N_c	44
Figura 5 – Resposta de $G_{11}(z)$ para a variação de Γ	45
Figura 6 – Controlador <i>CDMPC</i> no espaço de estados.	45
Figura 7 – Fluxograma do projeto do Controlador <i>CDMPC</i>	46
Figura 8 – Fluxograma do projeto do controlador <i>LDMPC</i>	57
Figura 9 – Diagrama de blocos do projeto do controlador <i>LDMPC</i>	58
Figura 10 – Resposta para a variação de a com N pequeno.	60
Figura 11 – Resposta para a variação de a com N grande.	60
Figura 12 – Comparação desempenho entre os controladores <i>LDMPC</i> e <i>CDMPC</i> para os mesmos parâmetros de sintonia.	62
Figura 13 – Controladores <i>LDMPC</i> e <i>CDMPC</i> com o mesmo desempenho.	63
Figura 14 – Conflito entre viabilidade do sinal de controle e desempenho de malha fechada.	66
Figura 15 – Fluxograma da programação quadrática de Hildreth.	69
Figura 16 – Saída simulada do sistema: com restrição.	73
Figura 17 – Sinal de Controle: com restrição.	74
Figura 18 – Sinal de Controle incremental: com restrição.	74
Figura 19 – Manipulador robótico e diagrama do Mentor Desktop robot.	78
Figura 20 – Circuito de acionamento e aquisição de dados.	79
Figura 21 – Simulações das saídas do Mentor: sem restrição.	81
Figura 22 – Simulações dos sinais de controle do Mentor: sem restrição.	81
Figura 23 – Simulações dos sinais de controle incremental do Mentor: sem restrição.	82
Figura 24 – Polos de malha fechada do Mentor.	82
Figura 25 – Simulações das saídas do Mentor: com restrição.	83
Figura 26 – Simulações dos sinais de controle do Mentor: com restrição.	83
Figura 27 – Simulações dos sinais de controle incremental do Mentor: com restrição.	84
Figura 28 – Sinal de saída dos testes experimentais do Mentor: com restrição.	86
Figura 29 – Sinal de controle dos testes experimentais do Mentor: com restrição.	86
Figura 30 – Sinal de controle incremental dos testes experimentais do Mentor: com restrição.	87

Figura 31 – Coluna de destilação binária de WB.	89
Figura 32 – Simulações de WB da saída do topo e da base: sem restrição.	92
Figura 33 – Simulações de WB do refluxo e de vapor da base: sem restrição.	92
Figura 34 – Simulações de WB da variação de refluxo e de vapor da base: sem restrição.	93
Figura 35 – Polos de malha fechada de WB.	93
Figura 36 – Simulações de WB da saída do topo e da base: sem restrição e com perturbação.	94
Figura 37 – Simulações de WB do refluxo e de vapor da base: sem restrição e com perturbação.	94
Figura 38 – Simulações de WB da variação de refluxo e de vapor da base: sem restrição e com perturbação.	95
Figura 39 – Simulações de WB da saída do topo e da base: com perturbação e restrição.	95
Figura 40 – Simulações de WB do refluxo e de vapor da base: com perturbação e restrição.	96
Figura 41 – Simulações de WB da variação de refluxo e de vapor da base: com perturbação e restrição.	96

LISTA DE TABELAS

Tabela 1 – Índices de desempenho dos controladores <i>LDMPC</i> e <i>CDMPC</i> para os mesmos parâmetros de sintonia.	62
Tabela 2 – Índices de desempenho: com restrição.	75
Tabela 3 – Índices de desempenho das simulações do Mentor: sem restrição.	80
Tabela 4 – Índices de desempenho das simulações do Mentor: com restrição.	84
Tabela 5 – Índices de desempenho dos testes experimentais do Mentor: com restrição.	87
Tabela 6 – Definição das variáveis e condições típicas de operação para WB (WOOD; BERRY, 1973).	89
Tabela 7 – Índices de desempenho de WB: sem restrição.	91
Tabela 8 – PI Desempenho: sem restrição e com perturbação.	91
Tabela 9 – Índices de desempenho de WB: sem restrição e com perturbação.	91
Tabela 10 – Índices de desempenho de WB: com perturbação e restrição.	97

LISTA DE ABREVIATURAS E SIGLAS

ARMAX	. AutoRegressive Moving Average with eXogenous inputs
ARX AutoRegressive with eXogenous inputs
CDMPC	.. Classic Discrete-time Model Predictive Control
DAQ Data Acquisition
DLQR Discrete-Time Linear Quadratic Regulators
DMC Dynamic Matrix Control
DMPC	... Discrete Model Predictive Control
DOF Degree Of Freedom
FIR Finite Impulse Response
GMV Generalized Minimum Variance
GPC Generalized Predictive Control
HCDMPC	Hildreth Classic Discrete-Time Model Predictive Control
HLDMPC	Hildreth Laguerre Discrete-Time Model Predictive Control
IAE Integral Absolute Error
IDCOM	.. Identification-Command
ISE Integral Of Squared Error
ISU Integral Of Squared Control
LDMPC	.. Laguerre Discrete-Time Model Predictive Control
LQ Linear Quadratic
LQG Linear Quadratic Gaussian
MAC Model Algorithmic Control
MIMO	... Multiple-Input, Multiple-Output
MPC Model Predictive Control
MPHC	... Model Predictive Heuristic Control
PI Proportional Integral
PID Proportional Integral Derivative
RHC Receding Horizon Control
SCDMPC	. Saturated Classic Discrete-Time Model Predictive Control
SISO Single-Input Single-Output
SLDMPC	. Saturated Laguerre Discrete-Time Model Predictive Control
ST Settling Time

TVC Total Variation Of Control

USB Universal Serial Bus

ZOH Zero-Order Hold

LISTA DE SÍMBOLOS

k — Instante de tempo atual

ΔU — Vetor com a predição da trajetória futura do sinal incremental do Controlador *CDMPC*

δ — Delta de Kronecker

$x(\cdot)$ — Variáveis de estado

$u(\cdot)$ — Sinal de controle

$y(\cdot)$ — Sinal de saída

$\xi(\cdot)$ — Ruído nos estados

$d(\cdot)$ — Perturbação na saída do sistema

A — Matriz de estados

B — Matriz de entrada

C — Matriz de Saída

D — Matriz de transição direta

$\Delta u(\cdot)$ — Sinal de controle incremental

$\Delta = 1 - z^{-1}$

$z^{-i} = z^{-i}[f(k)] = f(k - i)$

$x_a(\cdot)$ — Estados Aumentados por Δ

$m1$ — Dimensão do vetor $x_a(\cdot)$

$n1$ — Dimensão do vetor $x(\cdot)$

s — Número de saídas

$I_{s \times s}$ — Matriz identidade

$\bar{0}_{s \times n_1}$ — Matriz nula

$g_{s \times 1}$ — vetor coluna nulo

n — Número de entradas

A_a — Matriz de estados aumentado por Δ

B_a — Matriz de entrada aumentada por Δ

C_a — Matriz de Saída aumentada por Δ
 f, ϕ_x — Par de matrizes usadas na predição de X
 X — Vetor com os estados futuros preditos
 Y — Vetor com as saídas futuras preditas
 F, Φ — Par de matrizes usadas na predição de Y
 J — Função custo
 $y_r(\cdot)$ — Referência do sistema
 R_s — Vetor coluna com as informações de $y_r(\cdot)$
 \bar{I} — uma matriz diagonal com dimensão $nN_c \times nN_c$
 \bar{R} — Matriz diagonal com as informações de Γ
 I_n — Matriz diagonal com dimensões $n \times n$
 o_n — Vetor coluna nulo de dimensão n
 N_p — Horizonte de Predição
 T_s — Período de amostragem
 N_c — Horizonte de Controle
 Γ — Esforço de controle
 K_y — Ganho de malha fechada relacionado com a saída
 K_{mpc} — Ganho de malha fechada usado no MPC
 N — Ordem das funções de Laguerre, número de termos usados em sua expansão
 a — Polo discreto das funções de Laguerre
 $f(\cdot)^*$ — Complexo conjugado de $f(\cdot)$
 $l(\cdot)$ — Elemento das funções de Laguerre discreta
 $L(\cdot)$ — Funções de Laguerre discreta na forma de vetor
 $L_{MIMO}(\cdot)$ — Funções de Laguerre discreta na forma matricial
 σ — Vetor coluna nulo com comprimento N
 N_{sum} — Soma das ordens N das funções de Laguerre
 A_l — Matriz de Toeplitz com dimensão $N \times N$
 ρ — Matriz de zeros com dimensão $N \times N$
 ΔU_L — Vetor com a predição da trajetória futura do sinal incremental do Controlador *LDMPC*
 p — Coeficientes do vetor de η

η — Vetor com as predições de controle do Controlador *LDMPC*

m — Futuras amostras no tempo discreto

Ω, Ψ — Par de matrizes da função custo J

R_L, Q — Par de matrizes peso da função custo J

λ_l — Taxa de decaimento

$e(\cdot)$ — erro ($y(k) - y_r(k)$)

PO — Overshoot

v — Variável de decisão

\bar{J} — Função custo com restrições

M, γ — Matriz e vetor das inequações matriciais ($Mv \leq \gamma$)

λ — Multiplicador de Lagrange ou variável dual

\dot{I} — Matriz diagonal unitária com dimensões $N_c \times N_c$

C_I — Matriz triangular inferior unitária com dimensões $N_c \times N_c$

$\Delta u_{min}, \Delta u_{max}$ — Valores máximos e mínimos para $\Delta u(\cdot)$

u_{min}, u_{max} — Valores máximos e mínimos para $u(\cdot)$

$\Delta U(\cdot)_{restr}$ — Vetor com a predição da trajetória futura do sinal incremental com restrições

η_{restr} — Vetor η com restrições

$\Delta u(\cdot)_{restr}$ — Sinal de controle incremental com restrições

SUMÁRIO

1	INTRODUÇÃO	27
1.1	Introdução Geral	27
1.2	Motivação da pesquisa	31
1.3	Estado da arte	31
1.4	Objetivo do trabalho	33
1.4.1	<i>Objetivos gerais</i>	33
1.4.2	<i>Objetivos específicos</i>	33
1.5	Materiais e Métodos	34
1.6	Estrutura da dissertação	35
2	DMPC: ABORDAGEM CLÁSSICA	37
2.1	Introdução	37
2.2	Modelo no espaço de estados com matriz aumentada	38
2.3	Predição das variáveis da planta	39
2.4	Controle de horizonte retrocedente	40
2.5	Parâmetros de sintonia MPC (N_p, N_c, Γ)	42
2.5.1	<i>Parâmetro N_p</i>	42
2.5.2	<i>Parâmetro N_c</i>	42
2.5.3	<i>Parâmetro Γ</i>	43
2.6	Sistema em malha fechada	43
2.7	Conclusões	44
3	DMPC USANDO FUNÇÕES DE LAGUERRE	49
3.1	Introdução	49
3.2	Funções de Laguerre com base ortonormal	50
3.3	Controle Ótimo usando Funções Laguerre	52
3.3.1	<i>Função Custo</i>	53
3.3.2	<i>Minimização da Função Custo</i>	53
3.3.3	<i>Controle de Horizonte Retrocedente</i>	54
3.3.4	<i>Sistema em Malha Fechada</i>	55
3.4	Caso MIMO	55
3.5	Caso especial para $a = 0$	56
3.6	Parâmetros de Sintonia (R_L, Q, N, a)	57

3.6.1	<i>Parâmetros (R_L, Q)</i>	58
3.6.2	<i>Parâmetros (N, a)</i>	59
3.7	Comparações de Desempenho entre LDMPC e CDMPC para o caso SISO	61
3.8	Conclusões	63
4	DMPC COM RESTRIÇÕES	65
4.1	Introdução	65
4.2	Programação quadrática	66
4.2.1	<i>Fundamentos da programação quadrática</i>	67
4.2.2	<i>Programação quadrática de Hildreth</i>	68
4.3	Controlador CDMPC com Restrições	70
4.4	Controlador LDMPC com Restrições	71
4.5	Simulações com restrições	72
4.6	Conclusões	75
5	TESTES EXPERIMENTAIS	77
5.1	Introdução	77
5.2	Manipulador Robótico Mentor	77
5.2.1	<i>Simulações com o manipulador Mentor</i>	79
5.2.2	<i>Testes experimentais com o manipulador Mentor</i>	85
5.3	Coluna de destilação Binária de Wood e Berry (WB)	88
5.3.1	<i>Simulações de WB</i>	89
5.4	Conclusões	97
6	CONCLUSÃO	99
6.1	Principais contribuições	100
6.2	Publicações	101
6.2.1	<i>Trabalhos relacionados diretamente com a pesquisa</i>	101
6.2.2	<i>Trabalhos relacionados indiretamente com a pesquisa</i>	101
6.3	Proposta de trabalhos futuros	101
	REFERÊNCIAS	103
APÊNDICE A	CIRCUITO DRIVE DE ACIONAMENTO E AQUISIÇÃO DE DADOS	108
APÊNDICE B	CÓDIGO FONTE ARDUINO	109
APÊNDICE C	CÓDIGOS DO DMPC CLÁSSICO	115

APÊNDICE D

**CÓDIGOS DO DMPC USANDO FUNÇÕES DE LA-
GUERRE 124**

INTRODUÇÃO

1.1 Introdução Geral

O controle preditivo, *a priori*, foi uma resposta às necessidades apresentadas nas indústrias químicas, as quais tinham grandes dificuldades ao controle eficiente de processos com múltiplas entradas e múltiplas saídas com restrições. A utilização dessa estratégia de controle foi impulsionada pela crise do petróleo ocorrido na década de 1970 que elevou o preço do petróleo, fazendo com que as indústrias petroquímicas buscassem formas de otimizar ao máximo a produção de combustíveis e de seus derivados (RICHALET; O'DONOVAN, 2009). Este fato criou um espaço para novos tipos de controladores avançados, quebrando a hegemonia dos Controladores PID (Proporcional Integral Derivativo) no meio industrial, o qual apresenta grandes dificuldades para sintonizar e controlar sistemas multivariáveis (ROSICH; OCAMPO-MARTINEZ, 2015).

Segundo Mayne *et al.* (2000), Controle Preditivo Baseado em Modelo (MPC, Model Predictive control), também conhecido como Controle Preditivo de Horizonte Retrocedente (RHC, Recending Horizon Control) é um controlador que gera uma ação de controle, $u(k)$, a cada instante de tempo k , a qual é obtida através da solução de um problema de otimização em malha aberta com horizonte de predição finito. Para isso são usados os estados atuais da planta como se fossem os estados iniciais, assim gerando um vetor com uma sequência de sinais de controle, onde somente o primeiro elemento do vetor é aplicado na planta e o restante sendo descartado. Essa estratégia e a estrutura do controle MPC podem ser vistas na Figura 1 e Figura 2, respetivamente, onde são mostradas as futuras saídas para um determinado horizonte de predição (N_p) e as futuras entradas para um determinado horizonte de controle (N_c).

O MPC tem a capacidade de prever o comportamento futuro do processo através do modelo dinâmico e as medidas disponíveis. Sem dúvida, o aspecto-chave na estrutura do MPC é o fato deste controlador usar um modelo dinâmico (KWONG, 2005). Esse modelo, em sua grande maioria, é baseado em modelos empíricos, expressos na forma de resposta ao degrau ou resposta finita ao impulso (FIR, Finite impulse response). Esses modelos podem ser representados na

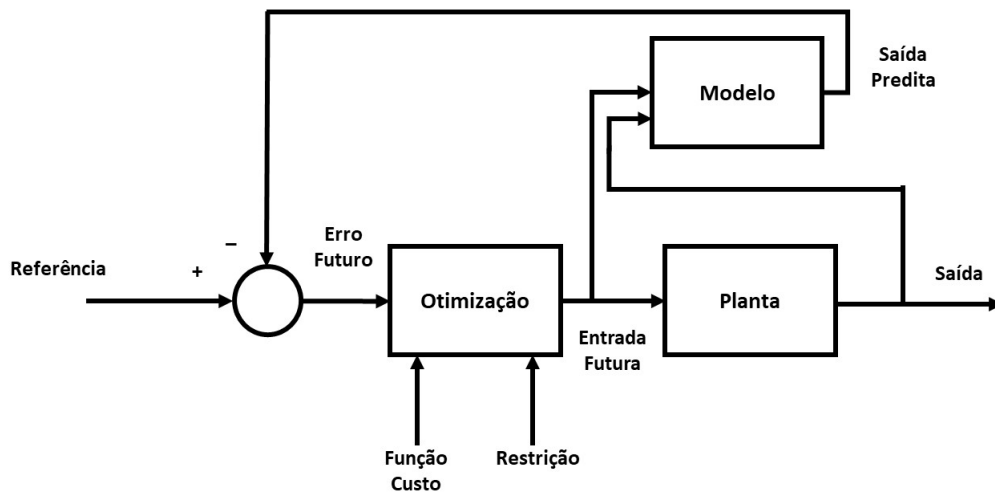


Figura 1 – Estrutura do Controlador MPC.

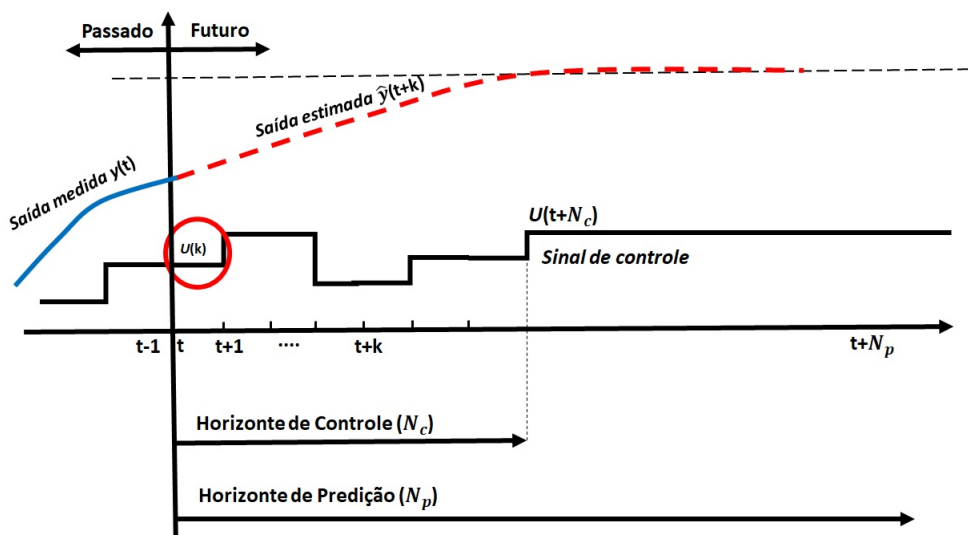


Figura 2 – Estratégia do Controlador MPC.

forma de função de transferência ou em espaço de estados que vem sendo cada vez mais utilizadas em projetos de controle preditivo (SATO *et al.*, 2016; ŁAWRYŃCZUK, 2015; GERKSIC; STRMCNIK, 2001). Isso por causa de sua facilidade em trabalhar com plantas multivariáveis e de ser facilmente generalizado para sistemas mais complexos com perturbação e ruído estocástico (CAMACHO; BORDONS, 2012).

A origem teórica do MPC se deu a partir do desenvolvimento das técnicas de controle ótimo, onde podem-se destacar o controle linear quadrático (LQ, Linear Quadratic) e controle linear quadrático gaussiano (LQG, Linear Quadratic Gaussian) (ANDERSON; MOORE, 1971). Esses trabalhos foram antecedidos pelos trabalhos de Kalman nos anos de 1960 (KALMAN *et al.*, 1960b; KALMAN *et al.*, 1960a), o qual definiu o conceito de controle ótimo na teoria de sistemas lineares e deu as bases para os controladores preditivos desenvolvidos posteriormente.

Porém, somente a partir do final da década de 1970 foram concebidos os primeiros controladores MPC com a atual estrutura que se conhece. Um dos primeiros controladores MPC a ganhar popularidade e notoriedade na indústria e na área acadêmica foi o Controle Preditivo Heurístico Baseado em Modelo (MPHC, Model Predictive Heuristic Control), posteriormente conhecido como Controle Algorítmico Baseado em Modelo (MAC, Model Algorithmic Control) (RICHALET *et al.*, 1978), comercializado como IDCOM (Identification-Command) e como Controle por Matriz Dinâmica (DMC, Dynamic Matrix Control) (CUTLER; RAMAKER, 1980). Esses controladores são considerados a raiz do controle MPC. Posteriormente, Clarke, Mohtadi e Tuffs (1987) desenvolveram o controle preditivo generalizado (GPC, Generalized Predictive Control) usando ideias do controlador de variância mínima generalizado (GMV, Generalized Minimum Variance) (CLARKE; GAWTHROP, 1979).

Nos últimos anos, avançadas técnicas de controle preditivo como em Dovžan e Škrjanc (2010), Causa *et al.* (2008), Potočnik *et al.* (2008), têm sido largamente usadas pelas indústrias, isso por causa do grande desenvolvimento do controle digital que tem se tornado cada vez mais confiável e custo reduzido. Existem também outras razões que justificam porquê o MPC é considerado a técnica de controle avançado mais popular da atualidade para processos industriais. Segundo Camacho e Bordons (2012) os principais motivos são:

- Facilidade em lidar com processos MIMO (Multiple-Input, Multiple-Output).
- Facilidade de aplicação em sistemas com restrições.
- Possibilidade de aplicação tanto para sistemas lineares como para sistemas não lineares.
- Facilidade em trabalhar com processos que apresentam atraso de transporte, instabilidade e fase não mínima.
- A robustez do sistema pode ser melhorada através de uma sintonia adequada dos parâmetros.
- Programação linear ou programação quadrática podem ser aplicadas facilmente para resolver problemas de otimização com restrição.

Os MPC também apresentam desvantagens, segundo Kwong (2005) as principais são:

- Sua estratégia é bastante diferente das estratégias de controle multi-malhas tradicionais, então dificultando sua utilização na indústria por parte dos operadores de planta, devido a seu difícil entendimento e a falta de familiarização com o controlador.
- Necessidade de grande esforço computacional, pois, resolve um problema de controle ótimo a cada instante de tempo.

- A sintonia do sistema torna-se mais trabalhoso devido ao seu grande número de parâmetros de sintonia, consumindo um grande tempo dos operadores das plantas, sendo umas das maiores desvantagens dos controladores preditivos.

De fato, umas das principais propriedades do controle MPC é a facilidade de trabalhar com processos multivariáveis com restrições, onde certas variáveis do sistema devem permanecer dentro de uma faixa de operação pré-estabelecida. Essas variáveis precisam ser facilmente manipuladas pelo controlador para evitar instabilidade ou um desempenho indesejado. A não violação das restrições em processos industriais é de grande importância, pois, violar os limites estipulados no processo ocasiona condições inseguras para os operadores de planta. Além disto, geram a diminuição da qualidade e da quantidade produtiva, causando a diminuição do lucro. Segundo Alvarez e Prada (1997), os principais categorias de restrições são:

- **Restrições físicas:** os limites que nunca podem ser ultrapassados e são determinados pelo funcionamento físico do sistema. Por exemplo, um manipulador robótico não pode ultrapassar os limites de cada uma de suas juntas robóticas, válvulas que não podem ser abertas mais de 100% ou menos de 0%.
- **Restrições de operação:** os limites fixados pelo operador do sistema ou por limites operacionais do sistema. Os quais definem a faixa dentro da qual se espera que as variáveis do sinal de controle $u(k)$, saídas do sistema $y(k)$ e as variáveis de estado $x(k)$ estejam em condições normais de operação. Por exemplo, pode ser necessário operar uma caldeira em certas faixas de temperaturas, de modo a favorecer a cinética da reação desejada que forma determinado produto. Motores ou válvulas que têm uma determinada faixa de tensão de operação para o seu correto funcionamento.

Embora o controle MPC esteja bem estabelecido e amplamente utilizado em alguns setores da indústria, ainda existem algumas questões relacionadas ao conflito entre viabilidade do sinal de controle e o desempenho de malha fechada em sistemas com restrições nas variáveis de entrada, os quais ainda não apresentam soluções satisfatórias sem a necessidade de um alto custo computacional (ROSSITER; WANG, 2008). Sendo assim, o desenvolvimento de técnicas que viabilizem a diminuição desse problema sem exigir um grande custo computacional é de extrema importância para indústria, a qual sempre busca reduzir os custos relacionados a equipamentos.

Tendo em vista que, na prática, todo processo está sujeito a restrições, este trabalho propõe apresentar o projeto de um controlador preditivo multivariável com restrições, desenvolvido por Wang (2004), com objetivo de reduzir o conflito entre viabilidade do sinal de controle e o desempenho do sistema em malha fechada. Além disso, são implementados testes experimentais com o método de controle proposto, aplicado em um manipulador robótico e na coluna de destilação binária de Wood e Berry (1973). O método proposto utiliza as Funções de Laguerre em conjunto com a Programação Quadrática de Hildreth para encontrar a solução ótima para processos com restrições de operação. Ainda, apresenta a vantagem de diminuir a carga com-

putacional utilizada para calcular o controle ótimo. Isto porque utiliza uma rede de filtros de funções ortonormais de Laguerre (detalhadas no Capítulo 3) para obter a trajetória futura do sinal de controle dentro de um horizonte de predição, ao invés dos operadores de avanço de tempo que são utilizados no MPC clássico. Além disso, é realizada a comparação entre o método desenvolvido usando as funções de Laguerre e o MPC clássico, tendo o objetivo de demonstrar às vantagens do controlador proposto.

1.2 Motivação da pesquisa

Na prática, todo processo está sujeito a restrições, principalmente restrições operacionais impostas à trajetória do sinal de controle $u(k)$, como por exemplo, no caso dos motores de corrente contínua com faixa de tensão $-12V \leq \text{Tensão} \leq +12V$, tendo a variável $u(k)$ limitada a essa faixa de operação. No entanto, os controladores MPC podem gerar ações de controle para um desempenho desejado que são muitas das vezes inviáveis para aplicações em processos reais, isso por estarem fora dos limites de operação. Assim, é necessário impor limites superior e inferior a $u(k)$ para torná-lo viável a sua aplicação na planta. Essas restrições, quando impostas aos controladores preditivos, acabam gerando perda de desempenho na saída do processo. Dessa maneira, essas limitações geram um conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada.

Este problema se mostrou de grande importância por ser pouco discutido em sistemas multivariáveis. Portanto, motivando a pesquisa e o desenvolvimento deste trabalho, que busca apresentar formas de reduzir essa divergência em controladores preditivos, sem a necessidade de agregar alta carga computacional ao algoritmo de controle e ainda garantindo a estabilidade e a robustez do sistema.

Além disso, o controle multivariável é algo de extrema relevância no cenário atual, sendo uma contribuição para a Pós-Graduação em Engenharia Elétrica da Universidade Federal do Pará. Isso pelo fato de terem sido encontrados poucos trabalhos que abordam esse tema no programa. Portanto, o controlador apresentado poderá servir de referência para futuros trabalhos que precisem atuar com sistemas com as peculiaridades aqui apresentadas.

1.3 Estado da arte

Nesta seção, é feita uma síntese dos principais trabalhos que abordam soluções para controladores MPC, com restrições de operação, com o objetivo de estudar as principais técnicas aplicadas e desenvolvidas pela comunidade científica.

Em Rossiter, Kouvaritakis e Gossner (1994) é proposto um algoritmo para reduzir a inviabilidade do sinal de controle para um horizonte finito e assim garantindo a estabilidade e rastreamento assintótico no sistema. Esta técnica foi desenvolvida e aplicada em controladores GPC com objetivo de melhorar o desempenho em processos com restrição. Ela se baseia na

modificação dos MWLS (Mixed Weights Least Squares) adequadamente, o que garante a recuperação da viabilidade, estabilidade global e o rastreamento assintótico de referência.

Alvarez e Prada (1997) demonstram um método para resolver problemas relacionados às restrições em um controlador MIMO GPC. Nele, a viabilidade da ação de controle é recuperada aplicando diferentes técnicas ou uma combinação de todas elas para reduzir os efeitos das restrições. Este método baseia-se na solução de um problema de Programação Linear Quadrática Estendida, na qual as variáveis são minimizadas de acordo com as mudanças nas entradas do sistema e nos limites das restrições selecionadas.

Fukushima e Bitmead (2003) desenvolveram uma programação quadrática para MPC robusto, com incertezas de modelo e perturbações na entrada. Neste trabalho é construído um MPC baseado na predição em malha fechada que é capaz de trabalhar com restrições e incertezas sem perder a robustez.

Wang (2004) elabora um controlador MPC discreto que usa as Funções de Laguerre para encontrar a solução ótima para processos com restrições na ação de controle. Posteriormente Rossiter e Wang (2008) exploram os benefícios das Funções de Laguerre para melhorar o compromisso entre a viabilidade do sinal de controle e o desempenho da saída do sistema. Estas funções ortonormais aplicadas em controladores preditivos são introduzidas e analisadas com maiores detalhes e profundidade no Capítulo 3 deste trabalho e em Wang (2009). Estes três trabalhos se tornaram a base para o desenvolvimento desta pesquisa, tendo sido escolhida como a técnica estudada e apresentada nesta dissertação por causa da ideia central que modela o futuro sinal de controle dentro de uma janela de horizonte móvel usando uma estrutura de rede de filtros de funções ortonormais, além da sua simplicidade para trabalhar com sistemas MIMO. Isso pelo fato de usar o modelo no espaço de estados que facilita todo o processo de controle para n entradas e s saídas.

Foram encontradas várias aplicações do controle MPC usando Funções de Laguerre em processos, podendo-se destacar:

- Valencia-Palomo e Rossiter (2010b) utilizam as Funções de Laguerre em conjunto com uma programação quadrática multi-paramétrica para reduzir o esforço computacional para problemas de otimização, o qual tem sua implementação em Valencia-Palomo e Rossiter (2010a) onde foi realizada em um Controlador Lógico Programável (PLC, Programmable Logic Controller) para demonstrar seu potencial no meio industrial.
- Wang *et al.* (2011) aplicam um Controle Preditivo Repetitivo multivariável usando Funções de Laguerre em um manipulador robótico configurado como um sistema MIMO com duas entradas e duas saídas.
- Yoo e Wang (2012) demonstram o controle MPC com restrição usando Funções de Laguerre em inversores de frequências, onde é usada a programação quadrática de Hildreth em conjunto com as Funções de Laguerre para solução ótima de problemas com restrição.

- Solaman e Francis (2015) empregam as Funções de Laguerre em colunas de destilação binária para melhorar o desempenho e simplificar a solução ótima do problema.
- Pham e Meesomboon (2016) abordam as Funções de Laguerre no controle de um dirigível com objetivo de melhorar o controle de altitude.

Os últimos trabalhos, encontrados até 2016, que tiveram contribuições com pesquisa envolvendo Funções de Laguerre em controles preditivos com restrição foram Abdullah e Rossiter (2016), que propõem um Controle Preditivo Funcional (PFC, Predictive Functional Control), o qual explora os potenciais benefícios das Funções de Laguerre no PFC. Em seguida, Ettefagh *et al.* (2016) aplicam um controlador MPC usando Funções de Laguerre em sistemas variantes no tempo.

1.4 Objetivo do trabalho

1.4.1 Objetivos gerais

O objetivo geral deste trabalho é apresentar o projeto de um controlador preditivo multivariável baseado em modelo com restrições usando as Funções de Laguerre. Assim, demonstrando suas vantagens em relação ao método MPC clássico e os benefícios das Funções de Laguerre para reduzir o problema entre a viabilidade do sinal de controle e desempenho de malha fechada em controladores preditivos com restrições, bem como diminuir a carga computacional utilizada pelas rotinas de otimização em tempo real normalmente usadas pelos algoritmos do tipo MPC.

1.4.2 Objetivos específicos

Para alcançar o objetivo principal, alguns objetivos específicos foram traçados:

- Aprofundar os conhecimentos sobre as Funções de Laguerre para entender o seu funcionamento e seus benefícios quando aplicadas a controladores MPC.
- Desenvolver, no *MATLAB*[®], o algoritmo do MPC clássico e do MPC usando as Funções de Laguerre, propostos por Wang (2004).
- Desenvolver, no *MATLAB*[®], o algoritmo da Programação Quadrática de Hildreth para os casos com restrição de operação no sinal de controle.
- Apresentar resultados que comprovem que as Funções de Laguerre diminuem a carga computacional utilizada para calcular o sinal de controle, e que também reduzem o conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada ocorrida pelas restrições impostas ao sistema.

- Com os algoritmos desenvolvidos, realizar testes experimentais no manipulador robótico e também simulações com a coluna de Destilação Binária de Wood e Berry (1973), além de comparar os resultados obtidos através de índices de desempenho e assim podendo verificar a eficiência do método proposto em relação ao método clássico.

1.5 Materiais e Métodos

Para o desenvolvimento deste trabalho foram utilizados os materiais listados a seguir:

- Um notebook com as configurações: 8GB 1600MHz DDR3, 6MB L3cache, 2 GHz Intel Core i7-3632QM.
- Um robô manipulador Mentor Desktop robot da Feedback Instruments Limited com 5 graus de liberdade, apresentado em maiores detalhes no Capítulo 5.
- Um circuito drive de acionamento e aquisição de dados, composto por um Arduino Micro, transístores, resistores, capacitores e pontes H, desenvolvido com o objetivo de realizar a comunicação analógico digital e digital analógico (AD/DA) entre o notebook e o robô manipulador. Eles também efetuam os acionamentos dos motores das juntas robóticas para a posição desejada. Para mais informações, é dado no Apêndice A o esquema do circuito e seus componentes.

A metodologia empregada é constituída pelos passos a seguir:

- No *MATLAB*[®] foram efetuadas simulações com os modelos matemáticos que representam a dinâmica do robô manipulador e a coluna de destilação binária, tendo o objetivo de testar os algoritmos desenvolvidos.
- Foram utilizadas as funções DaqDuino da biblioteca DAQ (Data Acquisition) para Matlab/Simulink, disponível para download na página da Mathworks¹. Elas foram usadas para realizar o envio e aquisição de dados via *MATLAB*[®] de processos MIMO.
- Foi desenvolvido um programa para o Arduino, chamado de *Arduino MIMO*, disponível no Apêndice B e que funciona da seguinte maneira: a cada vez que os dados, que representam os sinais de controle, são enviados pelo computador via comunicação serial USB (Universal Serial Bus) para o microcontrolador, eles são lidos e direcionados para seus respectivos canais PWM (Pulse Width Modulation), que representam as entradas do processo, logo após ocorre a leitura de cada conversor Analógico/Digital (A/D), que são as saídas do processo, esses dados são enviados via comunicação serial USB para o computador, assim finalizando o ciclo da função e aguardando receber os próximos dados enviados pelo computador.

¹ <http://www.mathworks.com/matlabcentral/fileexchange/50784-daquino>

- Os testes experimentais com o manipulador Mentor foram desenvolvidos da seguinte maneira: no *MATLAB*[®], a cada instante de tempo discreto k , os sinais de controle dos controladores preditivos são calculados e enviados utilizando as funções DaqDuino via serial para o microcontrolador, onde são lidos e direcionados para seus respectivos canais PWM, que através de uma ponte H acionam os motores das juntas robóticas. Em seguida, as tensões dos potenciômetros que realizam a realimentação da malha de controle são lidas através dos conversores A/D, e por fim, são enviadas via comunicação serial para o *MATLAB*[®]. Sendo assim, a cada instante de tempo discreto, todos esses passos descritos são realizados no experimento. Sendo o período de amostragem utilizado de 0.1 segundos.

1.6 Estrutura da dissertação

Além do capítulo introdutório, este trabalho está organizado da seguinte forma:

Capítulo 2: é apresentado o controlador DMPC (Discrete Model Predictive Control) multivariável no espaço de estados em sua abordagem clássica, mostrando a formulação de sua estrutura, a lei de controle de horizonte retrocedente e outras características do controlador.

Capítulo 3: demonstra-se o projeto do controlador DMPC multivariável no espaço de estados usando Funções de Laguerre. Além de detalhar a forma de funcionamento das Funções de Laguerre no controle preditivo, os parâmetros envolvidos e seus benefícios em relação à abordagem clássica são discutidos.

Capítulo 4: aborda-se o problema de controle com restrição em controladores preditivos multivariáveis, apresentando a Programação Quadrática de Hildreth para encontrar a solução ótima de problemas com restrições de operação em controladores DMPC.

Capítulo 5: são realizadas simulações com a coluna de destilação binária e testes experimentais com o manipulador robótico Mentor. Os resultados obtidos com os controladores apresentados são confrontados para demonstrar as melhorias do algoritmo que utiliza as Funções de Laguerre.

Capítulo 6: apresentam-se as conclusões finais, onde são discutidos os resultados, as contribuições e os trabalhos futuros que podem ser realizados a partir desse tema.

DMPC: ABORDAGEM CLÁSSICA

2.1 Introdução

A essência do MPC é sem dúvida a otimização da trajetória futura do sinal de controle para um horizonte de predição. Em sua abordagem clássica, essa trajetória futura é representada por um vetor ΔU , que para o caso SISO (Single-input Single-Output) é definido por,

$$\Delta U = \left[\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \cdots \quad \Delta u(k_i + N_c - 1) \right]^T \quad (2.1)$$

em que cada elemento de ΔU pode ser representado por δ para um instante de tempo discreto k ,

$$\Delta u(k_i + i) = \left[\delta(i) \quad \delta(i + 1) \quad \cdots \quad \delta(i + N_c - 1) \right] \Delta U \quad (2.2)$$

onde,

$$\delta(i) = \begin{cases} 0, & \text{se } i \neq 0 \\ 1, & \text{se } i = 0 \end{cases} \quad (2.3)$$

é o delta de Kronecker (EDWARDS, 1987), também conhecido como pulso de avanço discreto. Assim, no MPC clássico, a trajetória de controle é obtida através do pulso de avanço discreto, diferente do MPC usando Funções de Laguerre, que será explicado em detalhes no capítulo 3, o qual utiliza funções ortonormais para obter a trajetória futura do sinal de controle.

Neste capítulo será apresentado o controle preditivo discreto usando o modelo no espaço de estados em sua abordagem clássica para o caso *sem restrição*, denominado neste trabalho de Controle Preditivo Discreto Clássico Baseado em Modelo (CDMPC, Classic Discrete-time Model Predictive Control), onde serão demonstrados para os casos MIMO e SISO, o modelo aumentado no espaço de estados, a lei de controle de horizonte retrocedente, os parâmetros de sintonia e os ganhos de realimentação de estados em sua configuração em malha fechada.

2.2 Modelo no espaço de estados com matriz aumentada

O modelo no espaço de estados com n entradas e s saídas pode ser descrito por:

$$x(k+1) = Ax(k) + Bu(k) + \xi(k) \quad (2.4)$$

$$y(k) = Cx(k) + Du(k) + d(k) \quad (2.5)$$

Onde, $x(k)$ é o vetor de estados, $u(k)$ é a variável de entrada (Sinal de controle), $y(k)$ é a saída do processo e k é a variável para cada instante de amostragem. As matrizes (A, B, C, D) são respectivamente as matrizes de estados, de entrada, de saída e de transição direta. Sendo também $\xi(k)$ um vetor de ruídos nos estados e $d(k)$ um vetor de perturbações nas saídas do sistema. Tendo em vista que $u(k)$ não pode afetar a saída $y(k)$ no mesmo instante de tempo, é considerado $D = 0$.

Então, incluindo Δ em ambos os lados da Equação 2.4 e 2.5, para eliminar o erro de regime estacionário na saída e nos estados, têm-se os estados aumentados da planta, representados por:

$$\Delta x(k+1) = A\Delta x(k) + B\Delta u(k) \quad (2.6)$$

$\Delta x(k) = x(k) - x(k-1)$, $\Delta u(k) = u(k) - u(k-1)$. Semelhante a 2.6, obtém-se também a saída do sistema, dada como:

$$\begin{aligned} y(k+1) - y(k) &= C(x(k+1) - x(k)) \\ &= C\Delta x(k+1) \\ y(k+1) &= CA\Delta x(k) + CB\Delta u(k) + y(k) \end{aligned} \quad (2.7)$$

onde $\Delta = 1 - z^{-1}$, $z^{-i}[f(k)] := f(k-i)$.

Logo, o sistema aumentado por Δ é dado por:

$$\begin{bmatrix} \overbrace{\Delta x(k+1)}^{x_a(k+1)} \\ y(k+1) \end{bmatrix} = \begin{bmatrix} \overbrace{A}^{A_a} & \overbrace{\bar{0}_{s \times n_1}^T} \\ \overbrace{CA} & \overbrace{I_{s \times s}} \end{bmatrix} \begin{bmatrix} \overbrace{\Delta x(k)}^{x_a(k)} \\ y(k) \end{bmatrix} + \begin{bmatrix} \overbrace{B}^{B_a} \\ \overbrace{CB} \end{bmatrix} \Delta u(k) \quad (2.8)$$

$$y(k) = \begin{bmatrix} \overbrace{\bar{0}_{s \times n_1} & I_{s \times s}}^{C_a} \end{bmatrix} \begin{bmatrix} \Delta x(k) \\ y(k) \end{bmatrix}$$

onde tem-se o novo vetor de estados aumentados $x_a(k) = [\Delta x(k)^T \ y(k)^T]^T$ com dimensão de $m_1 \times 1$, $m_1 = (n_1 + s)$, sendo n_1 a dimensão do vetor $x(k)$ e s o número de saídas. Já $I_{s \times s}$ é uma matriz identidade, $\bar{0}_{s \times n_1}$ é uma matriz de zeros e $g_{s \times 1}$ é um vetor coluna composto de zeros, enquanto as matrizes aumentadas A_a, B_a, C_a têm as dimensões, $m_1 \times m_1, m_1 \times n$ e $s \times m_1$, respectivamente, onde n é o número de entradas.

2.3 Predição das variáveis da planta

Para um melhor entendimento, será explicada a predição das variáveis da planta levando em consideração um sistema SISO, e posteriormente estendido para o caso MIMO.

Tendo como base o modelo aumentado no espaço de estados em 2.8 e assumindo que $\xi(k) = 0$ e $d(k) = 0$, os estados futuros da planta são calculados sequencialmente por:

$$\begin{aligned}
 x_a(k_i + 1|k_i) &= A_a x_a(k_i) + B_a \Delta u(k_i) \\
 x_a(k_i + 2|k_i) &= A_a^2 x_a(k_i) + A_a B_a \Delta u(k_i) + B_a \Delta u(k_i + 1) \\
 &\vdots \\
 x_a(k_i + N_p|k_i) &= A_a^{N_p} x_a(k_i) + A_a^{N_p-1} B_a \Delta u(k_i) + \dots + A_a^{N_p-N_c} B_a \Delta u(k_i + N_c - 1)
 \end{aligned} \tag{2.9}$$

de onde obtém-se sua representação matricial (ROSSITER, 2003):

$$\begin{array}{c} \overbrace{\begin{bmatrix} x_a(k_i + 1|k_i) \\ x_a(k_i + 2|k_i) \\ x_a(k_i + 3|k_i) \\ \vdots \\ x_a(k_i + N_p|k_i) \end{bmatrix}}^X \end{array} = \begin{array}{c} \overbrace{\begin{bmatrix} A_a \\ A_a^2 \\ A_a^3 \\ \vdots \\ A_a^{N_p} \end{bmatrix}}^f \end{array} x_a(k_i) + \begin{array}{c} \overbrace{\begin{bmatrix} B_a & 0 & 0 & \dots \\ A_a B_a & B_a & 0 & \dots \\ A_a^2 B_a & A_a B_a & B_a & \dots \\ \vdots & \vdots & \vdots & \vdots \\ A_a^{N_p-1} B_a & A_a^{N_p-2} B_a & A_a^{N_p-3} B_a & \dots \end{bmatrix}}^{\phi_x} \end{array} \begin{array}{c} \overbrace{\begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \\ \vdots \\ \Delta u(k_i + N_c - 1) \end{bmatrix}}^{\Delta U} \end{array} \tag{2.10}$$

As saídas futuras são baseadas no modelo e nos estados atuais do sistema. Semelhante ao cálculo dos estados futuros, as predições das saídas futuras são dadas por:

$$\begin{aligned}
 y(k_i + 1|k_i) &= C_a A_a x_a(k_i) + C_a B_a \Delta u(k_i) \\
 y(k_i + 2|k_i) &= C_a A_a^2 x_a(k_i) + C_a A_a B_a \Delta u(k_i) + C_a B_a \Delta u(k_i + 1) \\
 &\vdots \\
 y(k_i + N_p|k_i) &= C_a A_a^{N_p} x_a(k_i) + C_p A_p^{N_p-1} B_p \Delta u(k_i) + \dots + C_p A_p^{N_p-N_c} B_p \Delta u(k_i + N_c - 1)
 \end{aligned} \tag{2.11}$$

cuja a forma matricial, recai em,

$$\begin{array}{c} \overbrace{\begin{bmatrix} y(k_i + 1|k_i) \\ y(k_i + 2|k_i) \\ y(k_i + 3|k_i) \\ \vdots \\ y(k_i + N_p|k_i) \end{bmatrix}}^Y \end{array} = \begin{array}{c} \overbrace{\begin{bmatrix} C_a A_a \\ C_a A_a^2 \\ C_a A_a^3 \\ \vdots \\ C_a A_a^{N_p} \end{bmatrix}}^F \end{array} x_a(k_i) + \begin{array}{c} \overbrace{\begin{bmatrix} C_a B_a & 0 & \dots \\ C_a A_a B_a & C_a B_a & \dots \\ C_a A_a^2 B_a & C_a A_a B_a & \dots \\ \vdots & \vdots & \vdots \\ C_a A_a^{N_p-1} B_a & \dots & C_a A_a^{N_p-N_c} B_a \end{bmatrix}}^{\Phi} \end{array} \begin{array}{c} \overbrace{\begin{bmatrix} \Delta u(k_i) \\ \Delta u(k_i + 1) \\ \Delta u(k_i + 2) \\ \vdots \\ \Delta u(k_i + N_c - 1) \end{bmatrix}}^{\Delta U} \end{array} \tag{2.12}$$

As Equações 2.10 e 2.12 podem ser reescritas na forma compacta em 2.13 e 2.14:

$$X = f x_a(k_i) + \phi_x \Delta U \tag{2.13}$$

$$Y = F x_a(k_i) + \Phi \Delta U \tag{2.14}$$

nas quais,

$$F = \begin{bmatrix} C_a A_a \\ C_a A_a^2 \\ C_a A_a^3 \\ \vdots \\ C_a A_a^{N_p} \end{bmatrix} \quad (2.15)$$

$$\Phi = \begin{bmatrix} C_a B_a & 0 & 0 & \cdots & 0 \\ C_a A_a B_a & C_a B_a & 0 & \cdots & 0 \\ C_a A_a^2 B_a & C_a A_a B_a & C_a B_a & \cdots & 0 \\ \vdots & & & & \\ C_a A_a^{N_p-1} B_a & C_a A_a^{N_p-2} B_a & C_a A_a^{N_p-3} B_a & \cdots & C_a A_a^{N_p-N_c} B_a \end{bmatrix} \quad (2.16)$$

onde, Φ é uma matriz triangular inferior com elementos não nulos com dimensão $sN_p \times nN_c$, F tem dimensão $sN_p \times m_1$. Y e ΔU têm as dimensões $sN_p \times 1$ e $nN_c \times 1$, respectivamente.

Tem-se N_p , chamado de horizonte de predição, o qual determina o número de termos usados para prever os estados e as saídas futuras da planta e N_c , horizonte de controle, o número de termos usados para capturar a futura trajetória de controle, o qual pode ser menor ou igual N_p .

Nota-se que todas as variáveis futuras do modelo preditivo no espaço de estados são preditas baseadas na informação atual medida de $x_a(k_i)$ e nos movimentos futuros de $\Delta u(k_i + j)$, sendo isso uma característica importante dos controladores preditivos de horizonte retrocedente.

2.4 Controle de horizonte retrocedente

O principal objetivo da função custo é estabelecer o problema de otimização, que no caso dos MPC se baseia em minimizar o erro entre a saída predita e o sinal de referência (*set-point*), assim sendo possível encontrar o melhor ΔU para um horizonte de predição. A forma generalizada da função custo pode ser descrita por:

$$J = (R_s - Y)^T (R_s - Y) + \Delta U^T \bar{R} \Delta U \quad (2.17)$$

onde R_s é um vetor com dimensões $sN_p \times 1$ que contém as informações de y_{r_n} (sinal de referência), definido por,

$$R_s = \begin{bmatrix} (y_{r_1}(k_i) \ y_{r_2}(k_i) \ \cdots \ y_{r_n}(k_i))^T \\ (y_{r_1}(k_i+1) \ y_{r_2}(k_i+1) \ \cdots \ y_{r_n}(k_i+1))^T \\ \vdots \\ (y_{r_1}(k_i+N_p-1) \ y_{r_2}(k_i+N_p-1) \ \cdots \ y_{r_n}(k_i+N_p-1))^T \end{bmatrix} \quad (2.18)$$

$\bar{R} = \Gamma \bar{I}$, sendo \bar{I} uma matriz diagonal com dimensão $nN_c \times nN_c$ e Γ ($\Gamma \geq 0$) um escalar de ponderação do esforço de controle desejado em malha fechada, ou seja, quanto menor o valor de Γ maior a energia empregada pelo sinal de controle.

O primeiro termo de J , definido por $(R_s - Y)^T (R_s - Y)$ está relacionado com a minimização do erro entre a saída predita e o sinal de referência. Já o segundo termo está relacionado com a trajetória de controle predita, sendo que esses dois termos devem tender a zero para uma otimização. Tradicionalmente, essa função custo é usada com o objetivo de manter a estabilidade de malha fechada e rejeitar perturbações no sistema (WANG, 2009).

Para encontrar a solução ótima para ΔU que minimize J , é necessário substituir Y da Equação 2.14 em 2.17. Então J pode ser expresso por:

$$J = (R_s - Fx_a(k))^T (R_s - Fx_a(k)) - 2\Delta U^T \Phi^T (R_s - Fx_a(k)) + \Delta U^T (\Phi^T \Phi + \bar{R}) \Delta U \quad (2.19)$$

Aplicando a primeira derivada a 2.19, tem-se.

$$\frac{\partial J}{\partial \Delta U} = -2\Phi^T (R_s - Fx_a(k)) + 2(\Phi^T \Phi + \bar{R}) \Delta U \quad (2.20)$$

Sabendo-se que a condição de minimização de J é obtida quando $\frac{\partial J}{\partial \Delta U} = 0$, então, se não houver restrição no sinal de controle, obtém-se como resultado a lei de controle que pode ser expressa como

$$\Delta U = (\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx_a(k)) \quad (2.21)$$

Já para o caso MIMO o vetor ΔU é dado por,

$$\Delta U = \begin{bmatrix} (\Delta u_1(k_i) \Delta u_2(k_i) \cdots \Delta u_n(k_i))^T \\ (\Delta u_1(k_i + 1) \Delta u_2(k_i + 1) \cdots \Delta u_n(k_i + 1))^T \\ \vdots \\ (\Delta u_1(k_i + N_c - 1) \Delta u_2(k_i + N_c - 1) \cdots \Delta u_n(k_i + N_c - 1))^T \end{bmatrix} \quad (2.22)$$

Embora o vetor ΔU tenha toda a trajetória futura predita para um determinado N_c , somente o seu primeiro termo $(\Delta u_1(k_i) \Delta u_2(k_i) \cdots \Delta u_n(k_i))^T$ é utilizado para controlar a planta, o restante dos termos são descartados. Isso porque distúrbios e erros de modelagem podem levar a diferença entre o comportamento estimado e real da planta. Este procedimento se repete a cada instante de tempo k , sendo por isso denominado controle de horizonte retrocedente.

Tendo em vista isso, para obter somente os primeiros valores do vetor ΔU para utilizar no instante atual e também com o objetivo de facilitar os cálculos e as análises, acrescentou-se na Equação 2.21, a Equação,

$$v = \overbrace{\begin{bmatrix} I_n & o_n & \cdots & o_n \end{bmatrix}}^{nN_c} \quad (2.23)$$

onde I_n é uma matriz diagonal com dimensões $n \times n$ e o_n é um vetor coluna nulo de dimensão n . Assim a lei de controle pode ser reescrita por,

$$\Delta u(k) = v(\Phi^T \Phi + \bar{R})^{-1} \Phi^T (R_s - Fx_a(k)) \quad (2.24)$$

2.5 Parâmetros de sintonia MPC (N_p , N_c , Γ)

Neste tópico serão abordados os parâmetros de sintonia do controlador *CDMPC*, mostrando o objetivo de cada um deles e suas características e influências em relação ao desempenho do sistema.

2.5.1 Parâmetro N_p

O parâmetro N_p é o intervalo de tempo futuro que o controlador *CDMPC* deve avaliar por previsão para otimizar ΔU em cada instante de tempo k . No início do projeto, recomenda-se escolher N_p e mantê-lo constante, sintonizando-se apenas os outros parâmetros (N_c , Γ) do *MPC*. O valor de N_p deve ser tal que o controlador seja internamente estável e antecipe violações de restrições com antecedência suficiente para permitir ações corretivas.

Para calcular um valor inicial aproximado para N_p , pode-se considerar a relação entre o tempo de acomodação (ST) desejado em malha fechada e o tempo de amostragem (T_s), assim, pode-se fazer a seguinte relação $N_p \approx \frac{ST}{T_s}$. Em seguida, é recomendado aumentar gradualmente N_p até que não haja mudanças perceptíveis no desempenho do sistema (GARRIGA; SOROUGH, 2010; BEMPORAD; MORARI; RICKER, 2010).

Já para Camacho e Alba (2013), é recomendável usar N_p compatível com o tempo de acomodação do sistema em malha aberta, se ele for estável em malha aberta, logicamente. A ideia é que todo o comportamento transitório do sistema seja conhecido pelo preditor.

Para um melhor entendimento sobre o parâmetro N_p , foram realizadas simulações para vários valores de N_p . Na Figura 3, pode-se ver a resposta da variação de $N_p = 6; 7; 8; 48; 49; 50$ (com $N_c = 5$ e $\Gamma = 0,01$) para a função de transferência que representa a dinâmica de uma junta robótica do manipulador Mentor (ver Capítulo 5), definida por $G_{11}(z) = \frac{-0,00072717z - 0,0039167}{z^2 - 1,6042z + 0,60415}$ com $T_s = 0,1$. A qual apresenta uma dinâmica de planta integradora, com um polo na origem.

Na Figura 3, pode-se perceber que para $N_p = 48; 49; 50$, não ocorrem melhoras significativas no desempenho do sistema, ou seja, valores $N_p > 48$ acabam sendo desnecessários, isso porque valores maiores somente iriam aumentar a carga computacional do controlador.

2.5.2 Parâmetro N_c

O parâmetro $N_c = \{N_c \in \mathbb{N} \text{ e } 1 \leq N_c \leq N_p\}$ é o número de movimentos de ΔU em um instante de tempo k . Quanto maior for este, maior será a carga computacional do controlador. Na Figura 4 são mostradas simulações para mesma função de transferência utilizada nas simulações para o caso N_p , porém, ocorre a variação de $N_c = 2; 3; 4; 5; 6; 48$ (com $N_p = 48$ e $\Gamma = 0,01$).

Apesar de intuitivamente se pensar que quanto maior N_c melhor será o desempenho do sistema, na prática, não é isso que ocorre conforme mostrado na Figura 4. Para $N_c = N_p$ houve redução de desempenho. Sendo assim, valores menores para N_c podem ter resultados de maior utilidade prática.

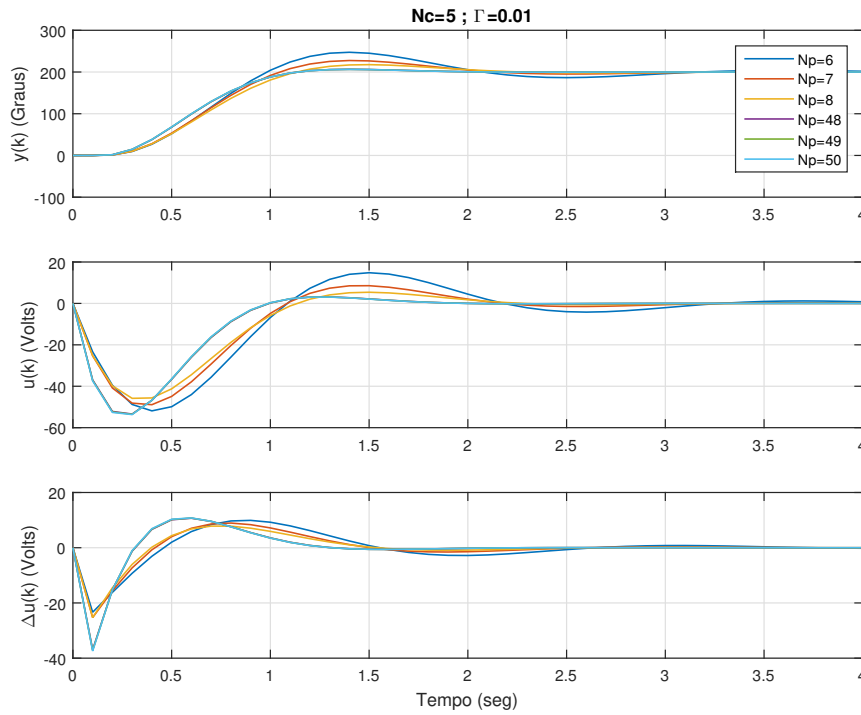


Figura 3 – Resposta de $G_{11}(z)$ para a variação de N_p .

2.5.3 Parâmetro Γ

O parâmetro Γ é responsável pelo esforço de controle e está relacionado com a energia liberada pelo sinal de controle. Quanto menor seu valor, maior será a energia liberada, sendo $\Gamma \geq 0$. Assim como as simulações realizadas anteriormente, é realizada a variação de $\Gamma = 0,01; 0,02; 0,03; 0,04; 0,05; 0,06$ (com $N_p = 48$ e $N_c = 3$). Na Figura 5 podem ser observados os resultados obtidos. Assim, pode-se ver quanto menor o valor de Γ maior a energia do sinal de controle, $\Delta u(k)$.

2.6 Sistema em malha fechada

A análise em malha fechada é muito importante para verificar a estabilidade de um sistema. No controle preditivo, os ganhos em malha fechada são obtidos através da Equação 2.24, que pode ser reescrita na forma a seguir:

$$\Delta u(k) = K_y R_s - K_{mpc} x_a(k) \quad (2.25)$$

sendo K_y com dimensões $s \times s$, dado por

$$K_y = v((\Phi^T \Phi + \bar{R})^{-1} \Phi^T) \quad (2.26)$$

e K_{mpc} com dimensões $n \times m_1$, obtido mediante,

$$K_{mpc} = v((\Phi^T \Phi + \bar{R})^{-1} \Phi^T F) \quad (2.27)$$

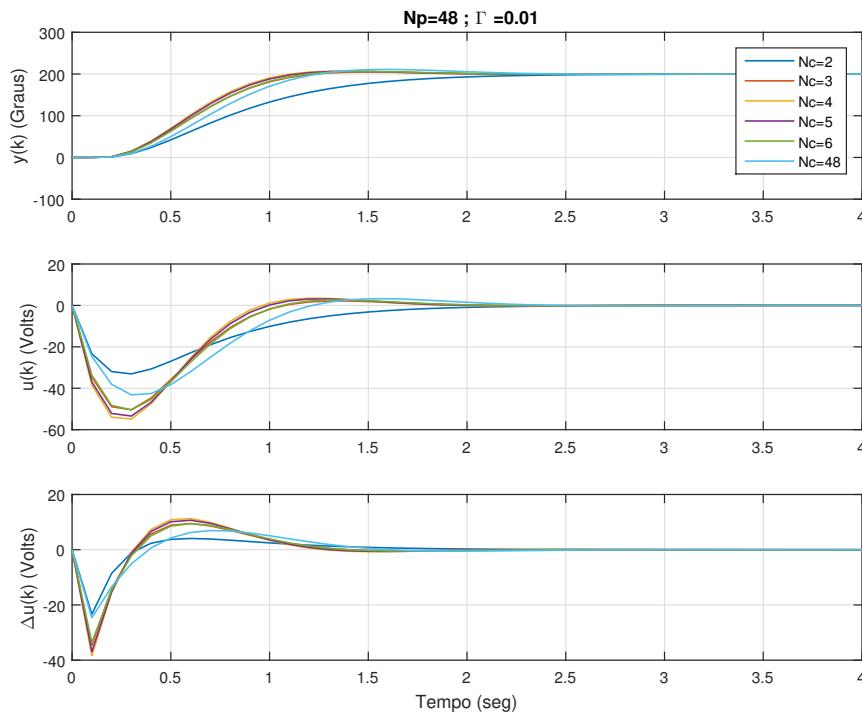


Figura 4 – Resposta de $G_{11}(z)$ para a variação de N_c .

Por meio do modelo dos estados aumentados por Δ é possível obter a Equação de malha fechada, como é demonstrado em,

$$\begin{aligned}
 x_a(k+1) &= A_a x_a(k) + B_a \Delta u(k) \\
 &= A_a x_a(k) - B_a K_{mpc} x_a(k) + B_a K_y R_s \\
 &= (A_a - B_a K_{mpc}) x_a(k) + B_a K_y R_s
 \end{aligned} \tag{2.28}$$

A fim de melhorar o entendimento desses controladores, nas Figuras 6 e 7, são apresentados o diagrama de blocos da estrutura no espaço de estados e o fluxograma do projeto do controlador *CDMPC*, respectivamente.

2.7 Conclusões

Neste capítulo foi discutido o controle preditivo baseado em modelo no espaço de estados, considerando sua abordagem clássica no domínio do tempo discreto. Assim, este capítulo teve como principal objetivo apresentar os conceitos teóricos, suas formulações e seus parâmetros de sintonia.

No projeto do controlador *CDMPC*, foi introduzido o conceito de modelo aumentado por Δ , sendo incorporado naturalmente no projeto, ocasionando o seguimento de referência e também rejeitando os distúrbios na entrada e na saída, sem erros de estado estacionário para referências do tipo degrau.

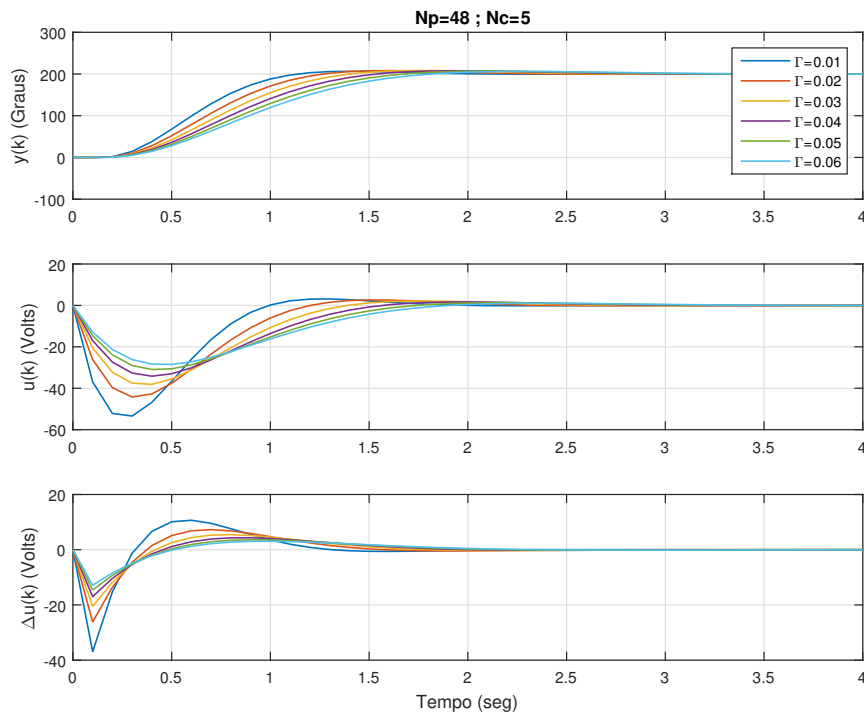


Figura 5 – Resposta de $G_{11}(z)$ para a variação de Γ .

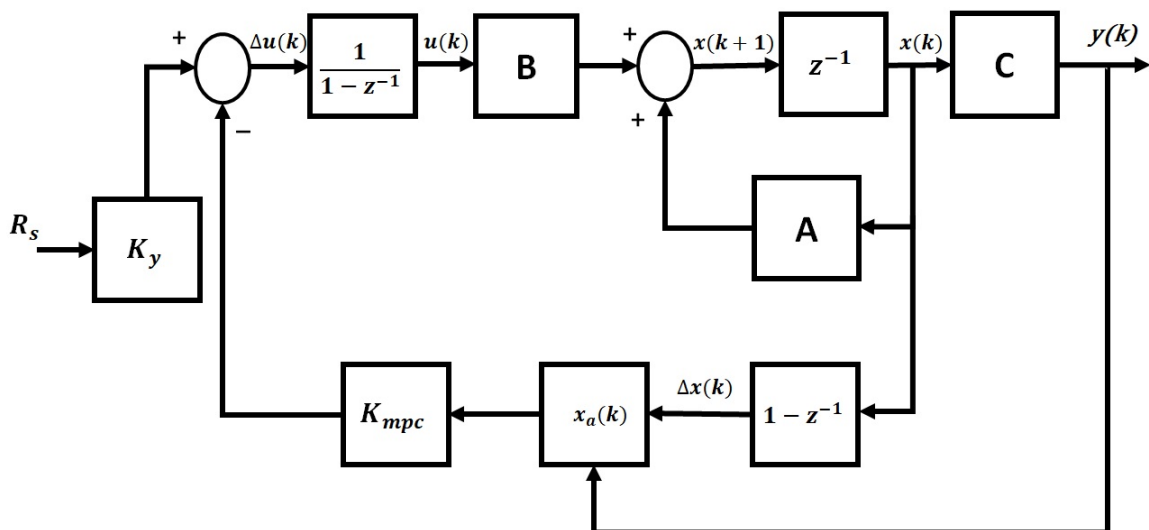


Figura 6 – Controlador $CDMPC$ no espaço de estados.

Foi mostrado que a predição do comportamento futuro do sistema depende do modelo no espaço de estados, que possuindo a atual medida dos estados $x_a(k)$, são obtidas a predição do comportamento futuro dos estados e as saídas da planta.

Foi também apresentado o projeto da lei de controle sem restrições, baseada no princípio do horizonte retrocedente, onde a trajetória futura do sinal de controle incremental é definida por um vetor de comprimento N_c , $\Delta u(k_i) \quad \Delta u(k_i + 1) \quad \dots \quad \Delta u(k_i + N_c - 1)$, sendo obtido através da minimização de uma função custo J para um horizonte de controle N_c e um horizonte de

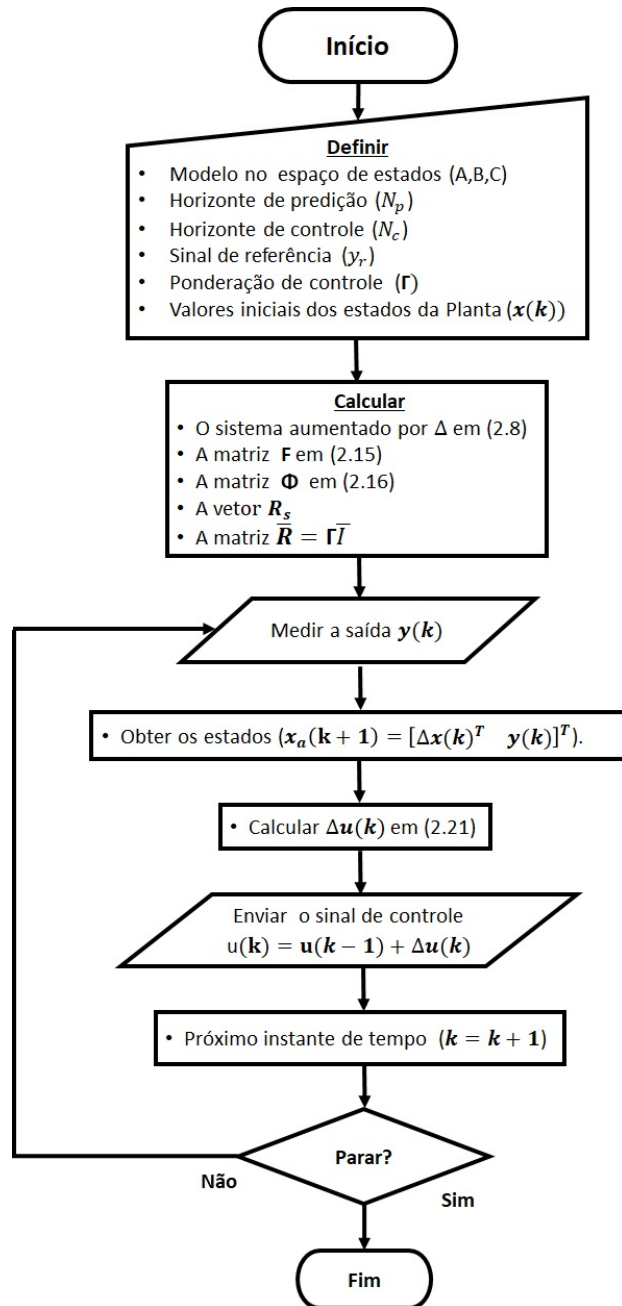


Figura 7 – Fluxograma do projeto do Controlador *CDMPC*.

predição N_p , em que para o controle do processo é utilizando apenas a primeiro termo $\Delta u(k_i)$, ignorando o resto da trajetória e repetindo-se esse procedimento de otimização a cada período de amostragem T_s .

Foram apresentados os parâmetros de sintonia do controlador *CDMPC*, definidos por N_p , N_c e Γ , onde suas características e influências em relação ao desempenho do sistema foram mostradas usando-se resultados simulados para um modelo de processo específico deste trabalho.

Por fim, foi apresentada a configuração em malha fechada do modelo preditivo no espaço de estados, apresentando os ganhos de malha fechada K_y e K_{mpc} , que são necessários para obter o modelo do sistema em malha fechada para a análise da estabilidade.

No Apêndice C, são mostrados os códigos do controlador *CDMPC* desenvolvidos no *MATLAB*[®], onde são apresentados exemplos para o caso SISO e MIMO, nos códigos 5 e 6, respectivamente.

Conteúdo similar ao apresentado neste capítulo pode ser estudado em, Chai, Wang e Rogers (2011b), Chai, Wang e Rogers (2011a), Huyck *et al.* (2013) e Mateescu, Stefanoiu e Christov (2013). Estes trabalhos apresentam o mesmo projeto de MPC descrito neste capítulo, além de demonstrarem os resultados obtidos em várias categorias de plantas.

DMPC USANDO FUNÇÕES DE LAGUERRE

3.1 Introdução

As Funções de Laguerre ou Polinômios de Laguerre são uma família de polinômios que foram desenvolvidas por Edmond Nicolas Laguerre (9 de Abril, 1834 - 14 de Agosto, 1886), um matemático francês que tinha suas pesquisas voltadas principalmente para as áreas da geometria e de polinômios ortonormais (BATEMAN *et al.*, 1955). No universo da engenharia, essas funções também têm sido utilizadas para identificação de sistemas, isso por causa da sua eficiência em melhorar a precisão numérica das soluções de problemas de estimação com regressão linear (WAHLBERG, 1991; AKÇAY; NINNESS, 1999).

As Funções de Laguerre com base ortonormal também têm sido aplicadas no controle preditivo, sendo utilizadas para simplificar a solução ótima da trajetória futura do sinal de controle. Então, uma predição satisfatória da trajetória futura do sinal de controle pode ser obtida sem a necessidade de um grande horizonte de predição. Isto leva a um sinal de controle mais suave e a diminuição da carga computacional, contrastando com o controle DMPC clássico descrito no Capítulo 2, onde a solução ótima da trajetória futura de controle é obtida através de operadores de avanço discreto (forward shift operator), que requer um grande número de termos adicionais devido ao aumento nos valores de N_c e N_p para os casos em que é desejado um elevado desempenho de malha fechada com uma elevada taxa de amostragem, podendo levar a uma solução ótima inadequada e à grande carga computacional.

Na literatura existem outros tipos de funções ortonormais que podem ser usadas em projetos de controladores preditivos, dentre as quais pode-se destacar as funções ortonormais de Kautz (MINGZHU *et al.*, 2009; YAKUB; MORI, 2014) e as funções ortonormais de Legendre (DUBRAVIĆ; ŠEHIĆ; BURGIĆ, 2014). Contudo, as Funções de Laguerre se destacam por serem uma cadeia de filtros que formam um grupo de funções de ordem N , onde a expansão ortonormal convergirá para uma trajetória de controle ótimo (WANG, 2009).

Neste capítulo será apresentado o projeto do Controlador Preditivo Discreto Baseado

em Modelo com Funções de Laguerre (LDMPC, Laguerre Discrete-Time Model Predictive Control) para o caso *sem restrição*, detalhando-se a definição das Funções de Laguerre com base ortonormal, o projeto do controlador, suas principais características, a lei de controle de horizonte retrocedente e os parâmetros de sintonia envolvidos. Além disso, demonstra-se-á as vantagens das funções ortonormais quando aplicadas a controladores preditivos e sua simples implementação em processos SISO e MIMO.

3.2 Funções de Laguerre com base ortonormal

Como é demonstrado em Wahlberg (1991), a rede de filtros de Laguerre (Laguerre network) no tempo discreto pode ser representada por uma sequência descrita por,

$$\begin{aligned}\varphi_1(z) &= \frac{\sqrt{(1-a^2)}}{1-az^{-1}} \\ \varphi_2(z) &= \frac{\sqrt{(1-a^2)}}{(1-az^{-1})} \frac{z^{-1}-a}{(1-az^{-1})} \\ &\vdots \\ \varphi_N(z) &= \sqrt{(1-a^2)} \frac{(z^{-1}-a)^{N-1}}{(1-az^{-1})^N}\end{aligned}\tag{3.1}$$

onde N e a são, respectivamente, a ordem e polo discreto da rede de filtros de Laguerre, sendo necessário $0 \leq a < 1$ para garantir a estabilidade da rede. No domínio da frequência φ_m , $m = 1, 2, \dots, N$, essa rede de filtros podem ser expressas como equações ortonormais dadas por,

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \varphi_m(e^{j\omega}) \varphi_r(e^{j\omega})^* d\omega = 1 \quad m = r\tag{3.2}$$

$$\frac{1}{2\pi} \int_{-\pi}^{\pi} \varphi_m(e^{j\omega}) \varphi_r(e^{j\omega})^* d\omega = 0 \quad m \neq r\tag{3.3}$$

sendo $f(\cdot)^*$ o complexo conjugado de $f(\cdot)$, onde $f(\cdot)$ representa uma função de variável complexa.

As Funções de Laguerre são obtidas através da transformada z inversa da rede de filtros de Laguerre,

$$l(k) = \mathcal{Z}^{-1} \{ \varphi(z) \}\tag{3.4}$$

as quais para controle preditivo no caso SISO podem ser expressas por um vetor na forma,

$$L(k) = \begin{bmatrix} l_1(k) & l_2(k) & \cdots & l_N(k) \end{bmatrix}^T\tag{3.5}$$

onde sua condição inicial é dada pelo vetor:

$$L(0)^T = \sqrt{1-a^2} \begin{bmatrix} 1 & -a & a^2 & \cdots & (-1)^{N-1} a^{N-1} \end{bmatrix}\tag{3.6}$$

Para o caso MIMO as condições iniciais das Funções de Laguerre são expressas por uma Matriz que tem a estrutura dada por,

$$L_{MIMO}(0) = \begin{bmatrix} L_1(0)^T & \sigma_2^T & \cdots & \sigma_n^T \\ \sigma_1^T & L_2(0)^T & \cdots & \sigma_n^T \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^T & \sigma_2^T & \cdots & L_i(0)^T \end{bmatrix} \quad (3.7)$$

onde n é o número de entradas, σ_i é um vetor coluna nulo com comprimento N_i , $i = 1, 2, \dots, n$. A matriz $L_{MIMO}(0)$ tem dimensão $N_{sum} \times n$, $N_{sum} = N_1 + N_2 + \dots + N_i$. No caso MIMO cada entrada tem sua respectiva rede de Filtros de Laguerre, cada uma com sua ordem N_i e seus polos a_i . Isso permite maior grau de liberdade de sintonia, apesar de aumentar o número de parâmetros de sintonia. Podendo-se assim obter um desempenho desejado para cada entrada e saída do sistema.

De acordo com a Equação 3.5, para o caso SISO, $L(k)$ pode ser resolvido recursivamente por,

$$L(k) = A_l L(k-1) \quad (3.8)$$

onde A_l é uma matriz de Toeplitz (BÖTTCHER; GRUDSKY, 2000), com dimensões $(N \times N)$ composta pelos parâmetros a e $\beta = (1 - a^2)$,

$$A_l = \begin{bmatrix} a & 0 & 0 & 0 & \cdots & 0 \\ \beta & a & 0 & 0 & \cdots & 0 \\ -a\beta & \beta & a & 0 & \cdots & 0 \\ a^2\beta & -a\beta & \beta & a & \cdots & 0 \\ \vdots & \vdots & \vdots & \ddots & \ddots & 0 \\ (-1)^{N-2}a^{N-2}\beta & (-1)^{N-3}a^{N-3}\beta & \cdots & \cdots & \beta & a \end{bmatrix} \quad (3.9)$$

enquanto para o caso MIMO faz-se,

$$L_{MIMO}(k) = \begin{bmatrix} A_{l_1} & \rho_{N_1 \times N_2} & \cdots & \rho_{N_1 \times N_i} \\ \rho_{N_2 \times N_1} & A_{l_2} & \cdots & \rho_{N_2 \times N_i} \\ \vdots & \vdots & \ddots & \vdots \\ \rho_{N_i \times N_1} & \rho_{N_i \times N_2} & \cdots & A_{l_i} \end{bmatrix} L_{MIMO}(k-1) \quad (3.10)$$

sendo ρ uma matriz de zeros com dimensão $N \times N$.

As Funções expressas no domínio da frequência pelas Equações 3.2 e 3.3 também podem ser expressas no domínio discreto:

$$\sum_{k=0}^{\infty} l_i(k)l_j(k) = 0 \quad i \neq j \quad (3.11)$$

$$\sum_{k=0}^{\infty} l_i(k)l_j(k) = 1 \quad i = j \quad (3.12)$$

Esta ortonormalidade expressa pelas Funções de Laguerre será usada para projetar o controlador preditivo discreto.

3.3 Controle Ótimo usando Funções Laguerre

A fim de facilitar o entendimento, inicialmente será abordado o controle preditivo para o caso SISO e posteriormente para o caso MIMO. No projeto deste controlador preditivo é utilizado um conjunto de Funções de Laguerre, $l_1(m), l_2(m), \dots, l_N(m)$, para representar uma trajetória futura, $\Delta u(k_i), \Delta u(k_i + 1), \dots, \Delta u(k_i + m)$ para cada instante de tempo k . Isso difere do *CDMPC* que usa operadores de avanço discreto para obter a trajetória futura do sinal de controle para um N_c . Esta abordagem usando funções ortonormais pode ser aplicadas para expressar o incremento do sinal de controle,

$$\Delta u(k_i + m|k_i) = \sum_{j=1}^N p_j(k_i) l_j(m) = L(m)^T \eta \quad (3.13)$$

$$\Delta U_L = \begin{bmatrix} L(k_i)^T \eta & L(k_i + 1)^T \eta & L(k_i + 2)^T \eta & \cdots & L(k_i + m)^T \eta \end{bmatrix} \quad (3.14)$$

onde η é um vetor de coeficientes dado por,

$$\eta = \begin{bmatrix} p_1 & p_2 & \cdots & p_N \end{bmatrix}^T \quad (3.15)$$

Estes coeficientes $p_j, j = 1, 2, \dots, N$ são os valores iniciais para o instante inicial k_i do horizonte de predição de controle. O vetor η é equivalente ao vetor ΔU que contém a predição da trajetória futura do sinal incremental de controle do controlador *CDMPC*.

No projeto do controlador *LDMPC* o parâmetro N_c é desconsiderado e substituído por N que é a ordem das Funções de Laguerre, que agora define o tamanho do horizonte que descreve a trajetória futura do sinal de controle. Em 3.13 é utilizada a propriedade ortonormal das Funções de Laguerre vista em 3.11. Assim com um adequado N_p é possível garantir o decaimento para zero de $\Delta u(k_i + m|k_i)$, como mostra a propriedade a seguir:

$$\sum_{m=0}^{N_p} l_i(m) l_j(m) = 0 \quad i \neq j \quad (3.16)$$

Deve-se destacar que no controlador *LDMPC* a trajetória futura do sinal incremental de controle é definida por ΔU_L em 3.14, que tem um comprimento $m = 1, 2, \dots, N_p$, suficientemente grande para decair $\Delta u(k_i + m|k_i)$ para zero, enquanto sua predição de controle com as dinâmicas futuras são determinadas por η . Isso difere do controlador *CDMPC*, que tem tanto o comprimento do vetor ΔU quanto a predição de controle são determinados por N_c .

Quando são usadas as Funções de Laguerre para um sistema no espaço de estados (A_a, B_a, C_a) aumentado por Δ , a predição dos estados futuros da planta para um instante m é dada por,

$$x_a(k_i + N_p|k_i) = A_a^{N_p} x_a(k_i) + \sum_{i=0}^{N_p-1} A_a^{N_p-i-1} B_a L(i)^T \eta \quad (3.17)$$

onde $\Delta u(k_i + i)$ da Equação 2.9 foi substituído por $L(i)^T \eta$. Similarmente, a predição da saída é dada por,

$$y(k_i + N_p | k_i) = C_a A_a^{N_p} x_a(k_i) + \sum_{i=0}^{N_p-1} C_a A_a^{N_p-i-1} B_a L(i)^T \eta \quad (3.18)$$

Assim, a predição das variáveis de estado e de saída do sistema estão em função do vetor η . A otimização e os cálculos dos coeficientes do vetor η serão demonstrados a partir da minimização da função custo, sendo isso tratado detalhadamente nos subtópicos a seguir.

3.3.1 Função Custo

Considerando a função custo da Equação 2.17 do Capítulo 2, o vetor ΔU e \bar{R} podem ser substituídos, respectivamente, por seus equivalentes, vetor η e R_L . Então a função custo pode ser reescrita na forma dada por:

$$J = \sum_{m=1}^{N_p} (y_r(k_i) - y(k_i + m | k_i))^T (y_r(k_i) - y(k_i + m | k_i)) + \eta^T R_L \eta \quad (3.19)$$

Essa função custo é baseada na minimização do erro entre o sinal de referência e o sinal de saída. Esta função pode ser reformulada para conectar o controle MPC com o Regulador Linear Quadrático no tempo Discreto (DLQR, Discrete-Time Linear Quadratic Regulators) (KAILATH, 1980; BAY, 1999), e assim encontrar os coeficientes do vetor η que minimizem a função custo dada por,

$$J = \sum_{m=1}^{N_p} x_a(k_i + m | k_i)^T Q x_a(k_i + m | k_i) + \eta^T R_L \eta \quad (3.20)$$

onde $Q \geq 0$ e $R_L \geq 0$ são as matrizes pesos, sendo Q uma matriz quadrada com dimensão $m_1 \times m_1$, R_L é uma matriz diagonal com dimensão $N \times N$, tendo sua diagonal multiplicada por um escalar $\Gamma \geq 0$.

Para incluir o sinal de referência na Equação 3.20, é necessário fazer uma modificação na variável de estados $x_a(k_i + m | k_i)$ para:

$$x_a(k_i + m | k_i) = \begin{bmatrix} \Delta x(k_i + m | k_i)^T & y(k_i + m | k_i) - y_r(k_i) \end{bmatrix}^T \quad (3.21)$$

Então, com $Q = C_a^T C_a$, a Equação 3.20 torna-se equivalente à função 3.19.

3.3.2 Minimização da Função Custo

Reescrevendo a Equação 3.17, referente à predição dos estados da planta, para a forma,

$$x_a(k_i + m | k_i) = A_a^m x_a(k_i) + \phi(m)^T \eta \quad (3.22)$$

onde,

$$\phi(m)^T = \sum_{i=0}^{m-1} A_a^{m-i-1} B_a L(i) \quad (3.23)$$

sendo $\phi(m)^T$ uma matriz com dimensão $m_1 \times N$. Onde a soma da convolução de $\phi(m)^T$ pode ser calculada recursivamente através de uma Equação linear descrita por,

$$S_c(m) = A_a S_c(m-1) + S_c(1)(A_l^{m-1})^T \quad (3.24)$$

onde $S_c(1) = B_a L(0)^T$ e $m = 2, 3, 4, \dots, N_p$.

Substituindo a Equação 3.22 em 3.20, é obtido,

$$J = \eta^T \left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L \right) \eta + 2\eta^T \left(\sum_{m=1}^{N_p} \phi(m) Q A_a^m \right) x_a(k_i) + \sum_{m=1}^{N_p} x_a(k_i)^T (A_a^T)^m Q A_a^m x_a(k_i) \quad (3.25)$$

e aplicando a derivada parcial e igualando a zero,

$$\frac{\partial J}{\partial \eta} = 2 \left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L \right) \eta + 2 \left(\sum_{m=1}^{N_p} \phi(m) Q A_a^m \right) x_a(k_i) \quad (3.26)$$

e assumindo que $\left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L \right)^{-1}$ existe quando $\frac{\partial J}{\partial \eta} = 0$, então a solução ótima para η é dada por,

$$\eta = - \overbrace{\left(\sum_{m=1}^{N_p} \phi(m) Q \phi(m)^T + R_L \right)^{-1}}^{\Omega} \overbrace{\left(\sum_{m=1}^{N_p} \phi(m) Q A_a^m \right) x_a(k_i)}^{\Psi} \quad (3.27)$$

podendo ser expressa em sua forma compacta,

$$\eta = -\Omega^{-1} \Psi x_a(k_i) \quad (3.28)$$

onde as matrizes Ω e Ψ têm as dimensões $N \times N$ e $N \times m_1$, respectivamente.

Assim, 3.25 pode ser reescrita na forma:

$$J = \eta^T \Omega \eta + 2\eta^T \Psi x_a(k_i) + \sum_{m=1}^{N_p} x_a(k)^T (A_a^T)^m Q A_a^m x_a(k_i) \quad (3.29)$$

3.3.3 Controle de Horizonte Retrocedente

Como já dito anteriormente, o controle de horizonte retrocedente se caracteriza por utilizar apenas o primeiro termo da trajetória de controle incremental predita (ΔU_L), definida em 3.14, para cada instante de tempo k , sendo o restante da trajetória descartada. Tendo em vista isso, a lei de controle de horizonte retrocedente do controlador *LDMPC* para o caso SISO *sem restrição* é definida por,

$$\Delta u(k) = L(0)^T \eta \quad (3.30)$$

que pode ser reescrita para uma forma mais usual e familiar,

$$\Delta u(k) = -K_{mpc} x_a(k) \quad (3.31)$$

onde o ganho de malha fechada para o caso SISO é obtido por,

$$K_{mpc} = L(0)^T \Omega^{-1} \Psi \quad (3.32)$$

Sabendo-se que nessa abordagem o sinal de referência está contido em $x_a(k_i)$, isso por causa da predição dos estados estar baseada nas atuais informações das variáveis de estado, o vetor de estados aumentados do sistema é modificado para,

$$x_a(k) = \begin{bmatrix} \Delta x(k)^T & y(k) - y_r(k) \end{bmatrix}^T \quad (3.33)$$

como foi evidenciado em 3.21.

3.3.4 Sistema em Malha Fechada

De posse do ganho K_{mpc} , é possível obter o sistema de malha fechada, dado por,

$$x_a(k_i + 1) = (A_a - B_a K_{mpc}) x_a(k_i) \quad (3.34)$$

assim podendo examinar a estabilidade e a posição dos polos de malha fechada do sistema. Essa configuração se difere da apresentada em 2.28 para o MPC em sua abordagem clássica. Isso porque neste projeto, o sinal de referência já está incluído no vetor de estados aumentados $x_a(k_i)$.

3.4 Caso MIMO

Para o caso MIMO são feitas algumas extensões dos vetores e matrizes usadas no caso SISO. Para cada entrada da planta é projetado um conjunto de Funções de Laguerre com seus respectivos polos (a) e ordem (N). A matriz de entrada B_a é particionada em colunas:

$$B_a = \begin{bmatrix} B_1 & B_2 & \dots & B_i \end{bmatrix} \quad (3.35)$$

sendo cada coluna de B_a relacionada a uma entrada do sistema e $i = 1, 2, \dots, n$. A predição das variáveis de estado e da saída do sistema para o caso MIMO são dadas, respectivamente, por,

$$x_a(k_i + m | k_i) = A_a^m x_i(k) + \sum_{j=0}^{m-1} A_a^{m-i-1} \begin{bmatrix} B_1 L_1(i) & B_2 L_2(i) & \dots & B_i L_i(j) \end{bmatrix} \eta \quad (3.36)$$

$$y(k_i + m | k_i) = C_a^m x_a(k_i) + \sum_{j=0}^{m-1} C_a A_a^{m-i-1} \begin{bmatrix} B_1 L_1(i) & B_2 L_2(i) & \dots & B_i L_i(j) \end{bmatrix} \eta \quad (3.37)$$

sendo o vetor η no caso MIMO, definido por,

$$\eta^T = \begin{bmatrix} \eta_1^T & \eta_2^T & \dots & \eta_i^T \end{bmatrix} \quad (3.38)$$

com dimensão $N_{sum} \times 1$.

A matriz $\phi(m)^T$ é estendida para,

$$\phi(m)^T = \sum_{j=0}^{m-1} A_a^{m-i-1} \begin{bmatrix} B_1 L_1(i) & B_2 L_2(i) & \dots & B_i L_i(j) \end{bmatrix} \eta \quad (3.39)$$

podendo ser calculada a partir da convolução da soma que é descrita por:

$$S_c(m) = A_a^{m-1}S(1) + \left[B_1L_1(m-1)^T \quad B_2L_2(m-1)^T \quad \cdots \quad B_iL_i(m-1)^T \right] \quad (3.40)$$

Para caso MIMO, a função custo continua a mesma definida em 3.29 e as matrizes Ω e Ψ são calculadas, respectivamente, por,

$$\Omega = \sum_{m=1}^{N_p} \phi(m)Q\phi(m)^T + R_L \quad (3.41)$$

$$\Psi = \sum_{m=1}^{N_p} \phi(m)QA_a^m \quad (3.42)$$

O valor ótimo para η , *sem restrições*, também continua sendo calculado por 3.28.

Para o caso MIMO, a lei de controle de horizonte retrocedente *sem restrições* é obtida por,

$$\Delta u(k) = -K_{mpc}x_a(k) \quad (3.43)$$

sendo o ganho de malha fechada para o caso MIMO dado por,

$$K_{mpc} = \left[\begin{array}{cccc} L_1(0)^T & \sigma_2^T & \cdots & \sigma_n^T \\ \sigma_1^T & L_2(0)^T & \cdots & \sigma_n^T \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^T & \sigma_2^T & \cdots & L_i(0)^T \end{array} \right] \Omega^{-1}\Psi \quad (3.44)$$

Para um melhor entendimento de todo o projeto apresentado, nas Figuras 8 e 9 apresentam-se o fluxograma e o diagrama de blocos do projeto do controlador *LDMPC*, respectivamente.

3.5 Caso especial para $a = 0$

Existe um caso especial para as Funções de Laguerre que é justamente quando $a = 0$. Quando isso ocorre a matriz A_{l_n} em 3.9 torna-se,

$$A_l = \left[\begin{array}{cccccc} 0 & 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & 0 & \cdots & 0 & 0 \\ 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \end{array} \right] \quad (3.45)$$

e o vetor dos valores iniciais em 3.6 torna-se,

$$L(0)^T = \left[1 \quad 0 \quad \cdots \quad 0 \right] \quad (3.46)$$

assim tem-se, como resultado, $l_1(k) = \delta(k)$, $l_2(k) = \delta(k-1)$, \dots , $l_N = (k-N+1)$. Então, as Funções de Laguerre tornam-se equivalentes a um conjunto de pulsos de avanço discreto, transformando o controlador *LDMPC* em um controlador *CDMPC*, demonstrado no Capítulo 2. Assim sendo, quando $a = 0$, $N = N_c$.

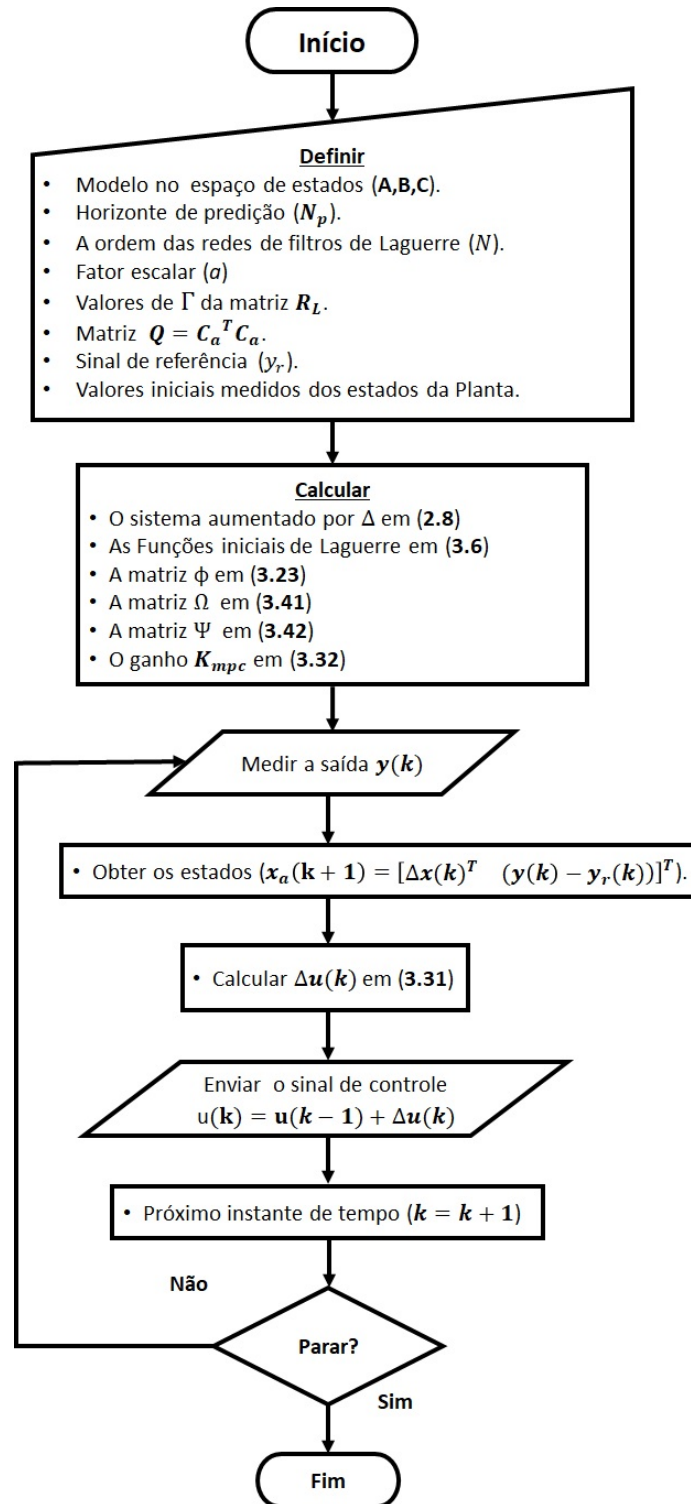
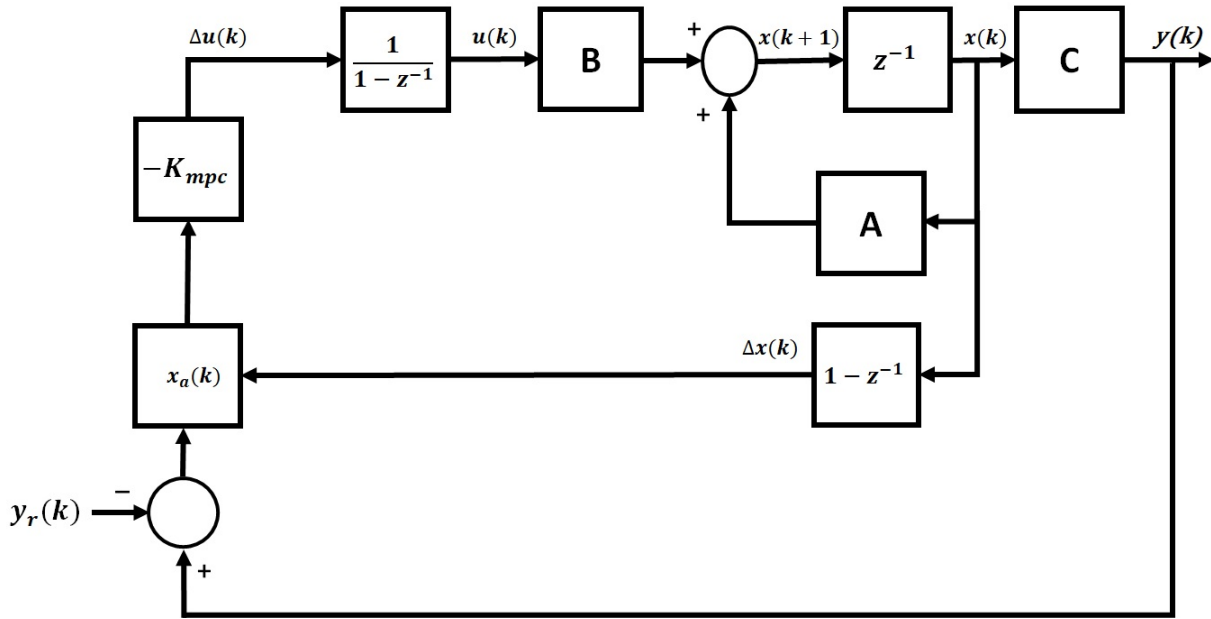


Figura 8 – Fluxograma do projeto do controlador LDMPC.

3.6 Parâmetros de Sintonia (R_L, Q, N, a)

Sem dúvida uma das maiores desvantagens do controlador proposto é a dificuldade de sintonia do sistema para o desempenho desejado, isso por causa do número de parâmetros utilizados para sintonizar a planta, tornando-se algo trabalhoso e consumindo um grande tempo

Figura 9 – Diagrama de blocos do projeto do controlador *LDMPC*.

para obter o desempenho desejado através de tentativa e erro. Para o caso multivariável essa complexidade aumenta pelo fato de haver um grande número de parâmetros para cada uma das entradas e saídas do sistema. Nos subtópicos a seguir são abordados os parâmetros de sintonia do controlador *LDMPC*, detalhando suas características e as influências no desempenho do sistema.

3.6.1 Parâmetros (R_L , Q)

R_L , Q são as matrizes de ponderação do sistema e têm a função similar às matrizes R e Q de um controlador *DLQR*. A matriz $Q = C_a^T C_a$ é usada nesta configuração para minimizar os erros na saída do sistema. Essa escolha produz uma excelente desempenho de malha fechada para um sistema em espaço de estados aumentado por Δ (WANG, 2009).

A matriz R_L é definida por:

$$R_L = \begin{bmatrix} \Gamma_1 I_{N_1 \times N_1} & v_{N_1 \times N_2} & \cdots & v_{N_1 \times N_i} \\ v_{N_2 \times N_1} & \Gamma_2 I_{N_2 \times N_2} & \cdots & v_{N_2 \times N_i} \\ \vdots & \vdots & \ddots & \vdots \\ v_{N_i \times N_1} & v_{N_i \times N_2} & \cdots & \Gamma_i I_{N_i \times N_i} \end{bmatrix} \quad (3.47)$$

onde $I_{N \times N}$ é uma matriz identidade, $i = 1, 2, \dots, n$, $v_{N \times N}$ é uma matriz de zeros e R_L é uma matriz com dimensão $N_{sum} \times N_{sum}$. $\Gamma \geq 0$ é usada para sintonizar a velocidade de resposta de malha fechada de cada saída do sistema. Sendo que, quanto menor o peso de Γ maior a velocidade de resposta do sistema em malha fechada. Em um sistema multivariável pode ser escolhido um Γ para cada entrada do sistema, assim sintonizando separadamente cada saída de acordo com o desempenho desejado.

3.6.2 Parâmetros (N, a)

O parâmetro $N = \{N \in \mathbb{N} \text{ e } 1 \leq N\}$ é a ordem da cadeia de filtros de Laguerre. Em outras palavras, é o número de termos usados para representar a dinâmica futura do incremento do sinal de controle e tem uma função similar a de N_c do controlador *CDMPC*.

O parâmetro a é o polo discreto das Funções de Laguerre, tendo seu valor $0 \leq a < 1$ para garantir a estabilidade da cadeia de filtros de Laguerre. Na literatura é também chamado como fator escalar (*scaling factor*). Em geral, o parâmetro a é selecionado como uma estimativa da parte real dos polos dominantes de malha fechada. A taxa de decaimento do sinal de controle incremental depende diretamente da escolha de N e a . Essa relação pode ser descrita por,

$$a \approx e^{\frac{-\gamma}{N}} \quad (3.48)$$

onde γ é a taxa de decaimento definida por um número real $\gamma, \gamma > 0$. Quando $a = 0$, as redes de filtros de Laguerre se tornam pulsos de avanço discreto, tornando o controlador *LDMPC* equivalente ao controlador *CDMPC*.

Para deixar mais clara a influência dos parâmetros N e a , foram realizadas simulações para os valores de $a = 0, 1; 0, 2; 0, 3; 0, 4; 0, 8; 0, 9$. Primeiramente para uma ordem de filtros de Laguerre pequena ($N = 3$) e depois para uma ordem grande ($N = 10$). Usando uma função de transferência que representa a dinâmica de uma junta robótica do manipulador Mentor (ver Capítulo 5), definida por $G_{11}(z) = \frac{-0,00072717z - 0,0039167}{z^2 - 1,6042z + 0,60415}$ com período de amostragem $T_s = 0, 1$ e $N_p = 50, \Gamma = 0, 01, Q = C_a^T C_a$.

Na Figura 10 mostra-se os resultados para a variação de a com $N = 3$. Analisando esses resultados, pode-se verificar que é possível obter uma grande variação no comportamento do sistema em malha fechada, sendo uma excelente maneira de fazer um *ajuste fino* no desempenho desejado. Contudo, também verifica-se que para um $a \approx 1$ resulta em um sinal de controle com uma lenta taxa de decaimento.

Já na Figura 11, apresentam-se os resultados para a variação de a com $N = 10$, onde é possível observar que não ocorre uma grande mudança no desempenho do sistema. Como consequência o controlador *LDMPC* torna-se equivalente ao controlador *DLQR* para os casos onde é usada uma rede de filtros de Laguerre com ordem elevada, como é demonstrado em (WANG, 2009)

Portanto, para uma maior liberdade de sintonia (*ajuste fino*) do sistema é recomendado usar valores pequenos para N , assim podendo obter melhores desempenhos de malha fechada de acordo com a necessidade do processo. Contudo, deve-se considerar que a definição de um N grande ou um N pequeno, depende da complexidade do sistema a ser controlado.

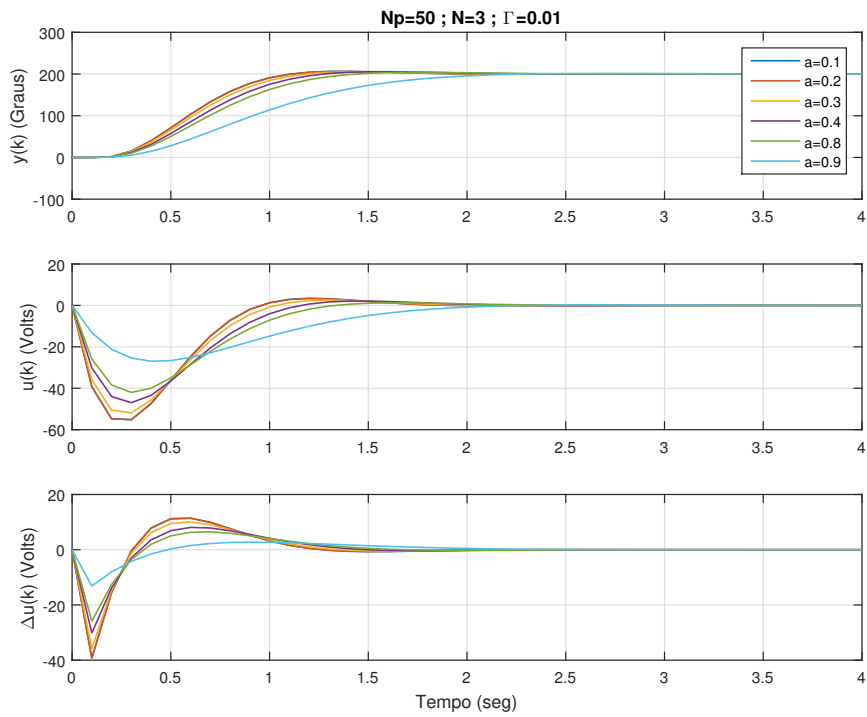


Figura 10 – Resposta para a variação de a com N pequeno.

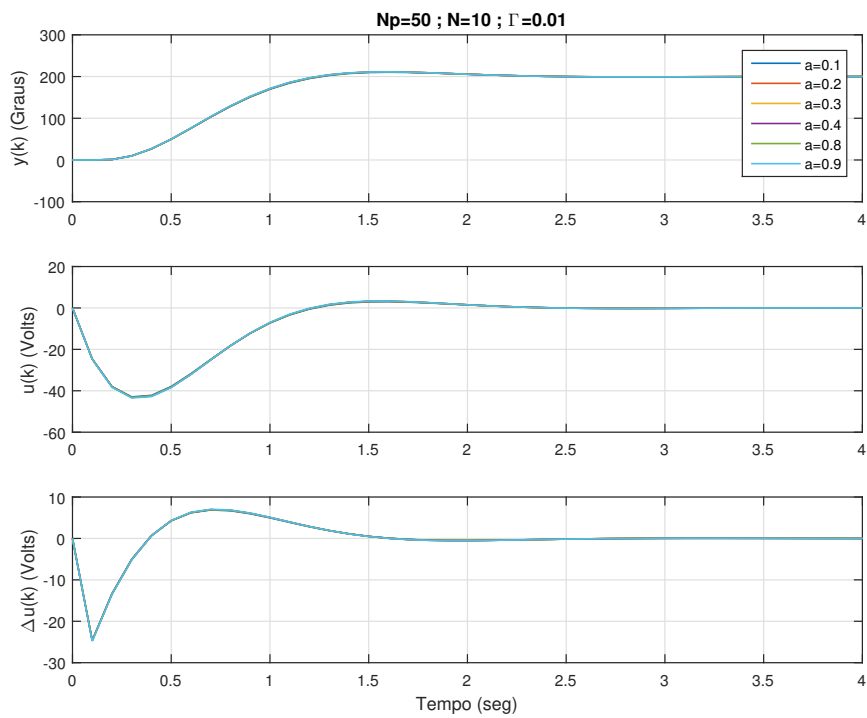


Figura 11 – Resposta para a variação de a com N grande.

3.7 Comparações de Desempenho entre LDMPC e CDMPC para o caso SISO

Neste tópico serão realizadas duas simulações para o mesmo sistema $G_{11}(z)$ das simulações anteriores, com objetivo de comparar o desempenho dos controladores *LDMPC* e *CDMPC* no caso SISO.

A primeira simulação, na Figura 12, apresenta os resultados dos controladores para os mesmos parâmetros de sintonia: $N_p = 50$, $N = N_c = 2$, $a = 0,4$ e $\Gamma = 0,01$. Analisando esses resultados é possível verificar que o controlador *LDMPC* obteve um melhor desempenho em malha fechada, no entanto, apresentando um pequeno sobre-sinal, um maior consumo de energia e uma maior variação no sinal de controle incremental. Isso demonstra a eficiência das Funções de Laguerre para melhorar o desempenho do controlador com um número mínimo de parâmetros de predição, tornando o horizonte de controle e de predição mais efetivos. Além disso, pode-se confirmar essa afirmação através dos índices de desempenho apresentados na Tab. 1, onde os índices utilizados foram:

- Integral do sinal de controle quadrático (ISU, Integral of Squared Control).

$$ISU = \sum_{k=1}^m u(k)^2 \quad (3.49)$$

- Integral do erro absoluto (IAE, integral absolute error).

$$IAE = \sum_{k=1}^m |e(k)| \quad (3.50)$$

- Integral do erro quadrático (ISE, Integral of Squared Error).

$$ISE = \sum_{k=1}^m e(k)^2 \quad (3.51)$$

- Variância Total do sinal de controle (TVC, Total Variation Of Control).

$$TVC = \sum_{k=1}^m |\Delta u(k)| \quad (3.52)$$

- Sobre-sinal (PO, Overshoot) .
- Tempo de assentamento (ST, Settling Time).

Já na Figura 13, são mostrados os dois controladores com desempenhos similares, porém, para que o controlador *CDMPC* obtivesse o mesmo desempenho, foi necessário aumentar seu horizonte de predição para $N_p = 100$ e o horizonte de controle para $N_c = 4$. Isso acaba tendo como consequência, uma maior carga computacional para o cálculo do sinal de controle.

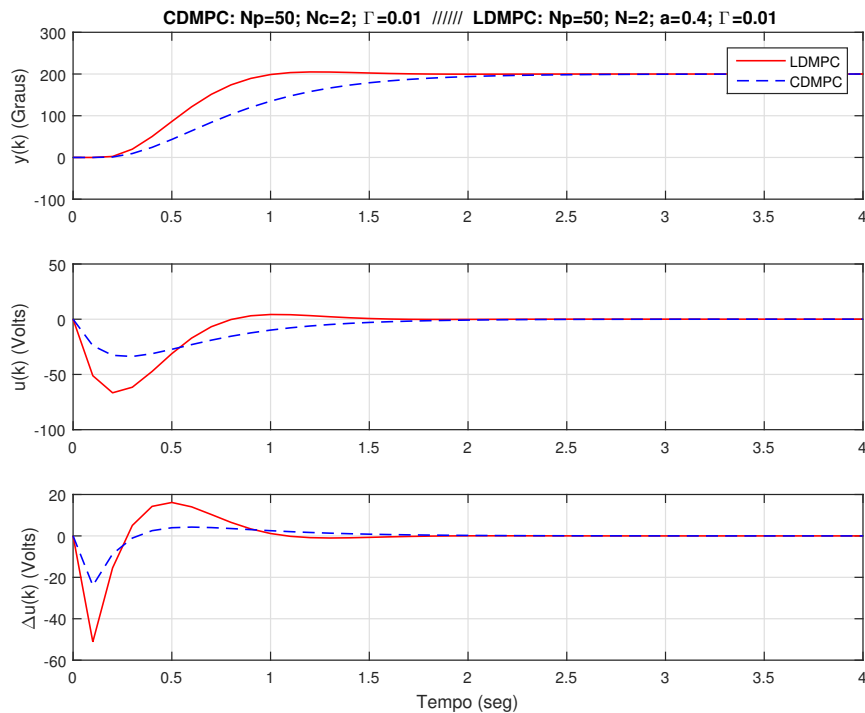


Figura 12 – Comparação desempenho entre os controladores *LDMPC* e *CDMPC* para os mesmos parâmetros de sintonia.

Tabela 1 – Índices de desempenho dos controladores *LDMPC* e *CDMPC* para os mesmos parâmetros de sintonia.

	LDMPC	CDMPC
ISU	302,724277	263,375160
ISE	37,638951	55,581627
IAE	15,968788	25,848851
PO(%)	0,494085	0
TVC	142,290971	67,534146
ST(seg)	1,0	2,6

Como é sabido, quanto maior for o horizonte de controle e de predição, maior será a carga computacional utilizada para calcular o sinal de controle em cada instante de tempo k . Tendo em vista isso, uma das principais vantagens do controlador *LDMPC* é o fato de obter um desempenho satisfatório sem a necessidade de um longo horizonte de predição, em outras palavras, o equivalente a um grande N_c pode ser obtido com um pequeno N . Isso afeta diretamente no custo computacional do processo, sendo interessante para sistemas *online* com alta taxa de amostragem. Além disso, o controlador *LDMPC* tem o parâmetro de sintonia a , ou seja, um parâmetro a mais de sintonia em relação ao *CDMPC*, levando a um maior grau de liberdade para realizar o chamado ajuste fino de sintonia do sistema.

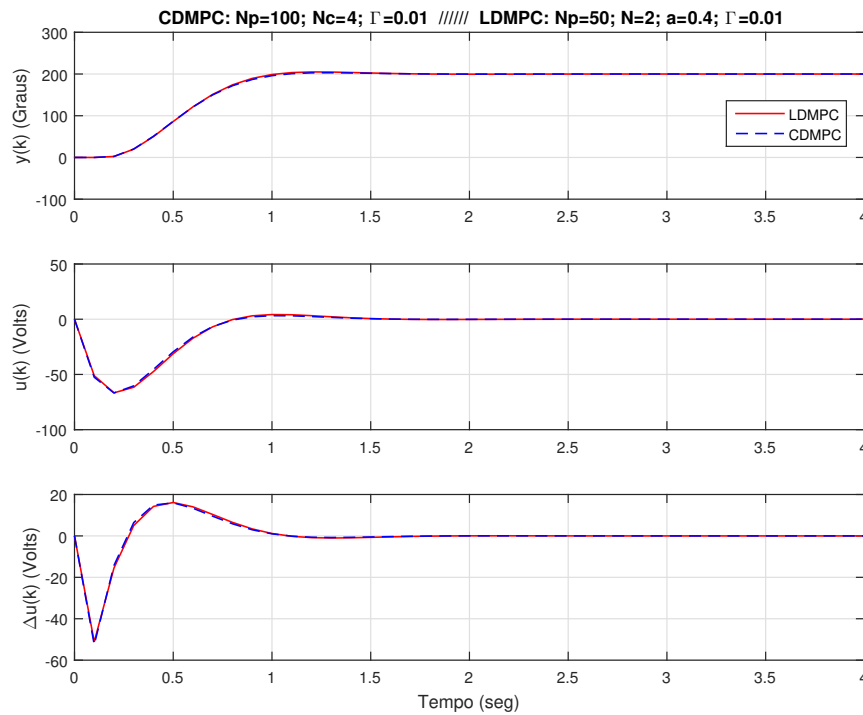


Figura 13 – Controladores *LDMPC* e *CDMPC* com o mesmo desempenho.

3.8 Conclusões

Neste capítulo foi apresentado o projeto do controlador *LDMPC*, que utiliza uma rede de Funções de Laguerre para obter a trajetória futura do sinal de controle. Por utilizar o modelo no espaço de estados, este controlador oferece a vantagem de sua simples implementação em sistemas MIMO. Além disto, apresenta um desempenho satisfatório sem a necessidade de um longo horizonte de predição e de controle, assim levando a um menor custo computacional para a realização do processo.

Foram também retratados os parâmetros de sintonia utilizados no controlador *LDMPC*, onde foram discutidas as influências dos parâmetros N e a no desempenho em malha fechada do sistema, além de apresentar o caso especial $a = 0$, que torna o controlador *LDMPC* equivalente ao controlador *CDMPC*, fazendo com que $N = N_c$. Além disso, também foi exposto o caso para N de ordem elevada, que deixa o controlador *LDMPC* com o comportamento similar a um controlador *DLQR*.

Foi apresentada a lei de controle de horizonte retrocedente, onde a trajetória de controle é determinada por η , que é obtida através da minimização de uma função custo J , sendo η equivalente ao vetor ΔU no controle preditivo em sua abordagem clássica apresentada no Capítulo 2.

Por fim, foi realizada a comparação de desempenho entre os controladores *CDMPC* e *LDMPC* para os mesmos horizontes de controle e de predição, onde através de simulações,

foi mostrado um melhor desempenho do controlador *LDMPC*, sendo necessário aumentar o horizonte de controle e de predição do controlador *CDMPC* para obter um desempenho similar, ocasionando em um maior custo computacional. Isso demonstra que as Funções de Laguerre tornam o horizonte de controle mais eficaz e otimizado, em relação ao clássico.

No Apêndice D, são mostrados os códigos dos algoritmos do controlador *LDMPC* desenvolvidos no *MATLAB*[®], onde são apresentados exemplos para o caso SISO e MIMO, nos códigos 14 e 15, respectivamente.

Existem muitos trabalhos que podem se destacar como leituras complementares para reforçar os pontos defendidos neste capítulo. Trabalhos como Wang (2004) onde a ideia inicial do *MPC* usando as Funções de Laguerre foi demonstrada; Rossiter, Wang e Valencia-Palomo (2010) e Rossiter e Wang (2008) abordam a utilização das Funções de Laguerre no projeto do *MPC* com o objetivo de melhorar a eficiência entre a viabilidade do sinal de controle e o desempenho de malha fechada do sistema.

DMPC COM RESTRIÇÕES

4.1 Introdução

Na prática, todos os processos estão sujeitos a restrições. Porém, a inclusão de restrições no sistema pode levar a uma significativa perda de desempenho ou estabilidade. Essas restrições geralmente são aplicadas no sinal de controle $u(k)$, nas saídas do sistema $y(k)$ e nas variáveis de estado $x(k)$. No entanto, neste trabalho será somente abordado caso de restrições nas variáveis de entrada da planta, focando nas variáveis $u(k)$ e $\Delta u(k)$, com o objetivo de explorar em controladores preditivos o conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada do sistema.

Uma das maiores vantagens dos controladores *MPC* é a facilidade em operar com sistemas MIMO com restrições, devido a sua simplicidade em trabalhar em conjunto com programações lineares ou quadráticas (JIMÉNEZ, 2000). O problema do controle com restrições para controladores preditivos é essencialmente um problema de programação quadrática, sendo as restrições um conjunto de inequações lineares que são resolvidas através de uma programação quadrática. Nisso pode-se destacar a programação quadrática de *Hildreth*, utilizada neste trabalho para resolver o problema de otimização da lei de controle para sistemas *MIMO* e *SISO* com restrições.

Para ficar mais claro o problema debatido, na Figura 14 é apresentado um problema recorrente do conflito mencionado, onde são realizadas duas simulações com o controlador CDMPC para um sistema SISO definido por $G_{11}(z)$ em 5.2. Na primeira simulação, o sinal de controle representado pela linha vermelha não sofre restrições. Porém, sendo um sinal de controle inviável para a aplicação em um sistema real, tendo seu valor superior a -30 Volts. Na segunda, o sinal de controle representado pela linha tracejada azul, sofre uma saturação dada por $-12 \leq u(k) \leq +12$ e $-2 \leq \Delta u(k) \leq +2$, para tornar viável a sua aplicação em uma planta real, ocasionando em uma perda considerável de desempenho. Portanto, as restrições impostas às variáveis de entrada do sistema, afetam o desempenho da saída, podendo levar à instabilidade

ou a um desempenho insatisfatório.

Tendo em vista isso, este trabalho apresenta formas de reduzir o conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada em controladores preditivos, sem a necessidade de uma alta carga computacional e ainda garantindo a estabilidade do sistema.

Neste capítulo, será apresentada a programação quadrática de *Hildreth*, que trabalha em conjunto com os controladores preditivos para resolver o problema de otimização da lei de controle com restrições. Também será demonstrado o projeto dos controladores *CDMPC* e *LDMPC* com restrições na lei de controle, além de simulações no caso SISO, tendo o objetivo de demonstrar a eficiência da técnica abordada.

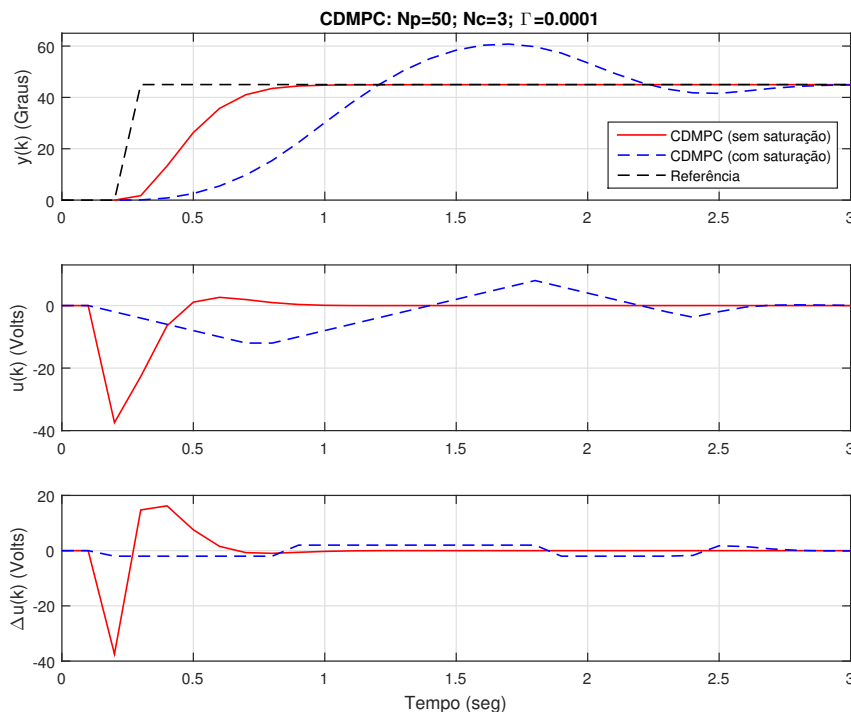


Figura 14 – Conflito entre viabilidade do sinal de controle e desempenho de malha fechada.

4.2 Programação quadrática

Neste tópico será realizada uma revisão sucinta sobre os fundamentos da programação quadrática. Tendo como objetivo contextualizar o leitor para um melhor entendimento da programação quadrática de *Hildreth* e sua aplicação em controladores preditivos com restrições.

4.2.1 Fundamentos da programação quadrática

O objetivo da programação quadrática é encontrar um vetor coluna v , com dimensão j , chamado na literatura de variável de decisão, que minimize a função custo dada por,

$$\bar{J} = \frac{1}{2}v^T E v + c^T v \quad (4.1)$$

estando sujeito às restrições,

$$Mv \leq \gamma \quad (4.2)$$

onde M é uma matriz com dimensões $j_1 \times j$, sendo j_1 o número de restrições. c é um vetor coluna com dimensão j , E é uma matriz Hessiana simétrica-positiva definida com dimensão $j \times j$ e γ é o vetor coluna dos termos independentes com dimensão j_1 .

A maioria dos métodos utilizados para se resolver programações quadráticas são categorizados dentro de conjuntos ativos (active set). Onde o conjunto de restrições pode ser ativos se $M_i v = \gamma_i$ ou inativo se $M_i v < \gamma_i$, sendo M_i e γ_i , elementos da matriz M e do vetor γ , respectivamente.

As restrições ativas pertencem ao grupo dos métodos primal e as restrições inativas pertencem ao grupo dos métodos dual. Para o caso $Mv < \gamma$, o método dual pode ser usado sistematicamente para resolver restrições inativas através de λ , chamado na literatura de multiplicador de Lagrange ou variável dual.

O problema primal é descrito por,

$$\max_{\lambda \geq 0} \min_v \left[\frac{1}{2}v^T E v + v^T c + \lambda^T (Mv - \gamma) \right] \quad (4.3)$$

obtido através da função de Lagrange, também conhecida como função objetivo dual, dada por,

$$\bar{J} = \frac{1}{2}v^T E v + v^T c + \lambda^T (Mv - \gamma) \quad (4.4)$$

A minimização da variável primal v , sem restrições ($\gamma = 0$), pode ser obtida por:

$$v = -E^{-1}(c + M^T \lambda) \quad (4.5)$$

Substituindo 4.5 em 4.3, tem-se o problema dual, que pode ser escrito como:

$$\max_{\lambda \geq 0} \left[-\frac{1}{2}\lambda^T H \lambda - \lambda^T K - \frac{1}{2}c^T E^{-1}c \right] \quad (4.6)$$

onde a matriz H e o vetor K são dados respectivamente por,

$$H = M E^{-1} M^T \quad (4.7)$$

$$K = \gamma + M E^{-1} c \quad (4.8)$$

Então, o problema dual em 4.6, torna-se um problema de programação quadrática descrito por,

$$\min_{\lambda \geq 0} \left[\frac{1}{2} \lambda^T H \lambda + \lambda^T K + \frac{1}{2} \gamma^T E^{-1} \gamma \right] \quad (4.9)$$

Assim, a função objetivo dual em 4.4, pode ser reescrita como,

$$\bar{J} = \frac{1}{2} \lambda^T H \lambda + \lambda^T K + \frac{1}{2} \gamma^T E^{-1} \gamma \quad (4.10)$$

sendo minimizada por um conjunto de multiplicadores de Lagrange sujeitos a restrição $\lambda \geq 0$.

Através da programação quadrática de Hildreth é possível resolver esse problema de programação quadrática. Assim podendo-se calcular o conjunto ótimo dos multiplicadores de Lagrange (λ_{calc}) que estão sujeitos a $\lambda \geq 0$, que minimize a função objetivo dual. Assim, a variável primal v pode ser calculada, diretamente, através de

$$v = -E^{-1}c - E^{-1}M^T \lambda_{calc} \quad (4.11)$$

tratando o problema como uma igualdade de restrições $Mv = \gamma$, resolvendo o problema de otimização com restrição.

4.2.2 Programação quadrática de Hildreth

A programação quadrática de Hildreth é um algoritmo desenvolvido por Hildreth (1957) e proposto para resolver problemas de programação quadrática do tipo dual (LUENBERGER, 1969; WISMER; CHATTERGY, 1978). Este algoritmo é um eficiente método para solucionar grandes sistemas de inequações matrizes esparsas (JAMIL; CHEN; CLONINGER, 2015). Sendo escolhido por causa de sua simples implementação e também pelo fato de que o algoritmo é baseado no acesso de elemento por elemento da matriz, portando, não requer inversão de matriz, diminuindo a carga computacional. Esse algoritmo pode ser expresso por,

$$\lambda_i^{m+1} = \max(0, w_i^{m+1}) \quad (4.12)$$

sendo,

$$w_i^{m+1} = -\frac{1}{h_{ii}} \left[k_i + \sum_{j=1}^{i-1} h_{ij} \lambda_j^{m+1} + \sum_{j=i+1}^n h_{ij} \lambda_j^m \right] \quad (4.13)$$

onde h_{ij} é um elemento da matriz $H = ME^{-1}M^T$ e k_i é um elemento do vetor $K = \gamma + ME^{-1}c$.

Este algoritmo é um método iterativo, onde para cada instante de tempo k , é obtido um vetor sujeito a restrição $\lambda \geq 0$. O cálculo é feito considerando apenas os elementos (λ_i) desse vetor, ajustando-o para minimizar a função de Lagrange, onde se $\lambda_i < 0$ é ajustado para $\lambda_i = 0$. Através dessa técnica é possível encontrar o vetor $\lambda_{calc} = \left[\lambda_i \quad \lambda_{i+1} \quad \dots \quad \lambda_{i+N-1} \right]^T$, que contém os valores ótimos dos multiplicadores de Lagrange que minimizam a função objetivo dual. Assim podendo calcular a variável primal (v) dada pela Equação 4.11.

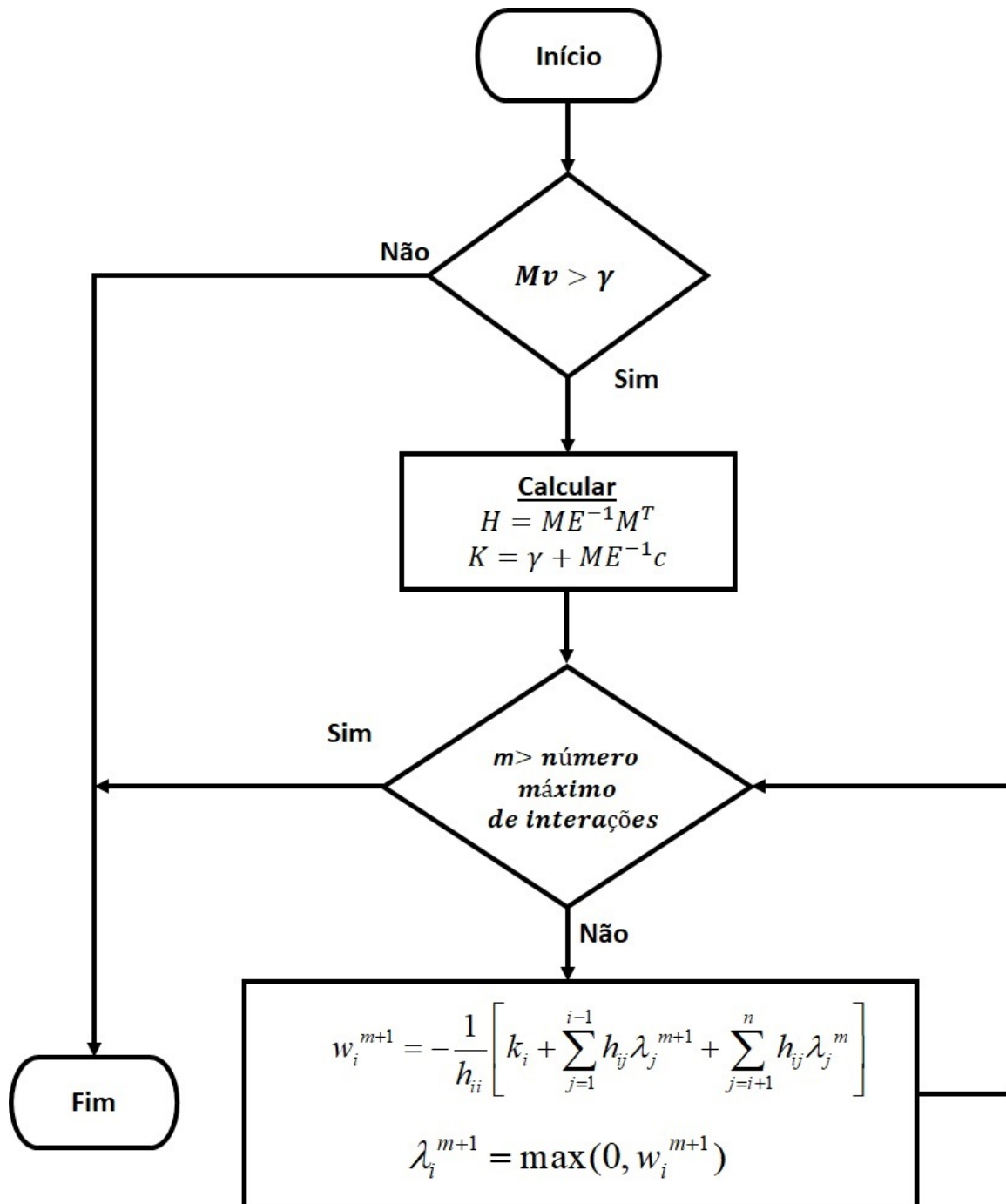


Figura 15 – Fluxograma da programação quadrática de Hildreth.

Para um melhor entendimento do processo, a Figura 15 apresenta o fluxograma da programação quadrática de Hildreth.

Existem vários outros métodos para solução deste problema quadrático clássico (BAZARAA; SHERALI; SHETTY, 2013; FLETCHER, 1987; PEREIRA, 1997). Porém, o método proposto neste trabalho se torna viável por causa de sua simples implementação em conjunto com controladores preditivos, e a não existência de inversão de matrizes no processo.

4.3 Controlador CDMPC com Restrições

Para o controlador *CDMPC*, o problema de otimização com restrições é baseado em encontrar o vetor,

$$\Delta U_{restr} = \begin{bmatrix} (\Delta u_1(k) \Delta u_2(k) \cdots \Delta u_i(k))^T \\ (\Delta u_1(k+1) \Delta u_2(k+1) \cdots \Delta u_i(k+1))^T \\ \vdots \\ (\Delta u_1(k+N_c-1) \Delta u_2(k+N_c-1) \cdots \Delta u_i(k+N_c-1))^T \end{bmatrix} \quad (4.14)$$

que minimize,

$$\bar{J} = \frac{1}{2} \Delta U_{restr}^T \overbrace{(\Phi^T \Phi + \bar{R})}^E \Delta U_{restr} + \overbrace{(\Phi^T R_s - \Phi^T F x_a(k))}^{c^T} \Delta U_{restr} \quad (4.15)$$

estando sujeita às restrições dadas por,

$$\begin{bmatrix} T_{\Delta U} \\ -T_{\Delta U} \\ T_U \\ -T_U \end{bmatrix} \stackrel{\gamma}{\Delta U_{restr}} \leq \begin{bmatrix} V_{\Delta U_{max}} \\ -V_{\Delta U_{min}} \\ V_{U_{min}} \\ -V_{U_{min}} \end{bmatrix} \quad (4.16)$$

onde,

$$T_{\Delta U} = \begin{bmatrix} (\dot{I}_1 & \dot{I}_2 & \cdots & \dot{I}_i)_1 \\ (\dot{I}_1 & \dot{I}_2 & \cdots & \dot{I}_i)_2 \\ \vdots \\ (\dot{I}_1 & \dot{I}_2 & \cdots & \dot{I}_i)_{N_c} \end{bmatrix}; T_U = \begin{bmatrix} (C_{I_1} & C_{I_2} & \cdots & C_{I_i})_1 \\ (C_{I_1} & C_{I_2} & \cdots & C_{I_i})_2 \\ \vdots \\ (C_{I_1} & C_{I_2} & \cdots & C_{I_i})_{N_c} \end{bmatrix} \quad (4.17)$$

$$V_{\Delta U_{max}} = \begin{bmatrix} (\Delta U^{max})_1 \\ (\Delta U^{max})_2 \\ \vdots \\ (\Delta U^{max})_{N_c} \end{bmatrix}; V_{U_{max}} = \begin{bmatrix} (U^{max} - U^{k-1})_1 \\ (U^{max} - U^{k-1})_2 \\ \vdots \\ (U^{max} - U^{k-1})_{N_c} \end{bmatrix} \quad (4.18)$$

$$V_{\Delta U_{min}} = \begin{bmatrix} (\Delta U^{min})_1 \\ (\Delta U^{min})_2 \\ \vdots \\ (\Delta U^{min})_{N_c} \end{bmatrix}; V_{U_{min}} = \begin{bmatrix} (U^{min} - U^{k-1})_1 \\ (U^{min} - U^{k-1})_2 \\ \vdots \\ (U^{min} - U^{k-1})_{N_c} \end{bmatrix} \quad (4.19)$$

Sendo C_I uma matriz triangular inferior unitária e \dot{I} uma matriz diagonal unitária, ambas com dimensões $N_c \times N_c$, $i = 1, 2, \dots, n$. A matriz M possui dimensão $4nN_c \times nN_c$ e os vetores γ e ΔU possuem dimensões $4nN_c \times 1$ e $nN_c \times 1$. $U^{k-1} = \begin{bmatrix} u_1(k-1) & u_2(k-1) & \cdots & u_i(k-1) \end{bmatrix}^T$ é o vetor com as amostras passadas de $u_i(k-1)$ para cada respectiva entrada n .

Os valores de restrição de amplitude máxima para o sinal incremental e sinal de controle são representados respectivamente pelos vetores

$$\begin{aligned}\Delta U^{\max} &= \begin{bmatrix} \Delta u_1^{\max} & \Delta u_2^{\max} & \dots & \Delta u_i^{\max} \end{bmatrix}^T \\ U^{\max} &= \begin{bmatrix} u_1^{\max} & u_2^{\max} & \dots & u_i^{\max} \end{bmatrix}^T\end{aligned}\quad (4.20)$$

Similarmente, para os valores de amplitude mínima,

$$\begin{aligned}\Delta U^{\min} &= \begin{bmatrix} \Delta u_1^{\min} & \Delta u_2^{\min} & \dots & \Delta u_i^{\min} \end{bmatrix}^T \\ U^{\min} &= \begin{bmatrix} u_1^{\min} & u_2^{\min} & \dots & u_i^{\min} \end{bmatrix}^T\end{aligned}\quad (4.21)$$

Utilizando a programação quadrática de Hildreth demonstrada na Seção 4.2.2, é possível obter os valores ótimos para ΔU_{restr} que solucionam o problema de otimização da lei de controle com restrição. Então a lei de controle com horizonte retrocedente com restrição para o controlador CDMPC é dada por:

$$\Delta u(k)_{restr} = -v((\Phi^T \Phi + \bar{R})^{-1}((\Phi^T R_s - \Phi^T F x_a(k)) + M^T \lambda_{calc})) \quad (4.22)$$

Onde para facilitar a nomenclatura, será chamado de HCDMPC (Hildreth Classic Discrete-Time Model Predictive Control).

4.4 Controlador LDMPC com Restrições

No controlador LDMPC com restrição, o problema de otimização está relacionado em encontrar η_{restr} que minimize a função custo,

$$\bar{J} = \frac{1}{2} \eta_{restr}^T \overbrace{\Omega}^E \eta_{restr} + \overbrace{(\Psi x_a(k))^T}^{c^T} \eta_{restr} \quad (4.23)$$

sujeita às restrições,

$$\begin{bmatrix} M_{\Delta U} \\ -M_{\Delta U} \\ M_U \\ -M_U \end{bmatrix} \overbrace{\eta_{restr}}^v \leq \begin{bmatrix} V_{\Delta U_{\max}} \\ -V_{\Delta U_{\min}} \\ V_{U_{\max}} \\ -V_{U_{\min}} \end{bmatrix} \quad (4.24)$$

onde,

$$M_{\Delta U} = \begin{bmatrix} D(1) \\ D(2) \\ \vdots \\ D(N_{Nsum}) \end{bmatrix} \quad (4.25)$$

$$M_U = \begin{bmatrix} D(1) \\ D(1) + D(2) \\ \vdots \\ D(N_{Nsum} - 1) + D(N_{Nsum}) \end{bmatrix} \quad (4.26)$$

sendo D uma matriz composta por,

$$D(k) = \begin{bmatrix} (A_{l_1}^{k-1} L_1(0))^T & \sigma_2^T & \cdots & \sigma_i^T \\ \sigma_1^T & (A_{l_2}^{k-1} L_2(0))^T & \cdots & \sigma_i^T \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^T & \sigma_2^T & \cdots & (A_{l_i}^{k-1} L_i(0))^T \end{bmatrix} \quad (4.27)$$

$k = 1, 2, 3, \dots, N_{sum}$ e σ_i^T um vetor de zeros com dimensão N_i , $i = 1, 2, \dots, n$. $N_{sum} = N_1 + N_2 + \dots + N_i$ é a soma de todas as ordens das Funções de Laguerre. Na Equação 4.24 a matriz M tem dimensão $4nN_{sum} \times nN_{sum}$, os vetores γ e η_{restr} têm dimensões $4nN_{sum} \times 1$ e $nN_{sum} \times 1$, respectivamente.

Utilizando a programação quadrática de Hildreth pode-se calcular λ_{calc} e assim encontrar o vetor η_{restr} ,

$$\eta_{restr} = -\Omega^{-1}(\Psi_{x_a}(k) + M^T \lambda_{calc}) \quad (4.28)$$

que minimize 4.23 estando sujeito às restrições de 4.24.

Então, a lei de controle de horizonte retrocedente com restrição para o controlador $LDMPC$ é dada por:

$$\Delta u(k)_{restr} = \begin{bmatrix} L_1(0)^T & \sigma_2^T & \cdots & \sigma_i^T \\ \sigma_1^T & L_2(0)^T & \cdots & \sigma_i^T \\ \vdots & \vdots & \ddots & \vdots \\ \sigma_1^T & \sigma_2^T & \cdots & L_i(0)^T \end{bmatrix} \eta_{restr} \quad (4.29)$$

Similarmente ao projeto anterior, esse projeto de controlador será denominado como $HLDMPCC$ (Hildreth Laguerre Discrete-Time Model Predictive Control).

4.5 Simulações com restrições

Neste tópico será realizada simulações *com restrições* com o objetivo de demonstrar a eficiência do método proposto. As simulações foram realizadas utilizando $G_{11}(z)$ em 5.2, os parâmetros de sintonia são: $N_p = 50$; $\Gamma = 0,0001$; $N_c = N = 2$ e $a = 0,3$ e $Q = C_a^T C_a$, sendo elas realizadas para dois casos distintos: com restrições usando saturação e com restrições usando a programação quadrática de Hildreth. As restrições impostas ao sistema foram $-6 \leq u(k) \leq +6$ e $-1 \leq \Delta u(k) \leq +1$.

Para facilitar a nomenclatura, o controlador $LDMPC$ com uma simples saturação no sinal de controle será denominado como $SLDMPC$ (Saturated Laguerre Discrete-Time Model

Predictive Control) e similarmente, o controlador *CDMPC* como *SCDMPC* (Saturated Classic Discrete-Time Model Predictive Control).

Nas Figuras 16, 17 e 18 são apresentados os resultados da saída do sistema, do sinal de controle e do incremento do sinal de controle, respectivamente. Na Tab. 2 são apresentados os resultados dos índices de desempenhos de cada controlador.

Analisando as simulações e a tabela de índices, verifica-se que *HLDMPC* apresentou o melhor desempenho quando sujeito a restrições no sinal de controle, mostrando o melhor equilíbrio entre a viabilidade do sinal de controle e o desempenho de malha fechada do sistema, o qual obteve o menor ST com menor gasto energético ISU. No entanto, *SLDMPC* apresentou o menor IAE mas com um PO elevado, sendo isso extremamente indesejado para servomecanismos. Já para *HCDMPC*, seu desempenho foi satisfatória, sendo superior a *SCDMPC* que apresentou uma dinâmica de sistema subamortecida com um elevado ST e IAE.

A programação quadrática de Hildreth demonstrou grande vantagem, que está relacionada diretamente ao incremento do sinal de controle. Na Figura 18 verifica-se que *HLDMPC* e *HCDMPC* apresentam os menores *TVC*. A restrição de variação do incremento de controle é de extrema importância, principalmente quando é necessário um controle apurado de posição, velocidade e aceleração de servomecanismos. Portanto, esse método se mostrou bastante eficaz, principalmente em conjunto com *LDMPC*, reduzindo consideravelmente o problema do conflito entre a viabilidade do sinal de controle e o desempenho em malha fechada para sistemas com restrições nas variáveis de entrada.

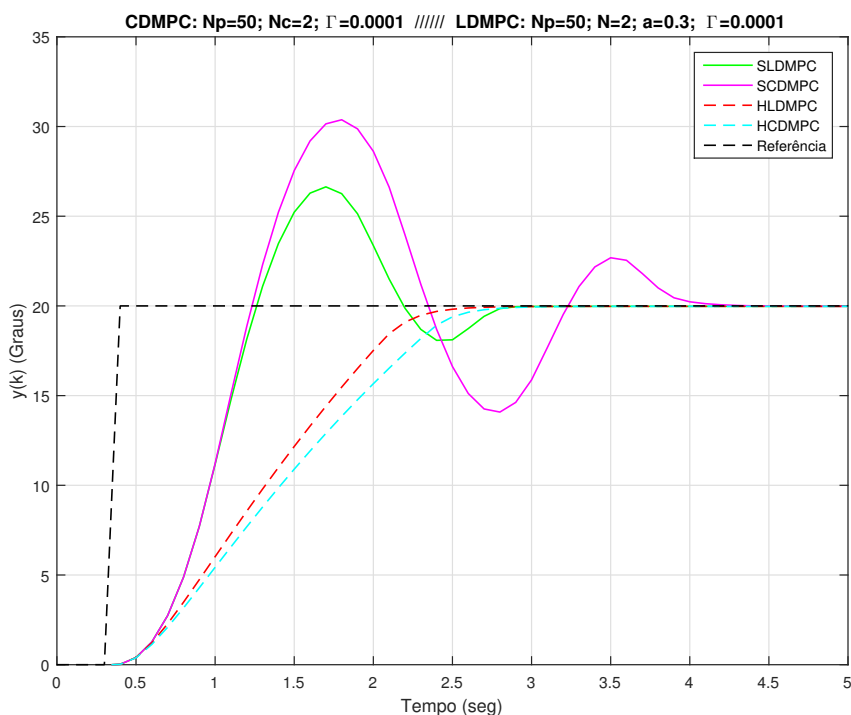


Figura 16 – Saída simulada do sistema: com restrição.

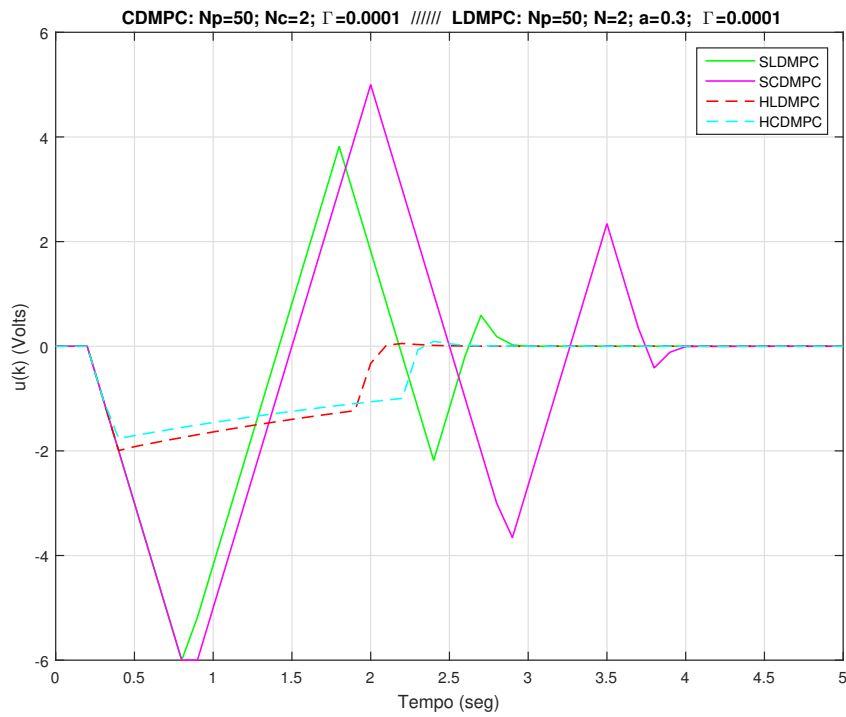


Figura 17 – Sinal de Controle: com restrição.

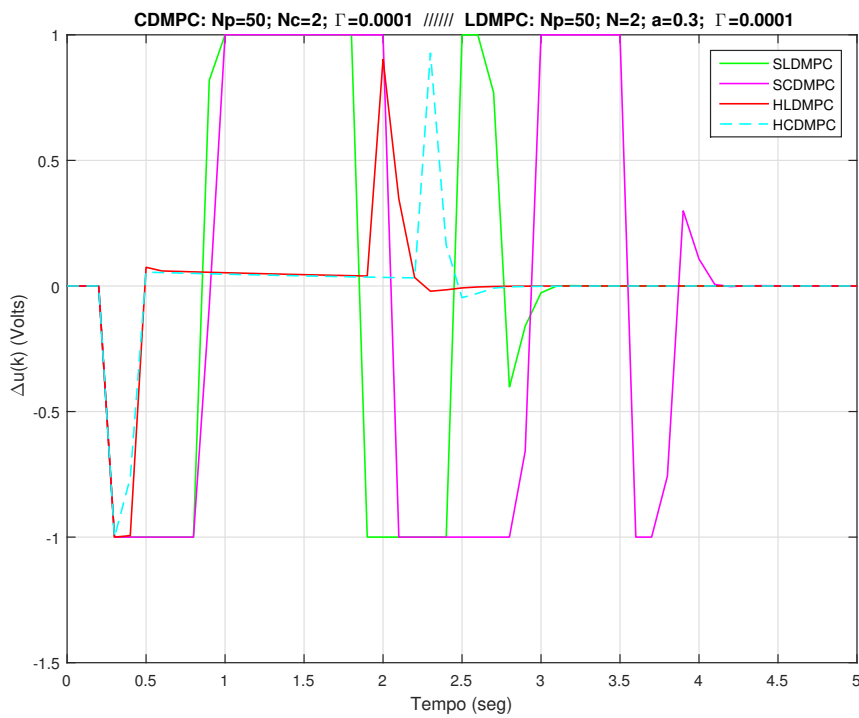


Figura 18 – Sinal de Controle incremental: com restrição.

Tabela 2 – Índices de desempenho: com restrição.

	ISU	IAE	PO(%)	ST(seg)	TVC
SLDMPC	57,527057	2,560631	33,345990	3,0	25,180020
SCDMPC	87,889601	3,705753	52,070301	4,5	34,914404
HLDMPC	26,743828	3,087789	0	2,9	4,093559
HCDMPC	26,819363	3,421157	0	3,2	3,713820

4.6 Conclusões

Neste capítulo, foi discutido o problema de otimização do sinal de controle com restrição para controladores preditivos, com o objetivo de apresentar métodos capazes de ajudar a reduzir o conflito entre a viabilidade do sinal de controle e o desempenho em malha fechada para sistemas com restrições impostas às variáveis de entrada ($\Delta u(k)$, $u(k)$).

Foram apresentados os fundamentos da programação quadrática, levando em consideração a minimização da função custo (\bar{J}) sujeita a restrições que são representadas por um conjunto de inequações matriciais ($Mv \leq \gamma$), caracterizando-se como um problema primal-dual. Para resolver esse problema de otimização com restrições é utilizada a programação quadrática de Hildreth, que resolve problemas do tipo dual, o qual a cada instante de tempo k são identificadas as restrições não ativas via multiplicadores de Lagrange ($\lambda < 0$). Assim através de um processo de interações elemento a elemento são obtidos os valores ótimos do vetor ($\lambda \geq 0$), o que a torna um problema do tipo primal, fazendo com que a solução ótima para a variável de decisão (v) seja uma igualdade de restrições ($Mv = \gamma$). Além disso, algo de extrema importância para a escolha desse algoritmo é a não necessidade de inversão de matrizes, portanto, para casos onde ocorre conflitos de restrições, a solução numérica não torna o sistema instável.

Foram também apresentadas as leis de controle de horizonte retrocedente com restrições para os controladores *LDMPC* e *CDMPC*, onde foi usando o algoritmo de Hildreth em conjunto com os controladores preditivos para calcular o controle ótimo com restrições (ΔU_{restr}).

Por fim, foram realizadas simulações com restrições no sinal de controle, tendo o objetivo de demonstrar a eficiência da programação quadrática de Hildreth para reduzir o conflito mencionado em controladores preditivos com restrições nas variáveis de entrada. Este método apresentou excelentes resultados, principalmente quando utilizado em conjunto com *HLDMPC*. Isto porque as Funções de Laguerre geram um desempenho aceitável, sem a necessidade de um longo horizonte de controle e de predição, tornando mais simples a solução das inequações matriciais ($Mv \leq \gamma$). Além disso, esse algoritmo gerou uma diminuição considerável da variação do sinal de controle, sendo isso uma de suas principais contribuições.

No Apêndice D, é demonstrado no código 11, o algoritmo da programação quadrática de Hildreth desenvolvido no *MATLAB*®.

Como trabalhos correlatos que discutem o mesmo assunto deste capítulo pode-se citar,

Chai, Wang e Rogers (2011a), Huyck *et al.* (2013), Mateescu, Stefanoiu e Christov (2013) e Yang, Zhang e Liu (2013).

TESTES EXPERIMENTAIS

5.1 Introdução

Nos capítulos anteriores foram discutidas e evidenciadas as vantagens das funções ortonormais de Laguerre para obter um desempenho satisfatório sem a necessidade de um longo horizonte de predição e de controle, além de também ter sido demonstrado os seus benefícios em conjunto com a programação quadrática de Hildreth, para reduzir o conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada em sistemas com restrições.

Neste capítulo, serão discutidos e apresentados os resultados obtidos nas simulações e experimentos com os controladores *LDMPC* e *CDMPC* para os casos *MIMO* com e sem restrições, sendo primeiramente apresentados os experimentos com o manipulador robótico Mentor, e posteriormente as simulações com a planta benchmark chamada de coluna de destilação binária de Wood e Berry (1973). Os resultados são comparados com a finalidade de evidenciar as vantagens do controlador preditivo usando as funções de Laguerre em relação ao método clássico que utiliza pulso de avanço de tempo discreto.

5.2 Manipulador Robótico Mentor

O braço robótico utilizado nos testes experimentais, é um robô manipulador Mentor Desktop robot, da Feedback Instruments Limited, apresentado na Figura 19, com 5 DOF (Degree of Freedom), porém, nesta experiência serão utilizados apenas 3 DOF, com o objetivo de simplificar os testes e a apresentação dos resultados obtidos. Este manipulador possui juntas revolutas, sendo cada articulação acionada por um motor de corrente contínua com potenciômetros como sensores de posição angular que realizam a realimentação da malha de controle. O circuito para o acionamento dos motores e para a conversão de sinal analógico digital e digital analógico (A-D/D-A), não são os originais do robô Mentor Desktop. No entanto, foi desenvolvido um novo circuito de acionamento e aquisição de dados apresentado na Figura 20 e detalhado no Apêndice

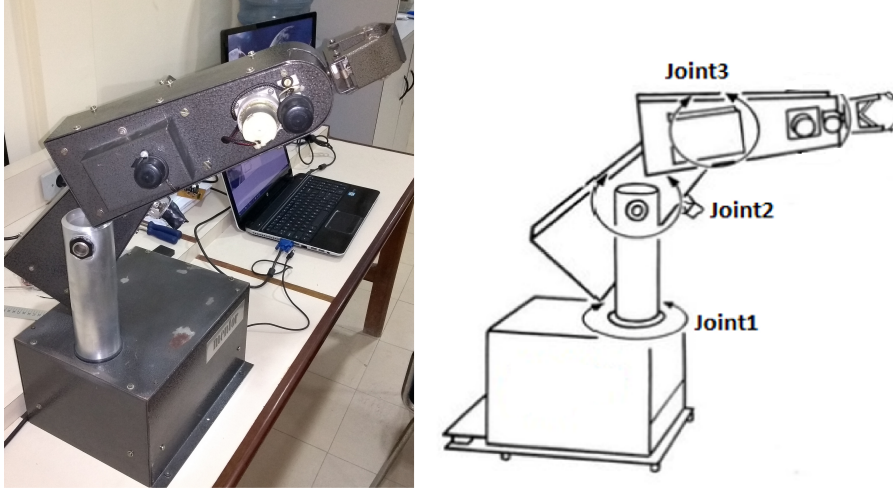


Figura 19 – Manipulador robótico e diagrama do Mentor Desktop robot.

A, onde utiliza um microcontrolador Arduino Micro para realizar a comunicação (A-D/D-A) com o computador.

O sistema MIMO que representa o braço robótico Mentor com 3 DOF foi obtido através de uma modelagem experimental usando o estimador de mínimos quadrados recursivo, que pode ser representado pela matriz de funções de transferência com 3 entradas e 3 saídas dada por:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \\ y_3(k) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) & G_{13}(z) \\ G_{21}(z) & G_{22}(z) & G_{23}(z) \\ G_{31}(z) & G_{32}(z) & G_{33}(z) \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \\ u_3(k) \end{bmatrix} \quad (5.1)$$

onde,

$$\begin{aligned} G_{11}(z) &= \frac{-0,0007272z-0,003917}{z^2-1,604073z+0,604073}; \\ G_{12}(z) &= \frac{-5,2074e-9z-3,4674e-11}{z^2-0,28848z-0,70734}; \\ G_{13}(z) &= \frac{5,6358e-09z-5,5471e-09}{z^2-0,5z-0,5}; \\ G_{21}(z) &= \frac{-1,1382e-9z-7,6199e-12}{z^2-0,49889z-0,498254}; \\ G_{22}(z) &= \frac{0,0002373z-0,005518}{z^2-1,534036z+0,534036}; \\ G_{33}(z) &= \frac{-0,002278z-0,01048}{z^2-1,539275z+0,539275}; \end{aligned} \quad (5.2)$$

$$G_{13}(z) = G_{23}(z); G_{31}(z) = G_{21}(z); G_{12}(z) = G_{32}(z);$$

tendo o período de amostragem $T_s = 0,1 \text{ seg}$. As entradas, $u(k)$, são dadas em volts e as saídas, $y(k)$, são as posições angulares das juntas em graus.

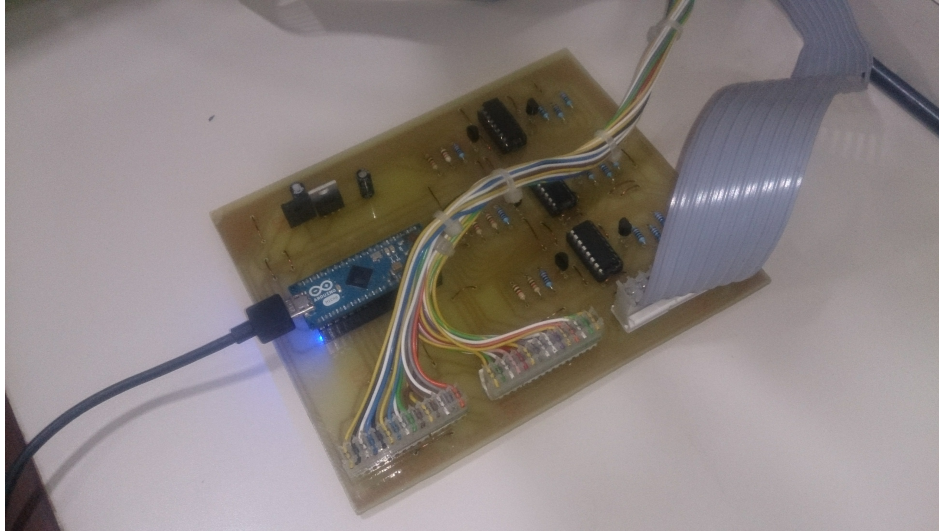


Figura 20 – Circuito de acionamento e aquisição de dados.

5.2.1 Simulações com o manipulador Mentor

Neste tópico são apresentadas e discutidas as simulações para os casos com e sem restrições no sinal de controle para o sistema MIMO apresentado em 5.1. Tendo o intuito de comparar o desempenho dos controladores *LDMPC*, *CDMPC*, *SLDMPC*, *SCDMPC*, *HLDMP* e *HCDMPC*.

As simulações dos testes de rastreamento de referência para os controladores foram realizadas utilizando os parâmetros de sintonia $N_p = 100$, $N_1 = N_2 = N_3 = N_c = 2$, $a_1 = a_2 = a_3 = 0,3$, $\Gamma = 0,008$ e $Q = C_a^T C_a$. As restrições impostas a todas as entradas do sistema são,

$$-5 \text{ volts} \leq \begin{bmatrix} u_1(k) & u_2(k) & u_3(k) \end{bmatrix} \leq +5 \text{ volts} \quad (5.3)$$

$$-2.5 \text{ volts} \leq \begin{bmatrix} \Delta u_1(k) & \Delta u_2(k) & \Delta u_3(k) \end{bmatrix} \leq +2.5 \text{ volts} \quad (5.4)$$

e as referências para cada saída são: $junta_1 = 180^\circ$, $junta_2 = 90^\circ$ e $junta_3 = 60^\circ$.

Nas Figuras 21, 22 e 23 são mostrados os resultados das simulações para o caso *sem restrições* no sinal de controle e na Tab. 3 são apresentados os seus respectivos índices de desempenho. Em seguida, nas Figuras 25, 26, 27 são apresentados os resultados das simulações para o caso *com restrições* no sinal de controle e na Tab. 4 são apresentados os índices de desempenho de cada controlador.

Analisando os resultados das simulações para o caso *sem restrições*, é possível afirmar que para os mesmos parâmetros de sintonia, *LDMPC* apresentou o melhor desempenho, porém, com maiores índices *TVC*, *ISU* e *PO*. Portanto, assim como foi evidenciado no caso SISO, as Funções de Laguerre também tornam o horizonte de controle mais efetivo para sistemas MIMO.

Já examinando os resultados para o caso *com restrições*, *SLDMPC* e *HLDMP* apresentaram os melhores desempenhos para $junta_1$ e $junta_2$, reduzindo de forma mais efetiva o

conflito entre a viabilidade do sinal de controle e desempenho de malha fechada em sistemas com restrições nas variáveis de entrada. Apesar desses dois controladores terem desempenhos similares, *HLDMP*C, que utiliza a programação quadrática de Hildreth, obteve uma redução da variação do sinal de controle incremental como é evidenciado através do índice *TVC*. Essa redução também pode ser observada em *HCDMP*C. Já para a *Junta*₃, *SCDMP*C apresentou os melhores índices *ST*, *IAE*, *ISE*, entretanto, com maiores *TVC*, *ISU* e *PO*.

Sem dúvida uma das maiores contribuições da programação quadrática de Hildreth é o fato de diminuir variação de controle incremental. Um controle adequado da variação de $\Delta u(k)$ é de suma importância para um controle preciso de posição e velocidade angular de servomecanismos. No entanto, apesar da programação quadrática de Hildreth aumentar o custo computacional por conta das várias interações realizadas a cada instante de tempo k , o método mostrou-se satisfatório quando utilizado em conjunto com as Funções de Laguerre. Isso pelo fato das funções ortonormais propiciarem um desempenho de malha fechada eficaz para um pequeno número de parâmetros N , diminuindo drasticamente a complexidade da solução das inequações matriciais ($Mx \leq \gamma$) que representam as restrições impostas ao sistema.

Na Figura 24 são mostradas as posições dos 17 polos de malha fechada do Mentor para *LDMPC* e *CDMPC*, obtidas através das Equações 3.33 e 2.28, respectivamente.

Tabela 3 – Índices de desempenho das simulações do Mentor: sem restrição.

		CDMPC	LDMPC
ISU	Junta 1	2,440544e+02	2,638873e+02
	Junta 2	1,379473e+02	1,499118e+02
	Junta 3	44,522334	43,238398
ISE	Junta 1	32,512304	26,841002
	Junta 2	10,069779	8,296901
	Junta 3	3,070524	3,062231
IAE	Junta 1	15,637420	12,548278
	Junta 2	8,644956	6,855677
	Junta 3	3,657212	3,679995
PO (%)	Junta 1	0,234785	0,256754
	Junta 2	0,302623	1,386270
	Junta 3	0	0
TVC	Junta 1	1,074136e+02	1,580382e+02
	Junta 2	58,341400	88,386394
	Junta 3	46,087244	46,563872
ST(seg)	Junta 1	1,92	1,44
	Junta 2	1,8	1,32
	Junta 3	1,2	1,2

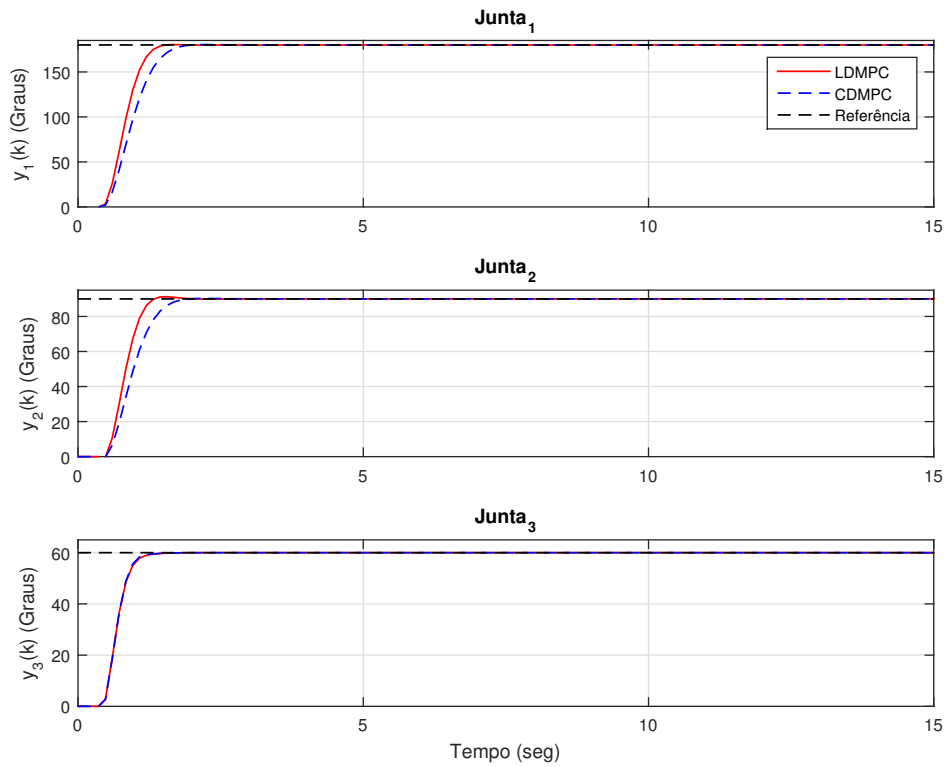


Figura 21 – Simulações das saídas do Mentor: sem restrição.

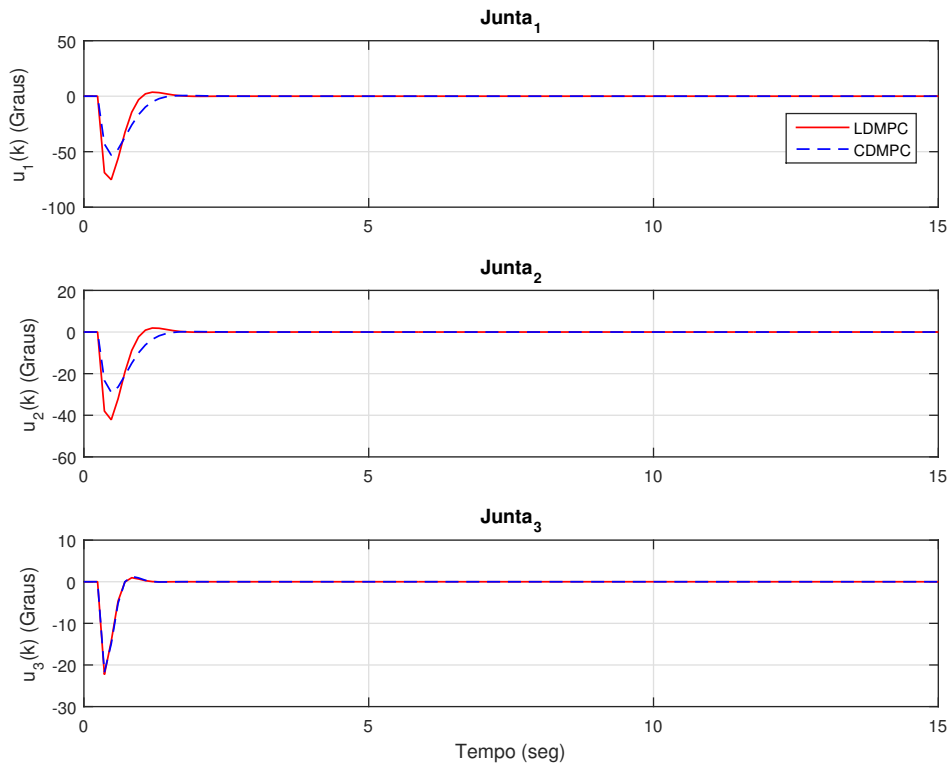


Figura 22 – Simulações dos sinais de controle do Mentor: sem restrição.

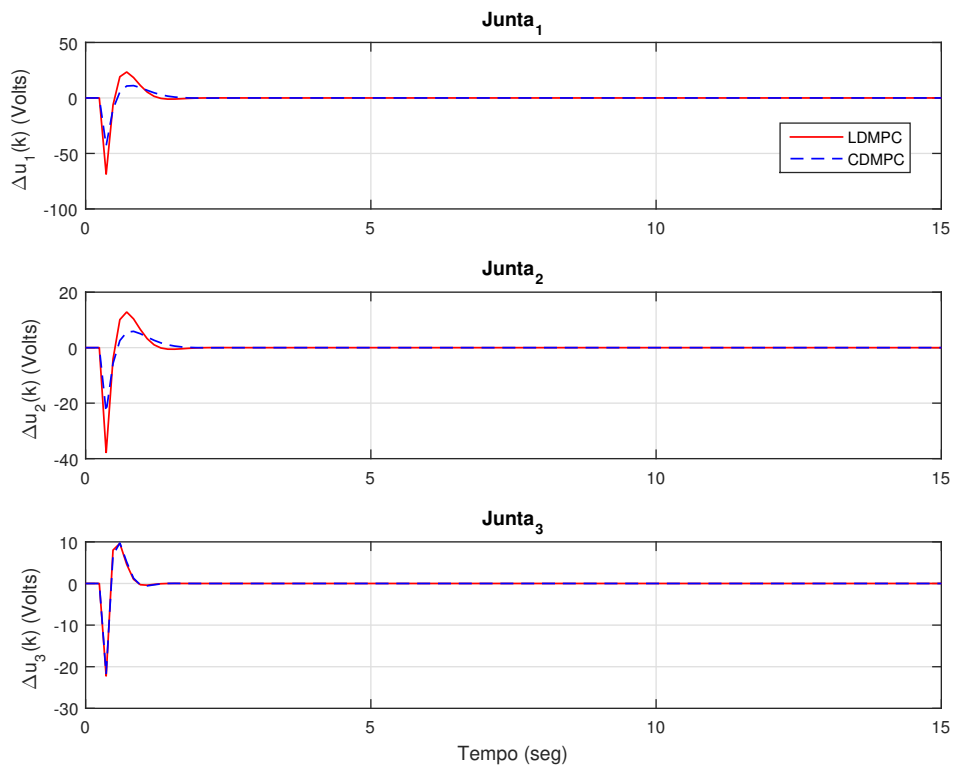


Figura 23 – Simulações dos sinais de controle incremental do Mentor: sem restrição.

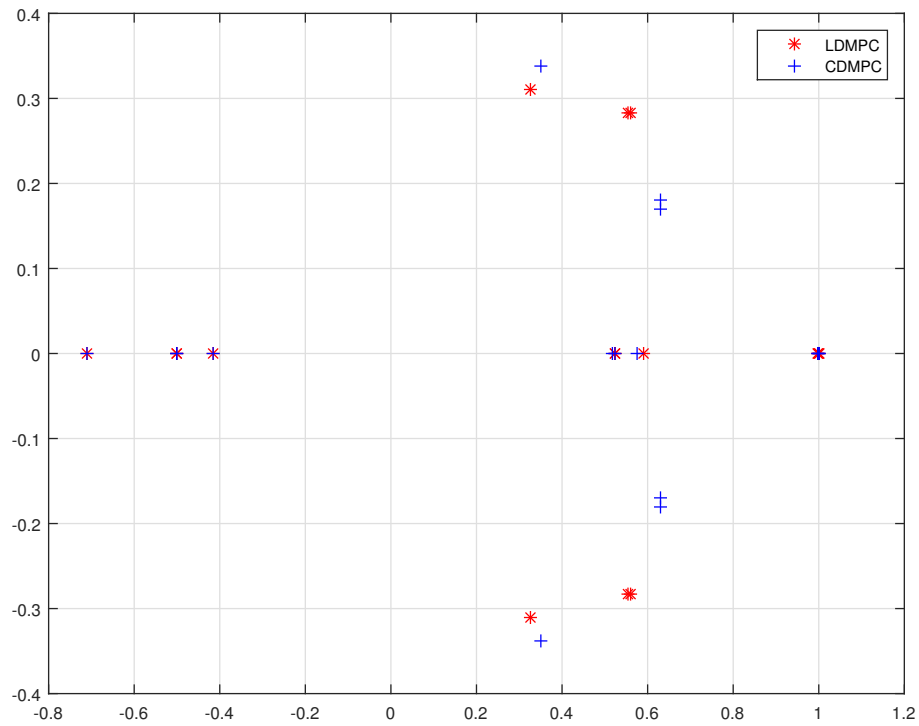


Figura 24 – Polos de malha fechada do Mentor.

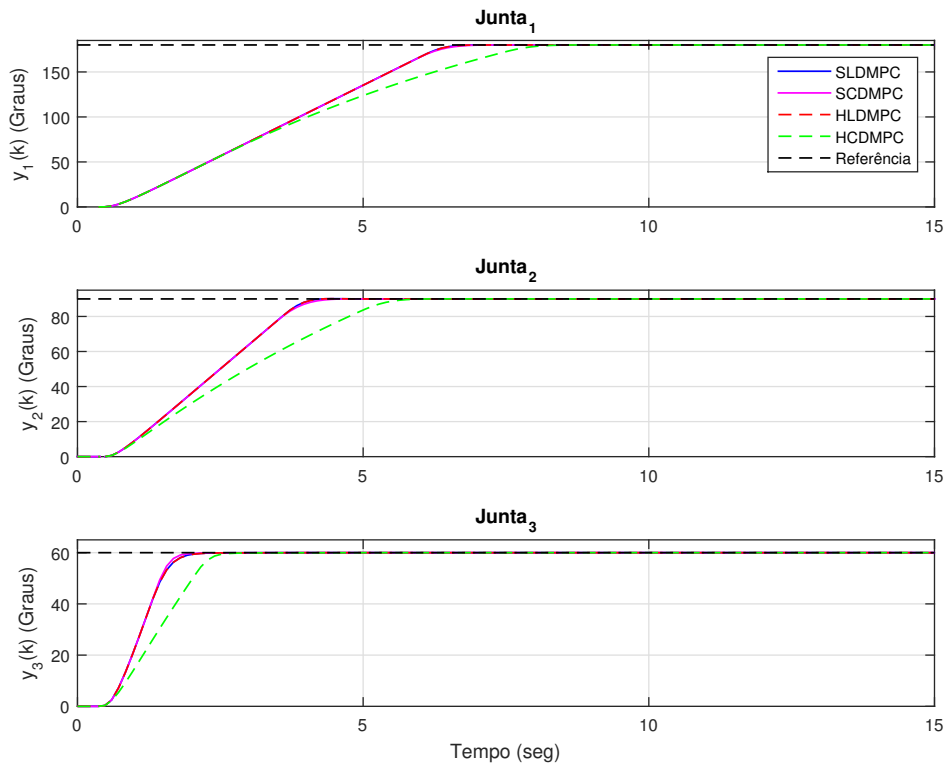


Figura 25 – Simulações das saídas do Mentor: com restrição.

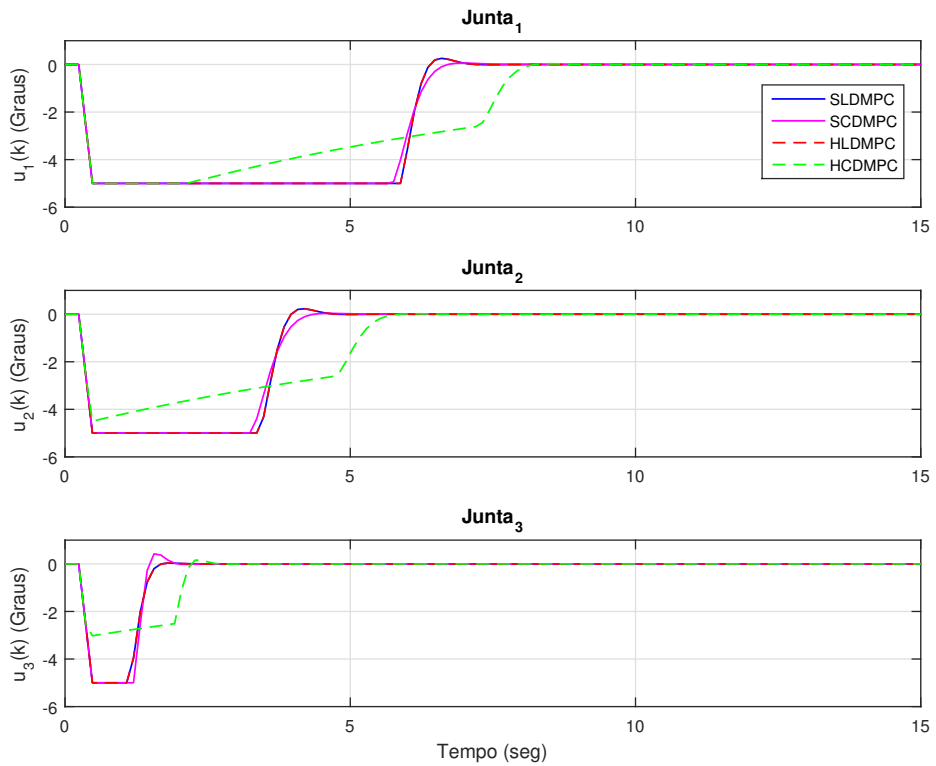


Figura 26 – Simulações dos sinais de controle do Mentor: com restrição.

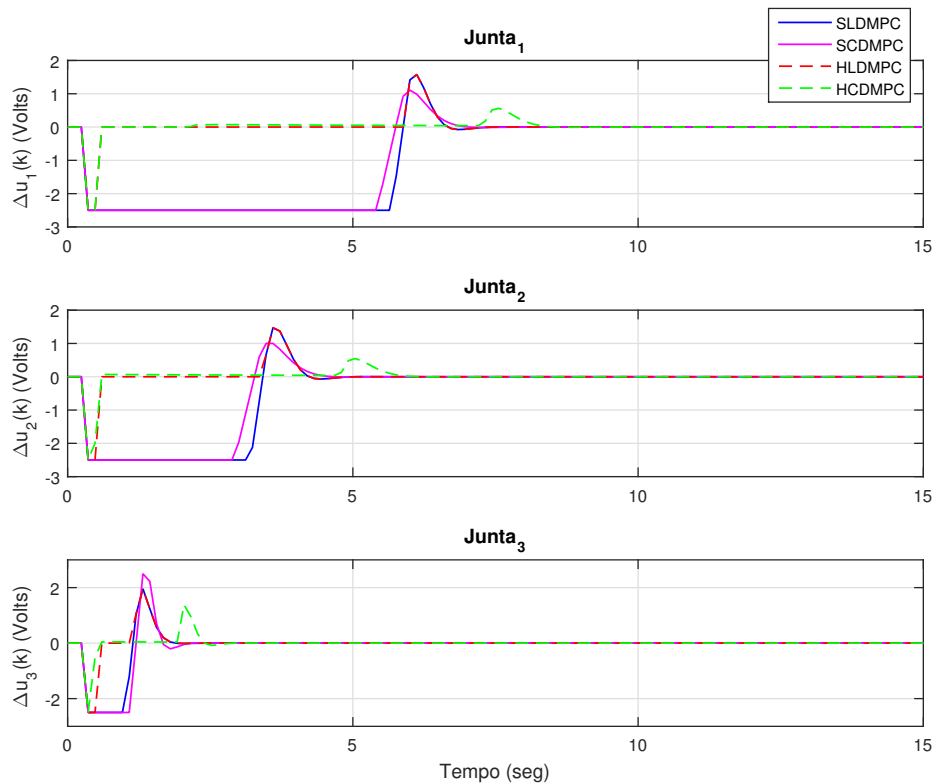


Figura 27 – Simulações dos sinais de controle incremental do Mentor: com restrição.

Tabela 4 – Índices de desempenho das simulações do Mentor: com restrição.

		SLDMPC	SCDMPC	HLDMPC	HCDMPC
ISU	Junta 1	2,399742e+02	2,387543e+02	2,399742e+02	2,384717e+02
	Junta 2	1,375421e+02	1,363158e+02	1,375421e+02	1,361500e+02
	Junta 3	39,556919	41,354962	39,534738	40,182890
ISE	Junta 1	1,4954778e+02	1,495566e+02	1,495477e+02	1,560965e+02
	Junta 2	28,850951	28,859624	28,850916	34,130755
	Junta 3	6,358108	6,340345	6,358633	8,310958
IAE	Junta 1	75,919637	76,006817	75,919636	83,18614
	Junta 2	25,857356	25,947798	25,857334	32,389416
	Junta 3	7,611112	7,510603	7,622212	10,380641
PO (%)	Junta 1	0,003634	0,023286	0,003634	0,011853
	Junta 2	0,149916	0,053343	0,149928	0,027619
	Junta 3	0	0	0	0
TVC	Junta 1	1,195431e+02	1,152020e+02	10,537263	10,062989
	Junta 2	68,328570	63,403718	10,478664	9,034058
	Junta 3	21,284118	23,405242	10,076145	6,381624
ST(seg)	Junta 1	6,60	6,72	6,60	8,16
	Junta 2	4,08	4,20	4,08	5,64
	Junta 3	1,92	1,80	1,92	2,5

5.2.2 Testes experimentais com o manipulador Mentor

Aqui são apresentados os resultados dos testes experimentais com o manipulador robótico Mentor, tendo como objetivo comprovar a eficiência do método proposto para um processo real e ainda confirmar os resultados obtidos anteriormente nas simulações.

Os resultados dos testes experimentais são apresentados nas Figuras 28, 29, 30 e na Tab. 5 são mostrados os seus respectivos índices de desempenho. Os parâmetros de sintonia e as restrições impostas ao sistema são os mesmos utilizados nas simulações realizadas anteriormente.

De posse dos índices da Tab. 5, é possível afirmar que *SLDMPC* e *HLDMP* apresentaram os melhores desempenhos de malha fechada, no entanto, *SLDMPC*, que utiliza uma simples saturação, demonstrou uma pequena superioridade na *junta*₁ para os índices *ISE*, *IAE*, *ISU* e *ST*, perdendo apenas nos índices *TVC* e *PO*. Já para *junta*₂ e *junta*₃, *HLDMP*, que utiliza a programação quadrática de Hildreth, apresentou o melhor desempenho em relação a todos os outros controladores, principalmente em relação ao índice *TVC*, que reduziu consideravelmente.

Para analisar esses resultados, é necessário considerar as não linearidades do Mentor, as perturbações e os ruídos naturais dos motores, além dos desgastes estruturais ocorridos após mais de 20 anos de utilização, portanto, apresentando certos limites de precisão. Tendo em vista isso, pode-se considerar que os testes experimentais apresentaram resultados satisfatórios, sendo equivalentes aos obtidos nas simulações. Assim confirmando a eficácia do controlador utilizando as Funções de Laguerre em conjunto com a programação quadrática de Hildreth em um experimento real, os quais conseguiram uma melhor redução do conflito ocasionado pelas restrições impostas ao sinal de controle, resultando em um melhor equilíbrio entre a viabilidade do sinal de controle e o desempenho de malha fechada do sistema.

As principais dificuldades encontradas nos testes experimentais com o Mentor foram:

- Problemas de zona morta apresentada pelos motores das juntas robóticas, afetando diretamente a precisão do controle de posição angular, principalmente quando necessário realizar pequenas mudanças de posição.
- A presença de um nível elevado de ruídos na planta, ocasionando em oscilações que podem ser verificadas nas Figuras 29 e 30, afetando diretamente o desempenho do controlador.
- Desgastes nas engrenagens das caixas de redução, provocando problemas estruturais relacionados a folgas (*backlash*), afetando a precisão do controle da posição angular das juntas.

É necessário ressaltar que a planta do Mentor é do tipo-1, ou seja, uma planta integradora com polos localizados em 1 no círculo unitário. Assim, pode-se afirmar que os controladores preditivos apresentados nessa dissertação são eficientes para esse tipo de sistema.

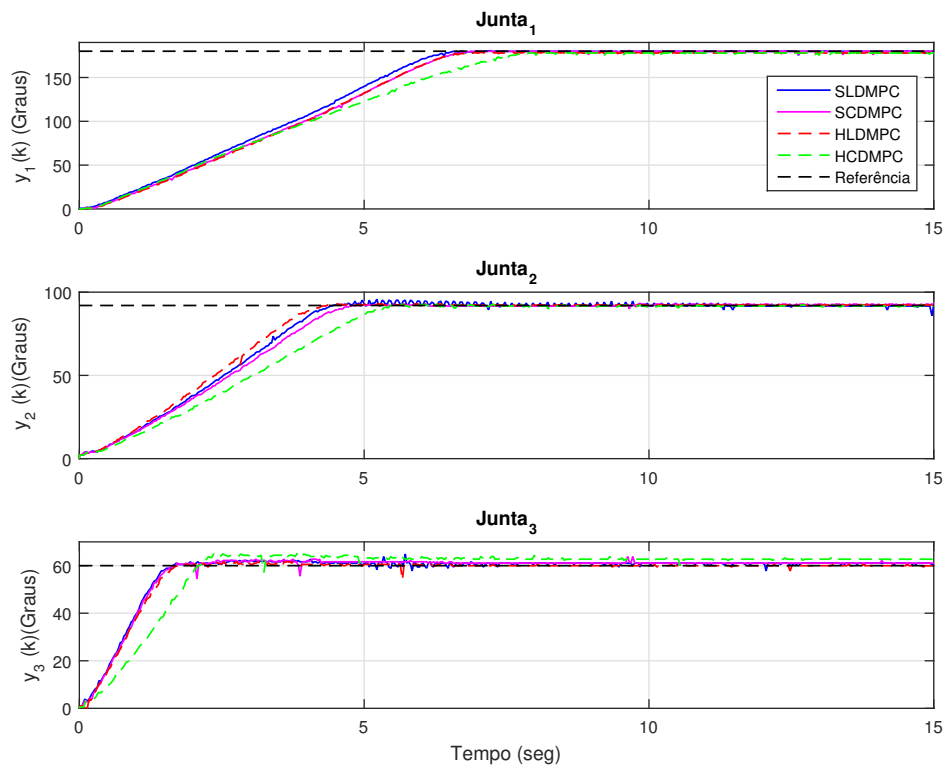


Figura 28 – Sinal de saída dos testes experimentais do Mentor: com restrição.

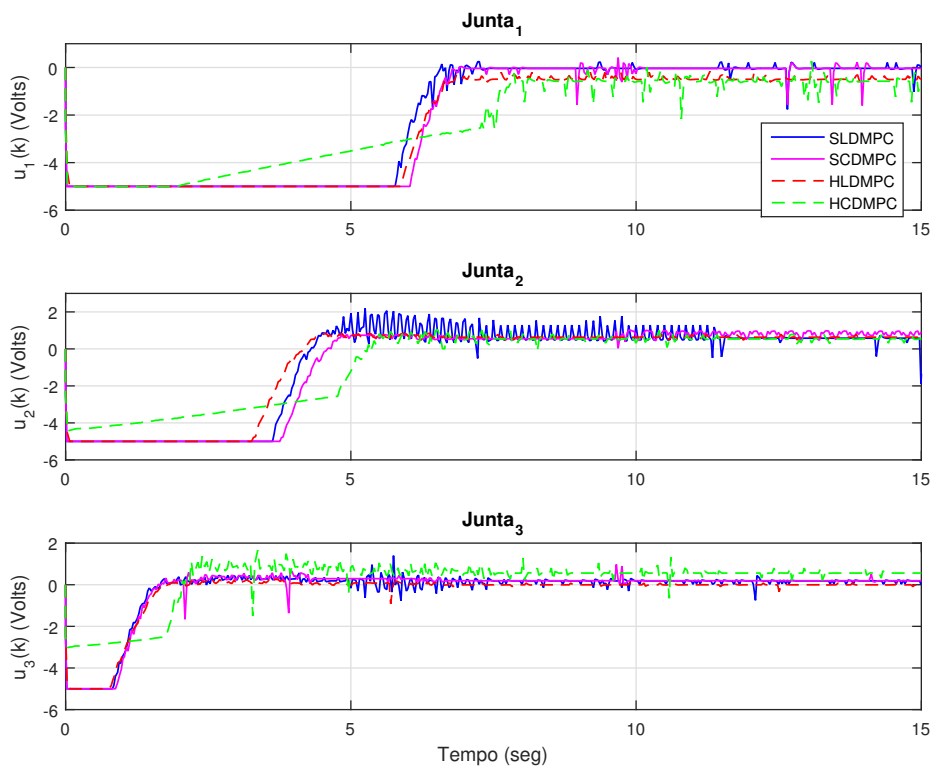


Figura 29 – Sinal de controle dos testes experimentais do Mentor: com restrição.

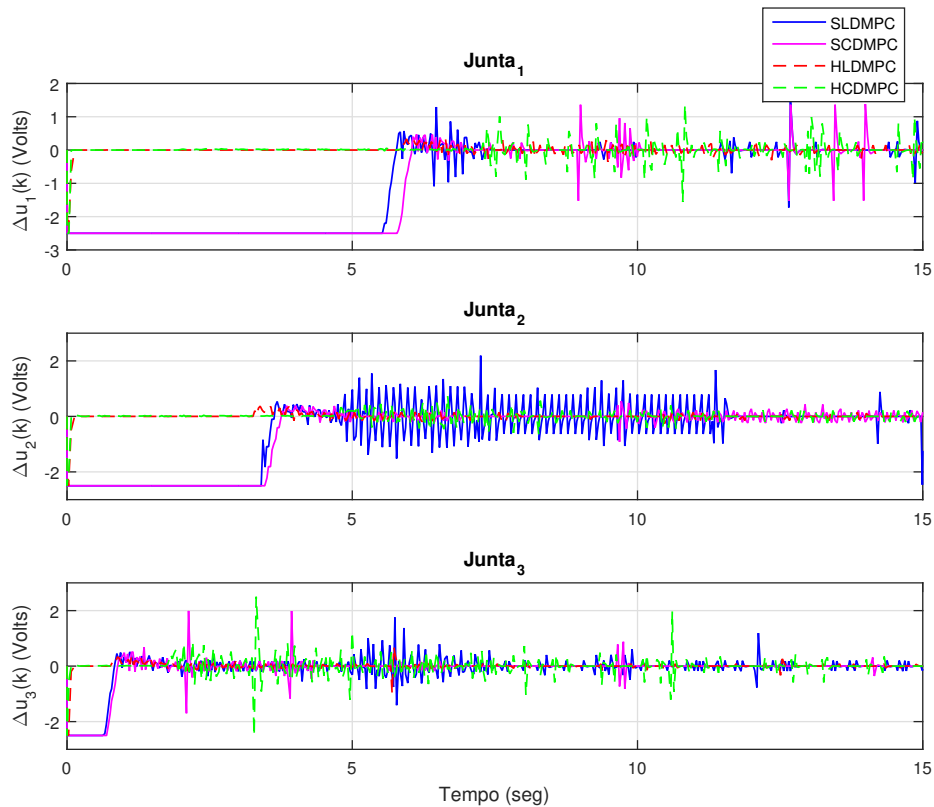


Figura 30 – Sinal de controle incremental dos testes experimentais do Mentor: com restrição.

Tabela 5 – Índices de desempenho dos testes experimentais do Mentor: com restrição.

		SLDMPC	SCDMPC	HLDMPC	HCDMPC
ISU	Junta 1	9,916632e+02	1,038599e+03	1,118855e+03	9,594759e+02
	Junta 2	8,860500e+02	8,945823e+02	7,985449e+02	6,172186e+02
	Junta 3	2,727560e+02	2,919535e+02	1,969860e+02	2,833262e+02
ISE	Junta 1	2,407956e+06	2,570734e+06	2,554809e+06	3,074729e+06
	Junta 2	4,112283e+05	4,289857e+05	3,793566e+05	4,850584e+05
	Junta 3	6,529193e+04	6,793756e+04	6,469071e+04	7,580861e+04
IAE	Junta 1	1,928211e+04	2,041156e+04	2,071518e+04	2,807881e+04
	Junta 2	7,247322e+03	7,549634e+03	6,838479e+03	7,728766e+03
	Junta 3	1,995837e+03	2,136626e+03	1,662248e+03	3,899473e+03
PO (%)	Junta 1	0,358288	0,716577	0,002345	1,086545
	Junta 2	4,466566	0,634034	0,634034	1,276161
	Junta 3	4,272727	4,504504	2,752292	3,478260
TVC	Junta 1	4,800597e+02	5,057912e+02	27,959702	26,024864
	Junta 2	4,095521e+02	3,252173e+02	23,279653	15,771925
	Junta 3	1,264332e+02	94,624064	24,307028	19,694226
ST(seg)	Junta 1	6,56	6,87	6,78	7,79
	Junta 2	4,40	4,81	4,35	5,49
	Junta 3	1,67	1,71	1,70	2,31

5.3 Coluna de destilação Binária de Wood e Berry (WB)

A coluna de destilação binária de Wood e Berry (1973) se tornou extremamente popular como um processo *benchmark* para testar o desempenho de controladores digitais. Isto foi motivado pelo fato de ter um tratamento simplificado e direto do problema de separação de metanol e água, aonde diversas malhas de controle internas, referentes ao controle de fluxos internos, temperatura, nível do reservatório, estão perfeitamente interligadas, resultando em um sistema MIMO 2×2 , com acoplamentos e atrasos assíncronos.

O modelo linear contínuo de WB pode ser representado por,

$$\begin{bmatrix} y_1(s) \\ y_2(s) \end{bmatrix} = \begin{bmatrix} \frac{12,8e^{-s}}{16,7s+1} & \frac{-18,9e^{-3s}}{21s+1} \\ \frac{6,6e^{-7s}}{10,9s+1} & \frac{19,4e^{-3s}}{14,4s+1} \end{bmatrix} \begin{bmatrix} u_1(s) \\ u_2(s) \end{bmatrix} + \begin{bmatrix} \frac{3,8e^{-8s}}{14,9s+1} \\ \frac{4,9e^{-3,4s}}{13,2s+1} \end{bmatrix} d(s) \quad (5.5)$$

É importante destacar que o modelo contínuo de WB é definido com a unidade de tempo em minutos. Então, sua forma discretizada via *ZOH* (Zero-Order Hold) para um período de amostragem $T_s = 1min$, é definida por:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} G_{11}(z) & G_{12}(z) \\ G_{21}(z) & G_{22}(z) \end{bmatrix} \begin{bmatrix} u_1(k) \\ u_2(k) \end{bmatrix} + \begin{bmatrix} G_{d11}(z) \\ G_{d21}(z) \end{bmatrix} d(k) \quad (5.6)$$

onde,

$$\begin{aligned} G_{11}(z) &= \frac{(0,7440-0,7094z^{-1})z^{-2}}{1-1,8954z^{-1}+0,8981z^{-2}}; \\ G_{12}(z) &= \frac{(-0,8789-0,8278z^{-1})z^{-4}}{1-1,8954z^{-1}+0,8981z^{-2}}; \\ G_{21}(z) &= \frac{(-0,5786-0,5398z^{-1})z^{-8}}{1-1,8452z^{-1}+0,8511z^{-2}}; \\ G_{22}(z) &= \frac{(-1,3015-1,1874z^{-1})z^{-4}}{1-1,8452z^{-1}+0,8511z^{-2}}; \\ G_{d11} &= \frac{0,2467z^{-9}}{1-0,9351z^{-1}}; \\ G_{d21} &= \frac{0,3575z^{-4}}{1-0,9270z^{-1}}; \end{aligned} \quad (5.7)$$

O processo de funcionamento de WB é bem simples e intuitivo. Consiste em quando há refluxo, $u_1(k)$, de mistura concentrada de metanol voltando à coluna a partir do topo, ocorre um aumento da concentração de metanol tanto no topo como na base. Ao injetar vapor, $u_2(k)$, a partir da base da coluna, a qual se aquece e aumenta a taxa de evaporação da mistura, reduzindo assim a concentração de metanol na base. A partir do aumento da taxa de alimentação, $d(k)$, a coluna é novamente perturbada pela adição de mais água e metanol. Assim a saída do topo, $y_1(k)$, libera o produto com o percentual mássico de metanol desejado e enquanto na saída da

base, $y_2(k)$, libera o produto com percentual mássico de metanol descartado. As variáveis em 5.5 e suas condições típicas de operação em que o modelo linearizado da coluna foi obtido, são definidas na Tab. 6, e na Figura 31 é apresentado a estrutura da coluna com suas respectivas entradas e saídas.

Tabela 6 – Definição das variáveis e condições típicas de operação para WB (WOOD; BERRY, 1973).

Variável	Descrição	Valores nominais
$y_1(k)$	saída do topo (Overhead MeOH).	96,25 mol %
$y_2(k)$	saída da base (Bottoms MeOH).	0,50 mol %
$u_1(k)$	Refluxo da mistura no topo.	1,95 lb/min (0,89 kg/min)
$u_2(k)$	Vapor na base (Reboiler steam).	1,71 lb/min (0,78 kg/min)
$d(k)$	Alimentação (Feed: MeOH / H_2O)	2,45 lb/min (1,11 kg/min)

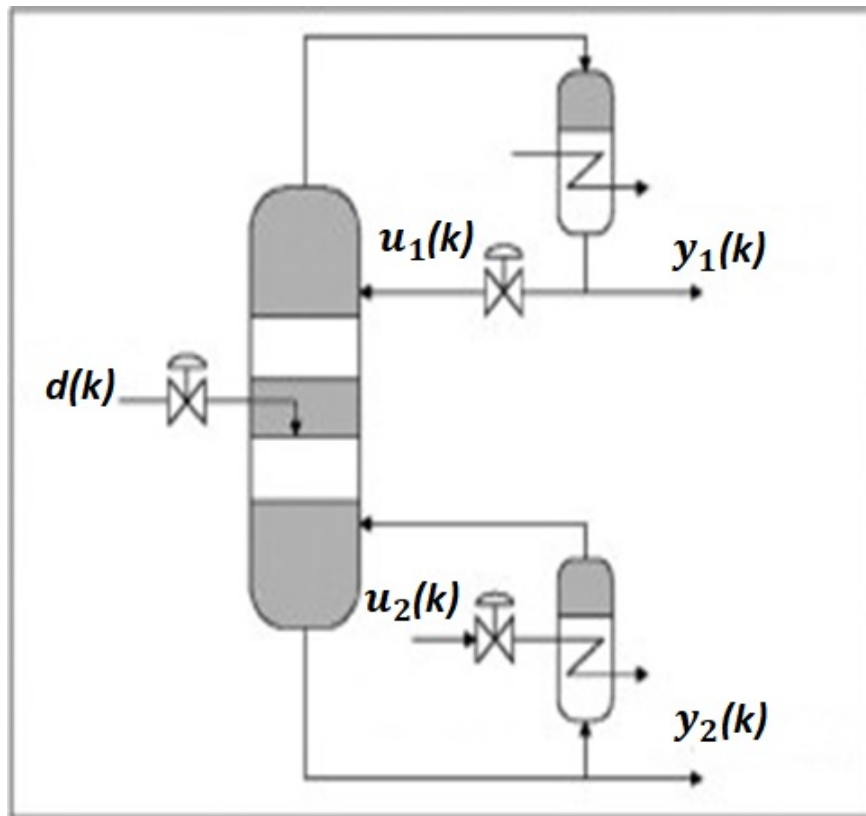


Figura 31 – Coluna de destilação binária de WB.

5.3.1 Simulações de WB

A seguir serão apresentados os resultados das simulações para os testes de rastreamento de referência e perturbação de carga com WB, para os casos com e sem restrições nas variáveis de entrada, tendo o objetivo de mostrar e comparar o desempenho dos métodos apresentados neste trabalho.

Os parâmetros de sintonia utilizados em todas as simulações foram: $N_p = 100$; $N_1 = N_2 = N_c = 2$; $a_1 = a_2 = 0,5$; $\Gamma = 100$ e $Q = C_a^T C_a$. Para o caso com restrições, foram impostas limitações somente nas variações do refluxo, $\Delta u_1(k)$, e do vapor da base, $\Delta u_2(k)$, sendo a elas, restrita uma faixa de operação definida por: $-0,003 \leq \Delta u(k) \leq +0,003$.

As Figuras 32, 33, 34, apresentam os resultados das simulações dos testes de rastreamento de referência para o caso *sem restrição*, e na Tab. 7 são mostrados os seus respectivos índices de desempenho. Em seguida, são apresentadas nas Figuras 36, 37, 38 as simulações com perturbação de carga para o caso *sem restrição* e na Tab. 9 são dados seus relativos índices de desempenho. Por fim, nas Figuras 39, 40, 41 são expostas às simulações com perturbação de carga para o caso *com restrição* e na Tab. 10 são exibidos seus pertinentes índices de desempenho.

Analisando os resultados dos testes de rastreamento de referência para o caso *sem restrição*, verifica-se que o *LDMPC* obteve os melhores índices *ISE*, *IAE* e *ST*, entretanto, *CDMPC* obteve os menores índices *ISU* e *TVC*. Similarmente para o caso *sem restrição* e com perturbação, $d(k)$, gerada pela alimentação da torre, *LDMPC* também obteve os melhores índices *ISE*, *IAE*, *ST* e melhor rejeição à perturbação de carga, porém, com maior *ISU* e *TVC*. De posse disso, pode-se afirmar que para a coluna de destilação de WB, o controlador utilizando as funções ortonormais de Laguerre obteve melhor desempenho. Isso demonstra a eficiência do *LDMPC* para sistemas MIMO com acoplamentos e atrasos assíncronos.

Para o caso *com perturbação de carga e restrição*, é possível afirmar que *SLDMPC* e *HLDMP* obtiveram os melhores resultados, com uma pequena superioridade para *SLDMPC* que apresentou os melhores índices *ISE*, *IAE*, *ST* e maior rejeição à perturbação ($d(k)$), porém, com maior sobre-sinal e com o pior *TVC*. Já em relação ao índice *ISU*, todos os controladores obtiveram gastos de energia similares.

Com o intuito de comparar os resultados obtidos com os dos controladores tradicionais baseados na estrutura PID, são apresentados na Tab. 8, os índices *IAE* de simulações com perturbação de carga de controladores digitais do tipo PI (Proportional Integral) e suas respectivas sintonias, sendo essas informações obtidas em Wood e Berry (1973). Analisando esses dados, verifica-se que o controlador *LDMPC* obteve melhores índices *IAE* quando sujeito ao mesmo tipo de perturbação de carga, como mostra a Tab. 9. Então, pode-se afirmar que o controlador *LDMPC* tem um desempenho superior em relação a certos controladores tradicionais utilizados na indústria.

Um fato que deve-se salientar, é que os controladores que utilizaram a programação quadrática de Hildreth para solucionar o cálculo do sinal de controle ótimo com restrições, apresentaram os menores *TVC*, sendo isso uma contribuição marcante deste algoritmo.

Na Figura 35 são apresentadas as posições dos 6 polos de malha fechada da coluna de destilação binária de WB para *LDMPC* e *CDMPC*.

Tabela 7 – Índices de desempenho de WB: sem restrição.

		CDMPC	LDMPC
ISU	u_1	4,991810	5,879128
	u_2	1,703555	1,992020
TVC	u_1	0,744055	1,343747
	u_2	0,572474	0,970497
ISE	y_1	12,001943	10,429439
	y_2	9,278226	6,677994
IAE	y_1	30,044560	22,575243
	y_2	24,755990	16,682652
ST(min)	y_1	144	118
	y_2	237	218

Tabela 8 – PI Desempenho: sem restrição e com perturbação.

Perturbação ($d(k) = 0,34 \text{ lb/min (0,15 kg/min)}$)						
Controlador (Code)	Saída do topo ($y_1(k)$)			Saída da base ($y_2(k)$)		
	K_p	K_i	IAE	K_p	K_i	IAE
PI (P1-3)	-0,20	-0,045	11,9	0,040	0,015	29,3
PI (R-1)	-0,09	-0,015	5,1	0,10	0,025	18,5
PI (C-1)	-0,35	-0,010	4,1	0,12	0,035	16,7

Tabela 9 – Índices de desempenho de WB: sem restrição e com perturbação.

		CDMPC	LDMPC
ISU	u_1	1,178505	1,156840
	u_2	0,234966	0,235421
TVC	u_1	0,025277	0,038944
	y_2	0,043238	0,044923
ISE	y_1	0,163192	0,052280
	y_2	0,244647	0,124460
IAE	y_1	2,684827	1,120043
	y_2	2,633007	1,744581
ST(min)	y_1	40	18
	y_2	34	28

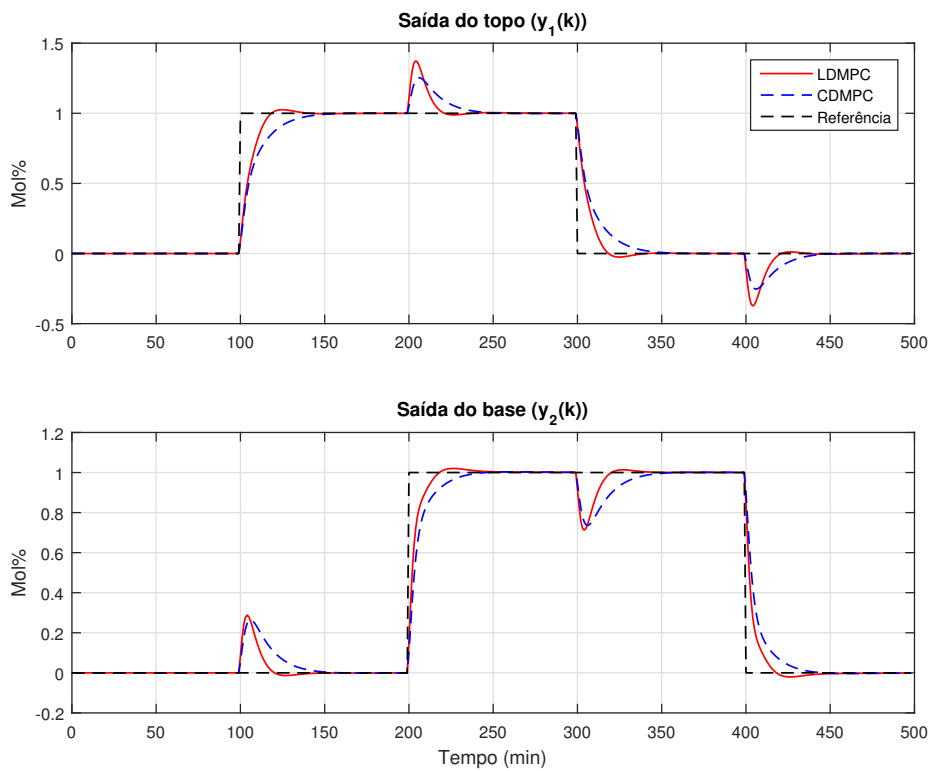


Figura 32 – Simulações de WB da saída do topo e da base: sem restrição.

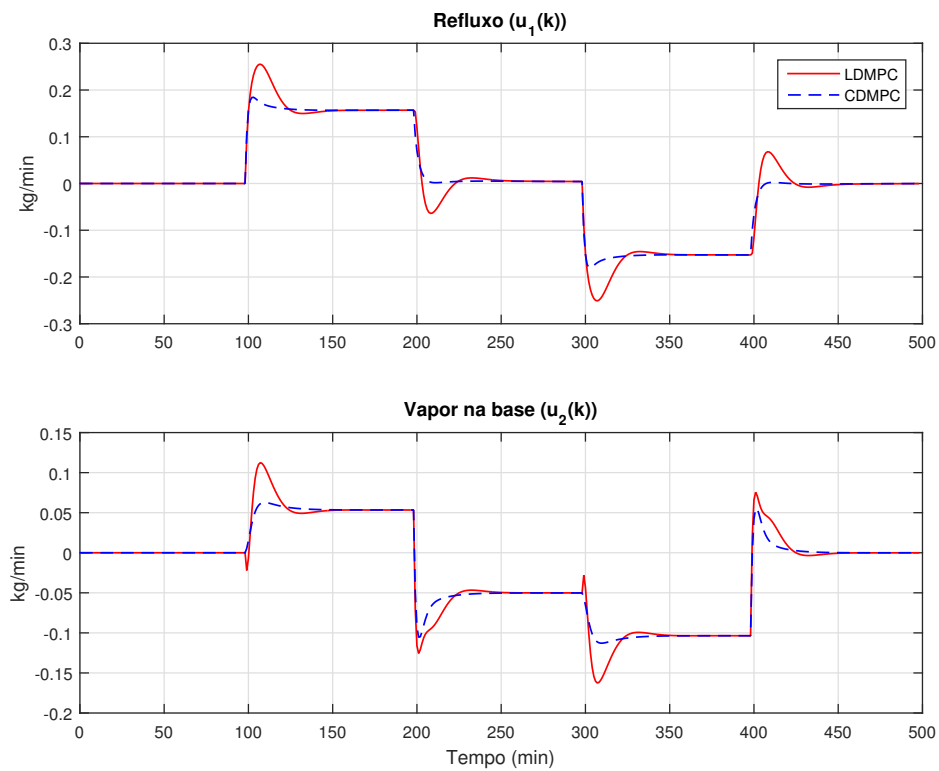


Figura 33 – Simulações de WB do refluxo e de vapor da base: sem restrição.

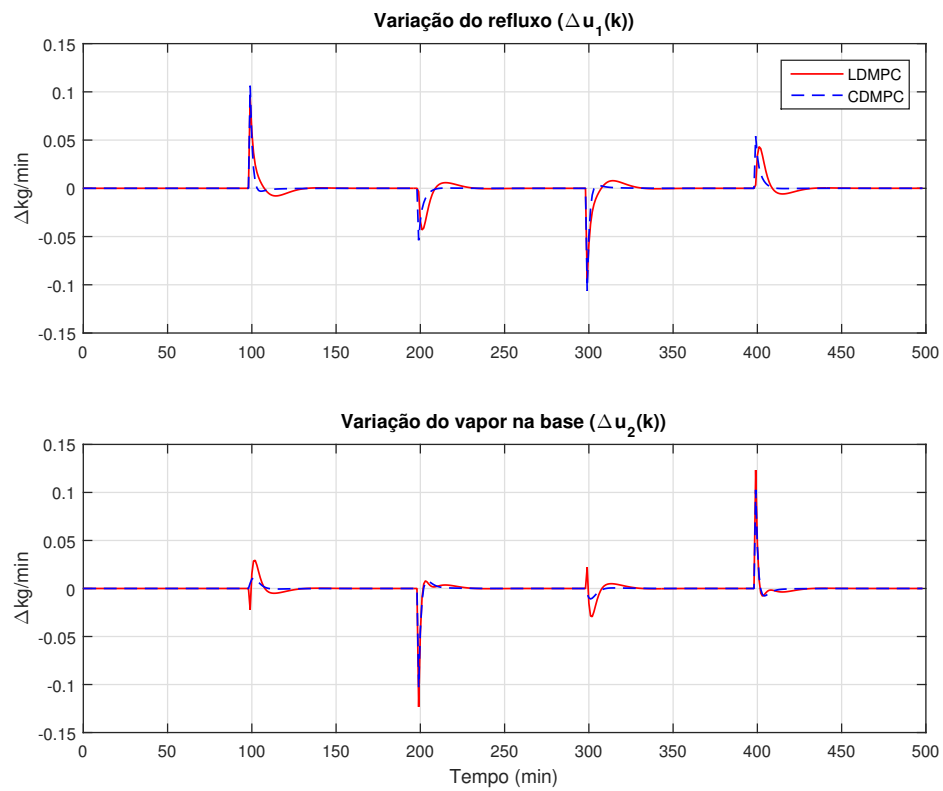


Figura 34 – Simulações de WB da variação de refluxo e de vapor da base: sem restrição.

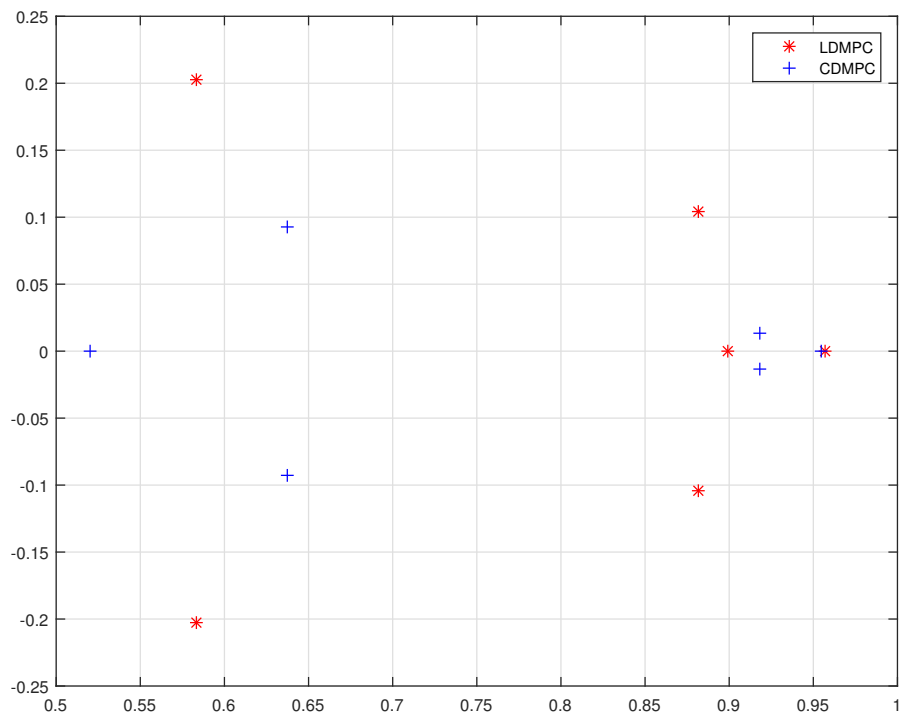


Figura 35 – Polos de malha fechada de WB.

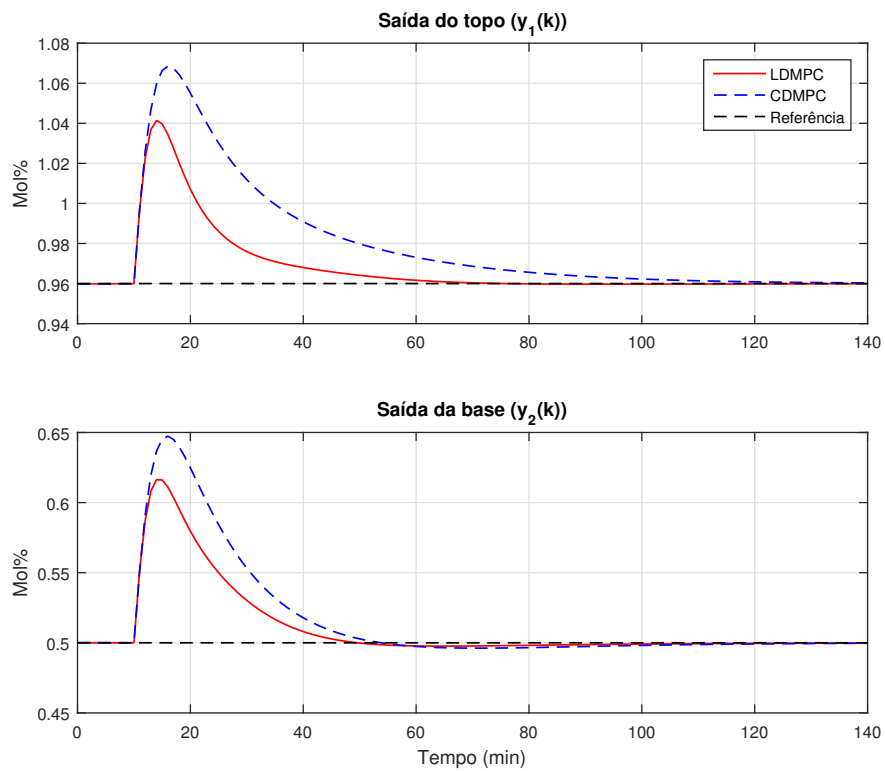


Figura 36 – Simulações de WB da saída do topo e da base: sem restrição e com perturbação.

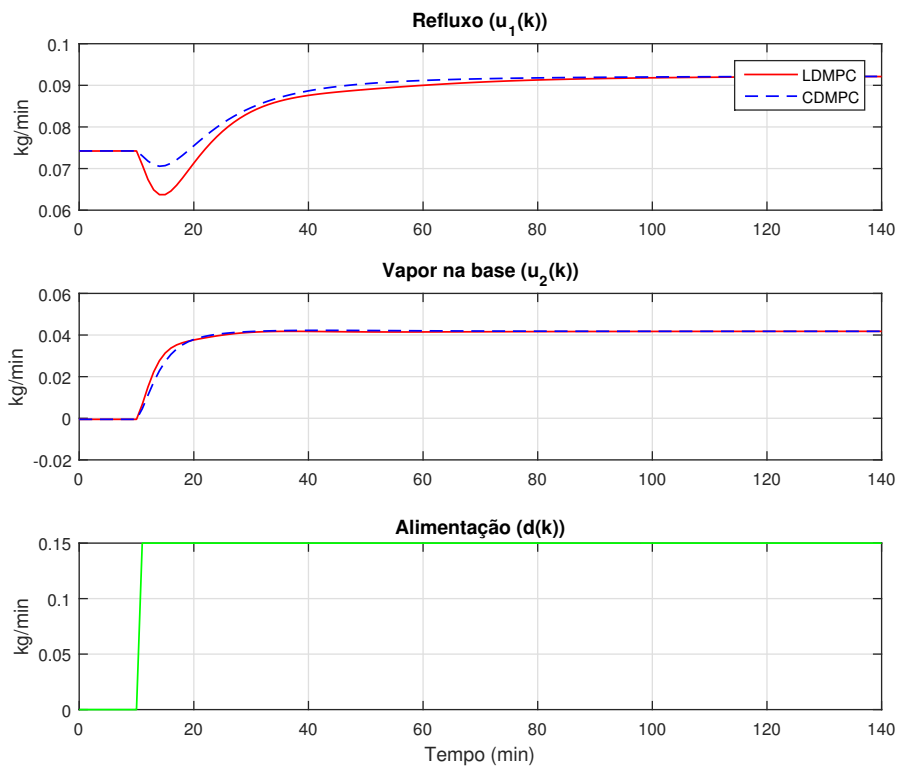


Figura 37 – Simulações de WB do refluxo e de vapor da base: sem restrição e com perturbação.

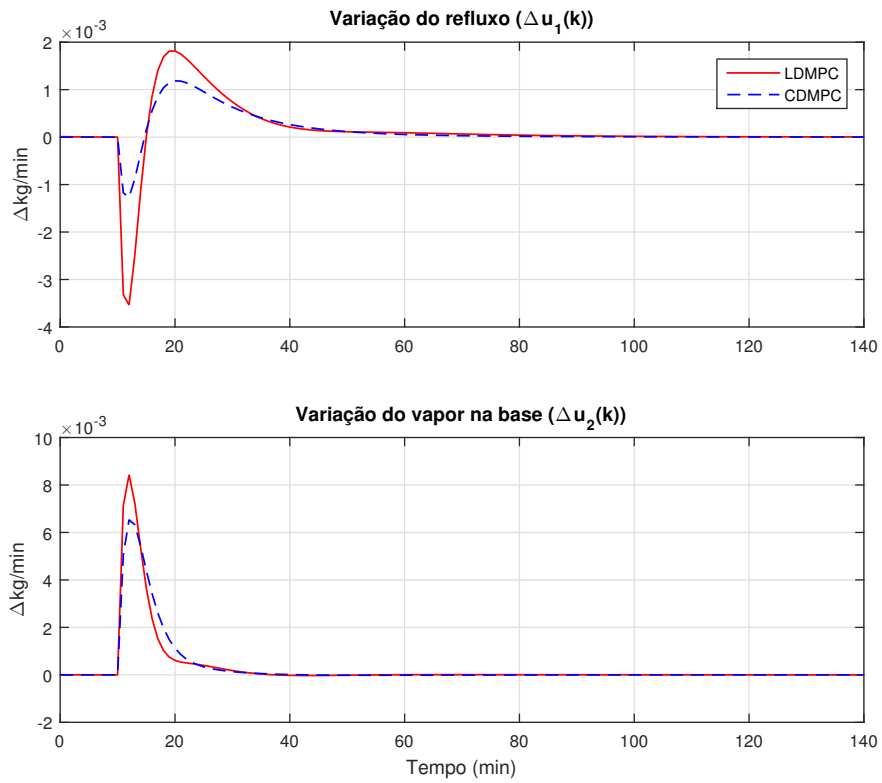


Figura 38 – Simulações de WB da variação de refluxo e de vapor da base: sem restrição e com perturbação.

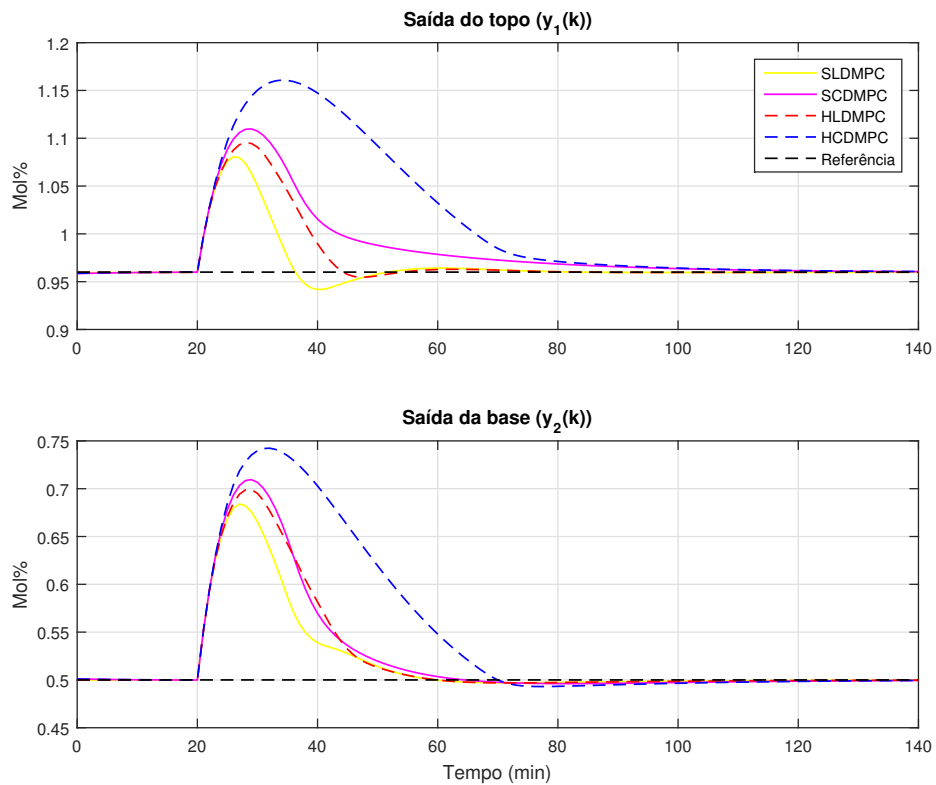


Figura 39 – Simulações de WB da saída do topo e da base: com perturbação e restrição.

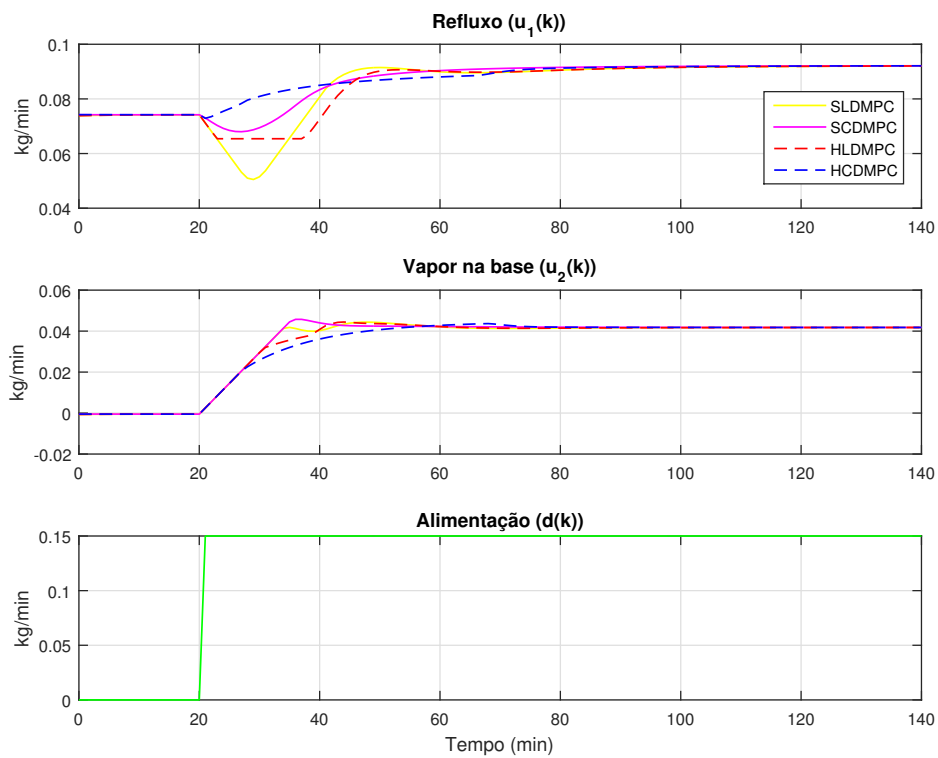


Figura 40 – Simulações de WB do refluxo e de vapor da base: com perturbação e restrição.

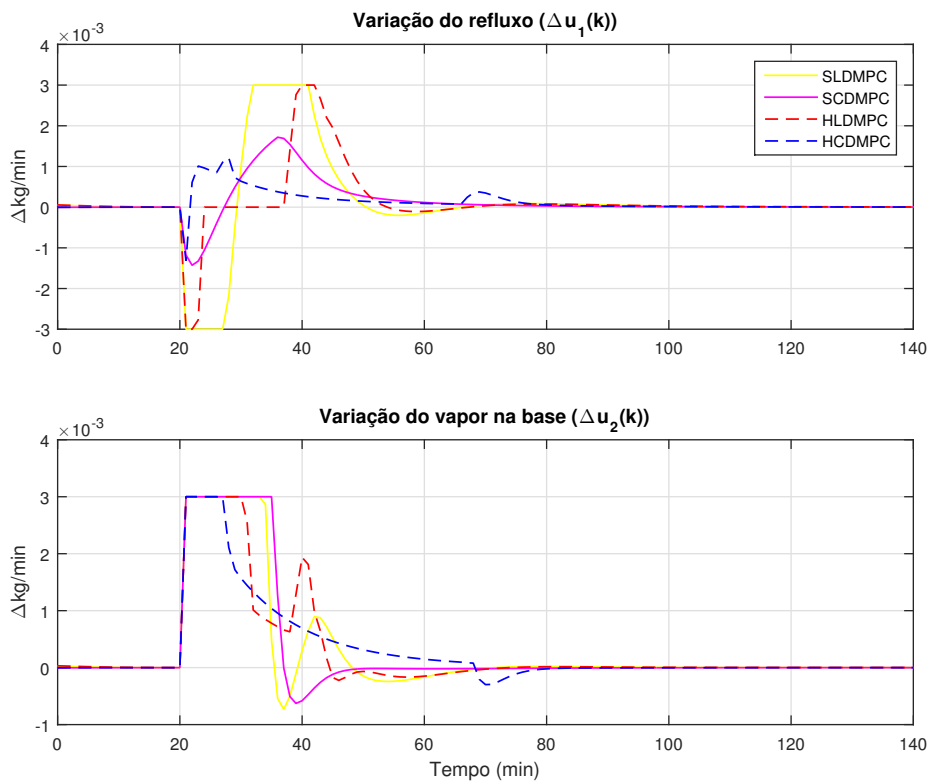


Figura 41 – Simulações de WB da variação de refluxo e de vapor da base: com perturbação e restrição.

Tabela 10 – Índices de desempenho de WB: com perturbação e restrição.

		SLDMPC	SCDMPC	HLDMPCC	HCDMPCC
ISU	u1	1,023857	1,052402	1,023804	1,063215
	u2	0,194683	0,197428	0,192807	0,187689
TVC	y1	0,069443	0,031357	0,030630	0,020554
	y2	0,052316	0,050254	0,048581	0,046094
ISE	y1	0,113895	0,282537	0,197799	0,965249
	y2	0,333310	0,516546	0,479182	1,190326
IAE	y1	1,438500	3,201492	2,001555	6,680095
	y2	2,765895	3,635663	3,431667	6,808582
ST(min)	y1	14	38	21	51
	y2	31	35	31	47

5.4 Conclusões

Neste capítulo foram apresentados e analisados os resultados de *LDMPC* e *CDMPC* aplicados no manipulador robótico Mentor e da coluna de destilação binária para os casos com e sem restrições no sinal de controle.

Inicialmente, foram realizadas simulações com o modelo matemático MIMO 3×3 em 5.1, o qual representa o manipulador robótico Mentor, sendo apresentados os resultados das simulações para o caso sem restrições e posteriormente para o caso com restrições. Após isso, foram realizados testes reais com restrições na faixa de operação do sinal de controle, onde foi utilizado um circuito drive de acionamento e aquisição de dados para realizar a conversão A-D/D-A e os acionamentos dos motores. Os resultados dos controladores foram comparados através de índices de desempenhos e gráficos, com isso mostrando a eficiência das funções de Laguerre em conjunto com a programação quadrática de Hildreth para reduzir a perda de desempenho em processos com restrições nas variáveis de entrada.

Em seguida, foi apresentada a torre de destilação binária de WB, descrevendo seu funcionamento, as variáveis envolvidas e seu modelo matemático descrito por uma matriz de funções de transferência MIMO 2×2 em 5.5. Onde primeiramente, foram realizadas simulações de testes de rastreamento sem restrições. Depois, foram apresentadas as simulações com perturbação de carga sem restrições. No final, foram mostradas as simulações com perturbação de carga com restrições na variação do fluxo da entrada da base ($\Delta u_2(k)$) e no refluxo do topo ($\Delta u_1(k)$). Também através de índices de desempenho, foi mostrada a eficiência do *LDMPC* para sistemas MIMO com restrições nas entradas e atrasos de transporte assíncronos com acoplamento.

Como trabalhos correlatos com processos similares aos utilizados neste capítulo, destacam-se: Wang *et al.* (2011), que utilizam as Funções de Laguerre para controlar um robô manipulador com 2DOF com restrições, e Solaman e Francis (2015), o qual aplicam o controlador preditivo

com Funções de Laguerre em uma coluna de destilação binária, porém, sem restrições nas variáveis de entrada.

CONCLUSÃO

Esta dissertação centrou-se em apresentar os potenciais benefícios das Função Laguerre dentro dos MPC aplicados em sistemas SISO e MIMO, para os casos com e sem restrições nas variáveis de entrada, a partir de uma abordagem no espaço de estados, mostrando suas vantagens em relação aos MPC que utilizam uma abordagem clássica de controle preditivo.

É válido ressaltar que os *LDMPC* conseguem simplificar o cálculo da lei de controle de horizonte retrocedente, $\Delta u(k)$, resultando em menor carga computacional no processo. Isso porque, as Funções de Laguerre conseguem tornar o horizonte de predição, N_p , e de controle, N , mais efetivos e otimizados, conseguindo desempenhos satisfatórios sem a necessidade de longos horizontes de predição e de controle. Quando comparados com os desempenhos do *CDMPC*, verifica-se que um grande N_p e N_c , equivalem a um pequeno N_p e N de *LDMPC*. Este fato é interessante para sistemas com altas taxas de amostragens e também para controladores preditivos adaptativos que precisam modificar seus parâmetros de sintonia a cada instante de tempo. Isso pode reduzir consideravelmente os custos com equipamentos, processadores e afins, trazendo uma enorme economia para os processos de produção industrial. Além do mais, o *LDMPC* apresenta um parâmetro a mais de sintonia em relação ao *CDMPC*, que é o parâmetro de sintonia a , responsável por uma maior liberdade de sintonia e também pelo *ajuste fino* de desempenho desejada.

Para os casos com restrições, o *LDMPC* se mostrou bastante eficiente, apresentando redução aceitável do conflito entre a viabilidade do sinal de controle e o desempenho em malha fechada, tendo um desempenho superior se comparado com o *CDMPC*. Quanto a utilização da programação quadrática de Hildreth em conjunto com as Funções de Laguerre, nos testes realizados com o Mentor e WB, o método se mostrou razoável, obtendo resultados equivalentes aos dos controladores com uma simples saturação. No entanto, é notável a redução da variação do incremento do sinal de controle, tanto para sistemas SISO e como para MIMO, sendo isso a principal contribuição deste algoritmo quadrático adaptado para trabalhar em conjunto com os controladores preditos. Porém, é chegada à conclusão que para determinados sistemas, principal-

mente para aqueles que não precisam de um controle apurado na variação de $\Delta u(k)$, o melhor é utilizar uma simples saturação, por conta de sua simplicidade e menor custo computacional.

É válido também observar que as técnicas utilizadas nos projetos dos controladores apresentados neste trabalho, são de simples implementação para sistemas SISO e MIMO. Isso pelo motivo de ser utilizada uma abordagem no espaço de estados, que torna o método generalizado, assim, sem muitas modificações, é possível projetar um controlador para um sistema com n entradas e s saídas.

Deve-se também dar destaque aos testes reais realizados no Mentor, configurado como um sistema MIMO 3×3 , além das contribuições construtivas dessa planta, deixando-a compatível com componentes atuais de processamento, tal como apresentado no Apêndice A. Nos testes com o Mentor foram mostrados a eficiência das Funções de Laguerre em um processo real com restrições de operação, comprovando os resultados obtidos nas simulações e teorias apresentadas. Além disto, os algoritmos desenvolvidos foram testados na coluna de destilação binária de WB, que é uma planta benchmark MIMO 2×2 , com atrasos assíncronos e acoplamentos, sendo um desafio interessante para os controladores apresentados neste trabalho.

Uma questão que deve ser ressaltar, é a análise de estabilidade e desempenho robusto para esses tipos de controladores preditivos multivariáveis no espaço de estados. As análises usuais como H_∞ e μ análise (SKOGESTAD; POSTLETHWAITE, 2007), precisam que o modelo do controlador esteja na forma de matriz de funções de transferência, no entanto, o modelo dos controladores apresentados neste trabalho, estão no espaço de estados ($\Delta u(k) = k_{mpc} x_a(k)$), assim, impossibilitando a análise de robustez pelos métodos tradicionais. Então, maneiras de realizar a análise de estabilidade e desempenho robusto para MPC MIMO no espaço de estados devem ser investigadas.

As principais contribuições da pesquisa apresentada nesta dissertação, são destacadas na seção a seguir.

6.1 Principais contribuições

- Apresentação e desenvolvimento do projeto de controladores preditivos multivariáveis no espaço de estados utilizando as Funções de Laguerre.
- Apresentação e discussão dos benefícios das Funções de Laguerre em controladores preditivos para: diminuir o custo computacional utilizado para o cálculo da lei de controle de horizonte retrocedente, melhorar a eficiência do horizonte de controle e de predição, e reduzir o conflito entre a viabilidade do sinal de controle e o desempenho de malha fechada para sistemas com restrições nas variáveis de entrada.
- Apresentação e implementação do projeto da programação quadrática de Hildreth em conjunto com as Funções de Laguerre para resolver problemas com restrições no sinal de controle.

- Desenvolvimento de uma nova “interface” de acionamento e aquisição de dados, detalhada no Apêndice A.
- Testes reais com os algoritmos desenvolvidos em um robô manipulador com 3DOF.

6.2 Publicações

Nesta seção são apresentados os trabalhos publicados ao longo do curso de Mestrado e que tiveram relação direta e indireta com o conteúdo apresentado nesta dissertação.

6.2.1 *Trabalhos relacionados diretamente com a pesquisa*

PINHEIRO, T. C. F.; DUTRA, B. G. ; CASTRO, L. A. M. ; ARAUJO, M. S. ; SILVA, M. G. ; SILVEIRA, A. S. Identificação de modelo discreto linear de um servomecanismo usando mínimos quadrados estendido recursivo. DINCON Conferência Brasileira de Dinâmica, Controle e Aplicações, 2017, São José do Rio Preto.

Neste artigo publicado no DICON 2017, foi apresentada a técnica de identificação de modelo discreto linear de um servomecanismo usando mínimos quadrados, utilizada para identificar o modelo discreto do manipulador Mentor em 5.1.

Outro trabalho com relação direta, foi o artigo com o título *Constrained Discrete Model Predictive Control of an Arm-Manipulator Using Laguerre's Function*, submetido em dezembro de 2017 para a revista *Control Engineering Practice-Elsevier* e estando em processo de revisão. Este artigo contém todos os resultados obtidos com as simulações e testes práticos com o Mentor no capítulo 5.

6.2.2 *Trabalhos relacionados indiretamente com a pesquisa*

PINHEIRO, T. C. F., SILVA, A. F., SILVEIRA, A. S. ARAÚJO, M. S., 2016. Stochastic Augmentation by Generalized Minimum Variance Control with RST LoopShaping. Vitória: XXI Congresso Brasileiro de Automática.

Neste artigo publicado no CBA 2016, apesar de ter sido um controlador diferente ao apresentado neste trabalho, a planta utilizada para os testes simulados foi a *junta₁* robótica do Mentor, representada pela função de transferência $G_{11}(z)$ em 5.1. Foi o primeiro contato com o controle de posição de uma junta do robô manipulador Mentor.

6.3 Proposta de trabalhos futuros

Tendo em vista o número crescente de novos algoritmos complexos de controle, o LDMPC já é uma realidade que as indústrias devem lidar de alguma forma em um futuro próximo. Hoje, o CDMPC já é um produto disponível no mercado e a sua configuração de

sintonia é baseada em três termos, N_c , N_p e Γ , está aos poucos se tornando um novo padrão entre técnicos e engenheiros da indústria, o qual começou na década de 1970 e ainda está em fase de transição. Quanto ao LDMPC, que é uma evolução do caso clássico por intermédio de funções ortonormais, a intenção é investigar maneiras de correlacionar os conceitos de sintonia do PID, que é comumente conhecido por técnicos e engenheiros, com os conceitos de sintonia do LDMPC, para torná-lo escondido para o usuário final, mas mantendo as vantagens de um controlador preditivo, ou seja, mascará-lo para que o operador pense que se trata de um simples controlador tradicional. Isso porque, os operadores, técnicos e engenheiros ainda têm uma certa resistência aos novos conceitos apresentados pelos controladores preditivos, optando em sua maioria pelo tradicional.

Outra proposta que pode-se destacar, é a extensão do controle multivariável usando as Funções de Laguerre para processos estocásticos. O método apresentado neste trabalho utiliza o modelo determinístico ARX (AutoRegressive with eXogenous inputs) e com as devidas modificações e adições no projeto, pode ser utilizado o modelo ARMAX (AutoRegressive Moving Average with EXogenous inputs), então podendo controlar processos, sujeitos a ruídos e incertezas estocásticas. Esta proposta está em fase de desenvolvimento, sendo que o projeto para sistemas SISO já foi desenvolvido, faltando apenas a generalização para o caso MIMO.

Por fim, outra proposta de trabalho, é o desenvolvimento do controle multivariável usando as Funções de Laguerre com compensador de zona morta, tendo o objetivo de aumentar a precisão do controle de posição de servomecanismos que normalmente apresentam esse tipo de não linearidade.

REFERÊNCIAS

- ABDULLAH, M.; ROSSITER, J. A. Utilising Laguerre function in predictive functional control to ensure prediction consistency. In: IEEE. **Control (CONTROL), 2016 UKACC 11th International Conference on**. [S.l.], 2016. p. 1–6. Citado na página 33.
- AKÇAY, H.; NINNESS, B. Orthonormal basis functions for modelling continuous-time systems. **Signal processing**, Elsevier, v. 77, n. 3, p. 261–274, 1999. Citado na página 49.
- ALVAREZ, T.; PRADA, C. de. Handling infeasibilities in predictive control. **Computers & chemical engineering**, Elsevier, v. 21, p. S577–S582, 1997. Citado 2 vezes nas páginas 30 e 32.
- ANDERSON, B. D.; MOORE, J. B. **Linear optimal control**. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1971. v. 197. Citado na página 28.
- ÅSTRÖM, K. J.; WITTENMARK, B. **Computer-controlled systems: theory and design**. [S.l.]: Courier Corporation, 2013. Nenhuma citação no texto.
- BATEMAN, H.; ERDÉLYI, A.; MAGNUS, W.; OBERHETTINGER, F.; TRICOMI, F. G. **Higher transcendental functions**. [S.l.]: McGraw-Hill New York, 1955. v. 2. Citado na página 49.
- BAY, J. S. **Fundamentals of linear state space systems**. [S.l.]: McGraw-Hill Science, Engineering & Mathematics, 1999. Citado na página 53.
- BAZARAA, M. S.; SHERALI, H. D.; SHETTY, C. M. **Nonlinear programming: theory and algorithms**. [S.l.]: John Wiley & Sons, 2013. Citado na página 69.
- BEMPORAD, A.; MORARI, M.; RICKER, N. L. Model predictive control toolbox 3 user's guide. **The mathworks**, 2010. Citado na página 42.
- BÖTTCHER, A.; GRUDSKY, S. M. **Toeplitz matrices, asymptotic linear algebra, and functional analysis**. [S.l.]: Springer Science & Business Media, 2000. v. 18. Citado na página 51.
- CAMACHO, E. F.; ALBA, C. B. **Model predictive control**. [S.l.]: Springer Science & Business Media, 2013. Citado na página 42.
- CAMACHO, E. F.; BORDONS, C. **Model predictive control in the process industry**. [S.l.]: Springer Science & Business Media, 2012. Citado 2 vezes nas páginas 28 e 29.
- CAUSA, J.; KARER, G.; NÚÑEZ, A.; SÁEZ, D.; ŠKRJANC, I.; ZUPANČIČ, B. Hybrid fuzzy predictive control based on genetic algorithms for the temperature control of a batch reactor. **Computers & chemical engineering**, Elsevier, v. 32, n. 12, p. 3254–3263, 2008. Citado na página 29.
- CHAI, S.; WANG, L.; ROGERS, E. Cascade model predictive control of a PMSM with periodic disturbance rejection. In: IEEE. **Australian Control Conference (AUCC), 2011**. [S.l.], 2011. p. 309–314. Citado 2 vezes nas páginas 47 e 76.

- _____. Model predictive control of a permanent magnet synchronous motor. In: IEEE. **IECon 2011-37th Annual Conference on IEEE Industrial Electronics Society**. [S.l.], 2011. p. 1928–1933. Citado na página 47.
- CLARKE, D.; GAWTHROP, P. Self-tuning control. In: IET. **Proceedings of the Institution of Electrical Engineers**. [S.l.], 1979. v. 126, n. 6, p. 633–640. Citado na página 29.
- CLARKE, D. W.; MOHTADI, C.; TUFFS, P. Generalized predictive control—part i. the basic algorithm. **Automatica**, Elsevier, v. 23, n. 2, p. 137–148, 1987. Citado na página 29.
- CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control - a computer control algorithm. In: **Joint automatic control conference**. [S.l.: s.n.], 1980. p. 72. Citado na página 29.
- DOVŽAN, D.; ŠKRJANC, I. Predictive functional control based on an adaptive fuzzy model of a hybrid semi-batch reactor. **Control Engineering Practice**, Elsevier, v. 18, n. 8, p. 979–989, 2010. Citado na página 29.
- DUBRAVIĆ, A.; ŠEHIĆ, Z.; BURGIĆ, M. Orthonormal functions based model predictive control of pH neutralization process. **Tehnički vjesnik**, Citeseer, v. 21, n. 6, p. 1249–1253, 2014. Citado na página 49.
- EDWARDS, H. M. An appreciation of kronecker. **The Mathematical Intelligencer**, Springer, v. 9, n. 1, p. 28–35, 1987. Citado na página 37.
- ETTEFAGH, M. H.; NARAGHI, M.; TOWHIDKHAH, F.; DONÁ, J. D. Model predictive control of linear time varying systems using Laguerre functions. In: IEEE. **Control Conference (AuCC), 2016 Australian**. [S.l.], 2016. p. 120–125. Citado na página 33.
- FLETCHER, R. Practical methods of optimization john wiley & sons. **New York**, v. 80, 1987. Citado na página 69.
- FUKUSHIMA, H.; BITMEAD, R. R. Robust constrained model predictive control using closed-loop prediction. In: IEEE. **American Control Conference, 2003. Proceedings of the 2003**. [S.l.], 2003. v. 3, p. 2511–2516. Citado na página 32.
- GARRIGA, J. L.; SOROUSH, M. Model predictive control tuning methods: A review. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 49, n. 8, p. 3505–3515, 2010. Citado na página 42.
- GERKSIC, S.; STRMCNIK, S. State-space predictive control of a gas conditioning unit. In: IEEE. **Control Conference (ECC), 2001 European**. [S.l.], 2001. p. 54–59. Citado na página 28.
- HILDRETH, C. A quadratic programming procedure. **Naval Research Logistics (NRL)**, Wiley Online Library, v. 4, n. 1, p. 79–85, 1957. Citado na página 68.
- HUYCK, B.; BRABANTER, J. D.; MOOR, B. D.; IMPE, J. V.; LOGIST, F. Model predictive control of a pilot-scale distillation column using a programmable automation controller. In: IEEE. **Control Conference (ECC), 2013 European**. [S.l.], 2013. p. 1053–1058. Citado 2 vezes nas páginas 47 e 76.
- JAMIL, N.; CHEN, X.; CLONINGER, A. Hildreth's algorithm with applications to soft constraints for user interface layout. **Journal of Computational and Applied Mathematics**, Elsevier, v. 288, p. 193–202, 2015. Citado na página 68.

- JIMÉNEZ, D. M. **Robustness aspects of Model Predictive Control**. [S.l.]: Publicacions de la Universitat Autònoma de Barcelona, 2000. Citado na página 65.
- KAILATH, T. **Linear systems**. [S.l.]: Prentice-Hall Englewood Cliffs, NJ, 1980. v. 156. Citado na página 53.
- KALMAN, R. E. *et al.* Contributions to the theory of optimal control. **Bol. Soc. Mat. Mexicana**, Citeseer, v. 5, n. 2, p. 102–119, 1960. Citado na página 28.
- _____. A new approach to linear filtering and prediction problems. **Journal of basic Engineering**, v. 82, n. 1, p. 35–45, 1960. Citado na página 28.
- KWONG, W. **Introdução ao Controle Preditivo com MATLAB**. [S.l.]: São Carlos, SP, 2005. Citado 2 vezes nas páginas 27 e 29.
- ŁAWRYŃCZUK, M. Approximate state-space model predictive control. In: IEEE. **Methods and Models in Automation and Robotics (MMAR), 2015 20th International Conference on**. [S.l.], 2015. p. 770–775. Citado na página 28.
- LUENBERGER, D. G. **Optimization by vector space methods**. [S.l.]: John Wiley & Sons, 1969. Citado na página 68.
- MATEESCU, R.; STEFANOIU, D.; CHRISTOV, N. Discrete-time MPC with constraints for variable speed wind turbines. In: IEEE. **Systems and Computer Science (ICSCS), 2013 2nd International Conference on**. [S.l.], 2013. p. 127–132. Citado 2 vezes nas páginas 47 e 76.
- MATHWORKS. **Choose Sample Time and Horizons**. [S.l.], 2017 (acessado em 12 de Outubro, 2017). <<https://www.mathworks.com/help/mpc/ug/choosing-sample-time-and-horizons.html>>. Nenhuma citação no texto.
- MAYNE, D. Q.; RAWLINGS, J. B.; RAO, C. V.; SCOKAERT, P. O. Constrained model predictive control: Stability and optimality. **Automatica**, Elsevier, v. 36, n. 6, p. 789–814, 2000. Citado na página 27.
- MINGZHU, X.; YIPING, J.; CUNZHI, P.; ZHANZHONG, W. An incremental predictive functional control based on kautz model. In: IEEE. **Sustainable Power Generation and Supply, 2009. SUPERGEN'09. International Conference on**. [S.l.], 2009. p. 1–5. Citado na página 49.
- PEREIRA, W. F. A. **Projeto e avaliação do controlador preditivo generalizado sujeito a restrições via métodos de otimização de pontos interiores**. Mestrado dissertação — UFSC, 1997. Citado na página 69.
- PHAM, T. T.; MEESOMBOON, A. Model predictive control with Laguerre functions for a buoyancy-driven airship. **Procedia Computer Science**, Elsevier, v. 86, p. 281–284, 2016. Citado na página 33.
- POTOČNIK, B.; MUŠIČ, G.; ŠKRJANC, I.; ZUPANČIČ, B. Model-based predictive control of hybrid systems: a probabilistic neural-network approach to real-time control. **Journal of Intelligent and Robotic Systems**, Springer, v. 51, n. 1, p. 45–63, 2008. Citado na página 29.
- RICHALET, J.; O'DONOVAN, D. **Predictive functional control: principles and industrial applications**. [S.l.]: Springer Science & Business Media, 2009. Citado na página 27.

- RICHALET, J.; RAULT, A.; TESTUD, J.; PAPON, J. Model predictive heuristic control: Applications to industrial processes. **Automatica**, Elsevier, v. 14, n. 5, p. 413–428, 1978. Citado na página 29.
- ROSICH, A.; OCAMPO-MARTINEZ, C. Real-time experimental implementation of predictive control schemes in a small-scale pasteurization plant. In: **Developments in Model-Based Optimization and Control**. [S.l.]: Springer, 2015. p. 255–273. Citado na página 27.
- ROSSITER, J.; KOUVARITAKIS, B.; GOSSNER, J. Feasibility and stability for constrained stable predictive control. In: **Proceedings of the Third IEEE Conference on Control Applications, Glasgow, UK**. [S.l.: s.n.], 1994. p. 1885–1890. Citado na página 31.
- ROSSITER, J. A. **Model-based predictive control: a practical approach**. [S.l.]: CRC press, 2003. Citado na página 39.
- ROSSITER, J. A.; WANG, L. Exploiting Laguerre functions to improve the feasibility/performance compromise in MPC. In: IEEE. **Decision and Control, 2008. CDC 2008. 47th IEEE Conference on**. [S.l.], 2008. p. 4737–4742. Citado 3 vezes nas páginas 30, 32 e 64.
- ROSSITER, J. A.; WANG, L.; VALENCIA-PALOMO, G. Efficient algorithms for trading off feasibility and performance in predictive control. **International Journal of Control**, Taylor & Francis, v. 83, n. 4, p. 789–797, 2010. Citado na página 64.
- SATO, T.; YANO, A.; MASUDA, S.; ARAKI, N.; KONISHI, Y. Two-degree-of-freedom multirate generalized predictive control based on state-space representation. In: IEEE. **Advanced Mechatronic Systems (ICAMechS), 2016 International Conference on**. [S.l.], 2016. p. 121–125. Citado na página 28.
- SKOGESTAD, S.; POSTLETHWAITE, I. **Multivariable feedback control: analysis and design**. [S.l.]: Wiley New York, 2007. v. 2. Citado na página 100.
- SOLAMAN, M.; FRANCIS, R. M. Model predictive controller based on Laguerre function for binary distillation column. **International Journal of Scientific Engineering and Research, IJSER**, 2015. Citado 2 vezes nas páginas 33 e 97.
- VALENCIA-PALOMO, G.; ROSSITER, J. PLC implementation of a predictive controller using Laguerre functions and multi-parametric solutions. IET, 2010. Citado na página 32.
- _____. Using Laguerre functions to improve efficiency of multi-parametric predictive control. In: IEEE. **American Control Conference (ACC), 2010**. [S.l.], 2010. p. 4731–4736. Citado na página 32.
- WAHLBERG, B. System identification using Laguerre models. **IEEE Transactions on Automatic Control**, IEEE, v. 36, n. 5, p. 551–562, 1991. Citado 2 vezes nas páginas 49 e 50.
- WANG, L. Discrete model predictive controller design using Laguerre functions. **Journal of process control**, Elsevier, v. 14, n. 2, p. 131–142, 2004. Citado 4 vezes nas páginas 30, 32, 33 e 64.
- _____. **Model predictive control system design and implementation using MATLAB ®**. [S.l.]: Springer Science & Business Media, 2009. Citado 5 vezes nas páginas 32, 41, 49, 58 e 59.

WANG, L.; FREEMAN, C. T.; CHAI, S.; ROGERS, E. Multivariable repetitive-predictive control of a robot arm with experimental results. **IFAC Proceedings Volumes**, Elsevier, v. 44, n. 1, p. 7672–7677, 2011. Citado 2 vezes nas páginas 32 e 97.

WISMER, D. A.; CHATTERGY, R. **Introduction to nonlinear optimization: a problem solving approach**. [S.l.]: North-Holland New York, 1978. Citado na página 68.

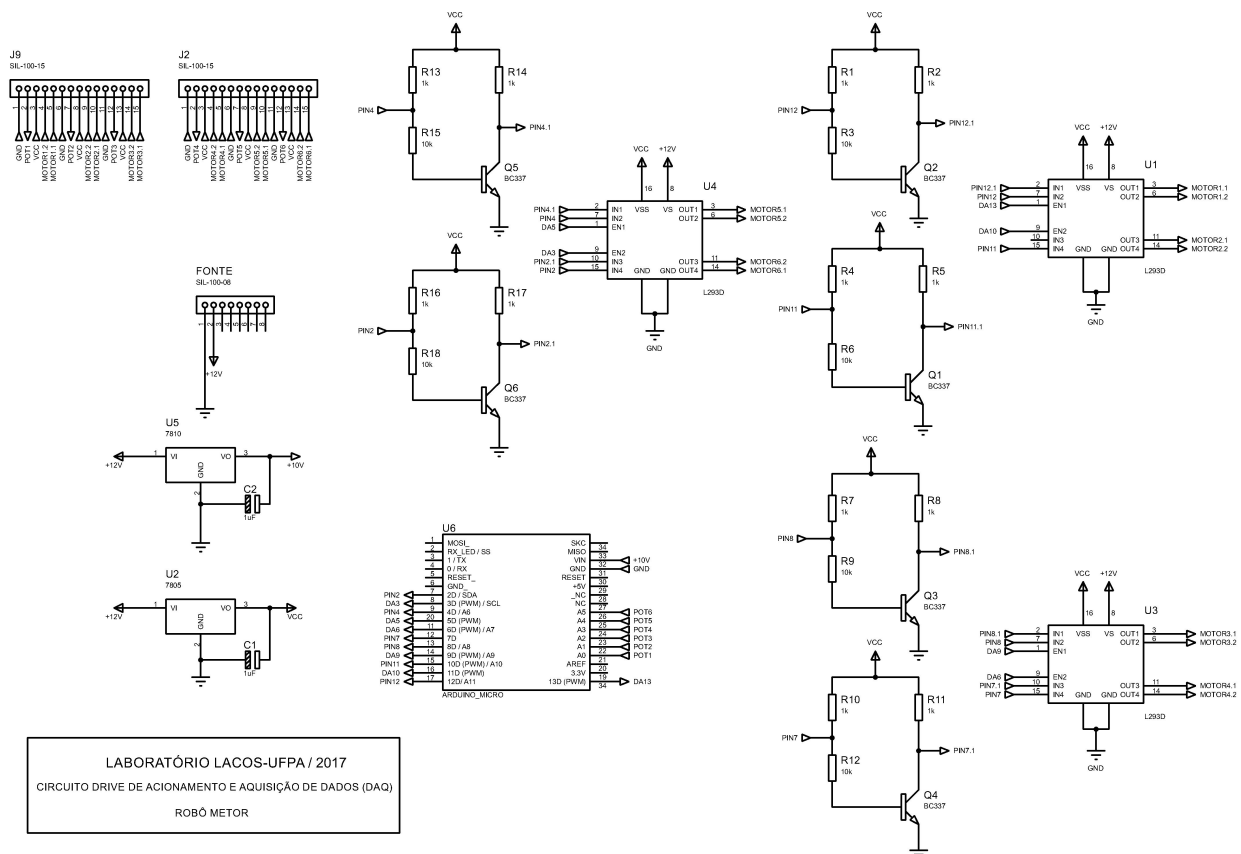
WOOD, R.; BERRY, M. Terminal composition control of a binary distillation column. **Chemical Engineering Science**, Elsevier, v. 28, n. 9, p. 1707–1717, 1973. Citado 7 vezes nas páginas 15, 30, 34, 77, 88, 89 e 90.

YAKUB, F.; MORI, Y. Model predictive control based on kautz function for autonomous ground vehicle path following control application. In: IEEE. **SICE Annual Conference (SICE), 2014 Proceedings of the**. [S.l.], 2014. p. 1035–1040. Citado na página 49.

YANG, J.; ZHANG, P.; LIU, X. Generalized predictive control of PMSM. In: IEEE. **Control and Decision Conference (CCDC), 2013 25th Chinese**. [S.l.], 2013. p. 816–820. Citado na página 76.

YOO, D. K.; WANG, L. Constrained predictive control of three-phase voltage source converters with explicit performance specifications. In: IEEE. **IECON 2012-38th Annual Conference on IEEE Industrial Electronics Society**. [S.l.], 2012. p. 5030–5036. Citado na página 32.

CIRCUITO DRIVE DE ACIONAMENTO E AQUISIÇÃO DE DADOS



CÓDIGO FONTE ARDUINO

Código-fonte 1: DAQDuino MIMO

```
1 // DAQ_Duino for Matlab (Arduino side code)
2 // Author: Antonio Silveira (asilveira@ufpa.br) and Tarcisio Pinheiro
  (tpinheiro@ufpa.br)
3
4 // Description: this is a very simple code to handle data acquisition
  essays
5 //           using Matlab and Arduino. The Matlab side code is
  required
6 //           in order to send I/O requests to this Arduino side.
7
8 //Motor 1
9 float u1[1]={0.0};
10 float y1[1]={0.0};
11 #define DA13 (13) // PWM Pin 13
12 const int Direction_motor1 = 12; // Pin 12 to direction
13 //Analog Read Pin A0
14
15 //Motor 2
16 float u2[1]={0.0};
17 float y2[1]={0.0};
18 #define DA11 (11) // PWM Pin 11
19 const int Direction_motor2 = 10; // Pin 10 to direction
20 //Analog Read Pin A1
21
22 //Motor 3
23 float u3[1]={0.0};
24 float y3[1]={0.0};
25 #define DA9 (9) // PWM Pin 9
26 const int Direction_motor3 = 8; // Pin 8 to direction
```

```
27 //Analog Read Pin A2
28
29 //Motor 4
30 float u4 [1]={0.0};
31 float y4 [1]={0.0};
32 #define DA6 (6) // PWM Pin 6
33 const int Direction_motor4 = 7; // Pin 7 to direction
34 //Analog Read Pin A3
35
36 //Motor 5
37 float u5 [1]={0.0};
38 float y5 [1]={0.0};
39 #define DA5 (5) // PWM Pin 5
40 const int Direction_motor5 = 4; // Pin 4 to direction
41 //Analog Read Pin A4
42
43 //Motor 6
44 float u6 [1]={0.0};
45 float y6 [1]={0.0};
46 #define DA3 (3) // PWM Pin 3
47 const int Direction_motor6 = 2;
48 //Analog Read Pin A5
49
50 void setup() {
51     //Motor 1
52     pinMode(Direction_motor1, OUTPUT);
53     analogWrite(DA13,0);
54
55     //motor 2
56     pinMode(Direction_motor2, OUTPUT);
57     analogWrite(DA11,0);
58
59     //motor 3
60     pinMode(Direction_motor3, OUTPUT);
61     analogWrite(DA9,0);
62
63     //motor 4
64     pinMode(Direction_motor4, OUTPUT);
65     analogWrite(DA6,0);
66
67     //motor 5
68     pinMode(Direction_motor5, OUTPUT);
69     analogWrite(DA5,0);
70
71     //motor 6
72     pinMode(Direction_motor6, OUTPUT);
73     analogWrite(DA3,0);
```

```
74
75
76 Serial.begin(9600);
77 Serial.setTimeout(5); // Maximum miliseconds to wait for
78                        // Serial.parseFloat to timeout
79 }
80
81 void loop() {
82
83   if (Serial.available() > 0)
84   {
85     // Read signal from MATLAB
86     u1[0] = Serial.parseFloat();
87     u2[0] = Serial.parseFloat();
88     u3[0] = Serial.parseFloat();
89     u4[0] = Serial.parseFloat();
90     u5[0] = Serial.parseFloat();
91     u6[0] = Serial.parseFloat();
92
93     //Motor_1
94
95     if (u1[0] > 0)
96       { // Check which direction to go.
97         digitalWrite (Direction_motor1,LOW); //sentido horario
98       }
99     else
100      {
101        digitalWrite (Direction_motor1,HIGH); //sentido anti-
horario
102      }
103
104      u1 [0]=abs(u1 [0]);
105      if (abs(u1 [0])>5)
106      {
107        u1[0]=5;
108      }
109
110      u1 [0] = u1 [0]*(255/5.0);
111      analogWrite(DA13,u1 [0]);
112
113      y1 [0] = analogRead(A0)*(5/1023.0); // Pin A0 as Input
114      if (y1 [0]<0) {y1 [0]=0;}
115      if (y1 [0]>5) {y1 [0]=5;}
116
117     //Motor 2
118
119     if (u2 [0] > 0)
```

```
120         { // Check which direction to go.
121             digitalWrite (Direction_motor2,LOW); //sentido horario
122         }
123     else
124     {
125         digitalWrite (Direction_motor2,HIGH);//sentido anti-
horario
126     }
127
128     u2[0]=abs(u2[0]);
129     if (abs(u2[0])>5)
130     {
131         u2[0]=5;
132     }
133
134     u2[0] = u2[0]*(255/5.0);
135     analogWrite(DA11,u2[0]);
136     y2[0] = analogRead(A1)*(5/1023.0); // Pin A1 as Input
137     if (y2[0]<0) {y2[0]=0;}
138     if (y2[0]>5) {y2[0]=5;}
139
140 //Motor_3
141
142     if (u3[0] > 0)
143     { // Check which direction to go.
144         digitalWrite (Direction_motor3,LOW); //sentido horario
145     }
146     else
147     {
148         digitalWrite (Direction_motor3,HIGH);//sentido anti-
horario
149     }
150
151     u3[0]=abs(u3[0]);
152     if (abs(u3[0])>5)
153     {
154         u3[0]=5;
155     }
156
157     u3[0] = u3[0]*(255/5.0);
158     analogWrite(DA9,u3[0]);
159     y3[0] = analogRead(A2)*(5/1023.0); // Pin A2 as Input
160     if (y3[0]<0) {y3[0]=0;}
161     if (y3[0]>5) {y3[0]=5;}
162
163 //Motor_4
164
```

```
165         if (u4[0] > 0)
166             { // Check which direction to go.
167                 digitalWrite (Direction_motor4,HIGH); //sentido horario
168             }
169         else
170             {
171                 digitalWrite (Direction_motor4,LOW); //sentido anti-
horario
172             }
173
174         u4[0]=abs(u4[0]);
175         if (abs(u4[0])>5)
176             {
177                 u4[0]=5;
178             }
179
180         u4[0] = u4[0]*(255/5.0);
181         analogWrite(DA6,u4[0]);
182         y4[0] = analogRead(A3)*(5/1023.0); // Pin A3 as Input
183         if (y4[0]<0) {y4[0]=0;}
184         if (y4[0]>5) {y4[0]=5;}
185
186 //Motor_5
187
188         if (u5[0] > 0)
189             { // Check which direction to go.
190                 digitalWrite (Direction_motor5,LOW); //sentido horario
191             }
192         else
193             {
194                 digitalWrite (Direction_motor5,HIGH); //sentido anti-
horario
195             }
196
197         u5[0]=abs(u5[0]);
198         if (abs(u5[0])>5)
199             {
200                 u5[0]=5;
201             }
202
203         u5[0] = u5[0]*(255/5.0);
204         analogWrite(DA5,0);
205         y5[0] = analogRead(A4)*(5/1023.0); // Pin A4 as Input
206         if (y5[0]<0) {y5[0]=0;}
207         if (y5[0]>5) {y5[0]=5;}
208
209 //Motor_6
```

```
210
211     if (u6[0] > 0)
212         { // Check which direction to go.
213             digitalWrite (Direction_motor6,HIGH); //sentido horario
214         }
215     else
216     {
217         digitalWrite (Direction_motor6,LOW); //sentido anti-
horario
218     }
219
220     u6[0]=abs(u6[0]);
221     if (abs(u6[0])>5)
222     {
223         u6[0]=5;
224     }
225
226     u6[0] = u6[0]*(255/5.0);
227     analogWrite(DA3,0);
228     y6[0] = analogRead(A5)*(5/1023.0); // Pin A0 as Input
229     if (y6[0]<0) {y6[0]=0;}
230     if (y6[0]>5) {y6[0]=5;}
231
232     Serial.print(y1[0]);
233     Serial.print(y2[0]);
234     Serial.print(y3[0]);
235     Serial.print(y4[0]);
236     Serial.print(y5[0]);
237     Serial.print(y6[0]);
238
239 }
240 }
```

CÓDIGOS DO DMPC CLÁSSICO

Código-fonte 2: DMPC clássico ganhos

```
1
2 % Eg. Tarcisio Pinheiro (13/01/2017)
3 % F é matriz da equação 2.15
4 % Phi é matriz da equação 2.16
5 % Rs é o vetor coluna da equação 2.18
6 % Aa Ba Ca são as matrizes aumentadas
7 function [Phi_Phi, Phi_F, Phi_Rs, BarR, Aa, Ba, Ca] = mpc_classico_ganhos(
    A, B, C, Nc, Np, Gamma);
8
9 % Calcular as modelo aumentado no espaço de estados Aa Ba Ca
10 [m1, n1] = size(C);
11 [n1, n_in] = size(B);
12
13 Aa = eye(n1+m1, n1+m1);
14 Aa(1:n1, 1:n1) = A;
15 Aa(n1+1:n1+m1, 1:n1) = C*A;
16
17 Ba = zeros(n1+m1, n_in);
18 Ba(1:n1, :) = B;
19 Ba(n1+1:n1+m1, :) = C*B;
20
21 Ca = zeros(m1, n1+m1);
22 Ca(:, n1+1:n1+m1) = eye(m1, m1);
23
24
25 %%
26 if n_in == 1 % Caso SISO
27
28     n = n1+m1;
```

```

29     h(1,:) = Ca;
30     F(1,:) = Ca*Aa;
31
32     for kk=2:Np
33         h(kk,:) = h(kk-1, :)*Aa;
34         F(kk,:) = F(kk-1, :)*Aa;
35     end
36
37     v=h*Ba;
38     Phi=zeros(Np,Nc);
39     Phi(:,1)=v; %
40     for i=2:Nc
41         Phi(:,i)=[zeros(i-1,1);v(1:Np-i+1,1)];% Matriz Toeplitz
42     end
43
44     BarR=Gamma*eye(Nc*n_in,Nc*n_in);
45     Rs=ones(n_in*Np,n_in);
46     Phi_Rs=Phi'*Rs;
47     Phi_Phi=Phi'*Phi;
48     Phi_F=Phi'*F;
49
50 else %Caso MIMO
51
52     h(:, :) = Ca;
53     F(:, :) = Ca*Aa;
54
55     b=1;
56     a=4;
57     for kk=2:Np
58         h(a:(n_in+a-1), :) = h(b:(n_in+b-1), :)*Aa;
59         F(a:(n_in+a-1), :) = F(b:(n_in+b-1), :)*Aa;
60         b=a;
61         a=n_in+a;
62     end
63
64     %Calcular os ganhos
65     v=h*Ba;
66     Phi=zeros(Np*n_in,Nc*n_in); %declare the dimension of Phi
67     Phi(:,1:n_in)=v;% first column of Phi
68     a=4;
69
70     for i=2:Nc
71         Phi(:,a:n_in+a-1)=[zeros(i-1,n_in);v(1:(Np*n_in)-i+1,1:n_in)]
; % Matriz Toeplitz
72         a=n_in+a;
73     end
74

```

```

75     Phi_Phi=Phi'*Phi;
76     Phi_F=Phi'*F;
77     BarR=Gamma*eye(Nc*n_in,Nc*n_in);
78     [L,N]=size(Phi_F);
79     Phi_Rs=Phi_F(:,N-2:N);
80
81 end

```

Código-fonte 3: : Controle DMPC clássico

```

1
2 % Eg.Tarcisio Pinheiro (13/01/2017)
3 % opc=1, sem saturação
4 % opc=2, com saturação
5 % opc=3, com saturação modificada
6 function [Delta_u,u]= controle_mpc_classico(
7     Deltau_max,Deltau_min,u_max,...
8
9     u_min,Phi_Phi,BarR,Phi_R,Phi_F,xa,u,yr,W,k,opc);
10
11 if opc==1
12     Delta_U=(inv(Phi_Phi+BarR)*(Phi_R*yr(:,k)-Phi_F*xa));
13     Delta_u=W*Delta_U;
14     u(:,k)=u(:,k-1)+Delta_u;
15 end
16 %%
17 if opc==2 % Saturação
18     Delta_U=(inv(Phi_Phi+BarR)*(Phi_R*yr(:,k)-Phi_F*xa));
19     Delta_u=W*Delta_U;
20     u(:,k)=u(:,k-1)+Delta_u;
21
22 for i=1:length(Delta_u)
23     if Delta_u(i,1)>Deltau_max(i)
24         Delta_u(i,1)=Deltau_max(i);
25     end
26     if Delta_u(i,1)<Deltau_min(i)
27         Delta_u(i,1)=Deltau_min(i);
28     end
29 end
30
31 u(:,k)=u(:,k-1)+Delta_u;
32
33 for i=1:length(Delta_u)
34     if u(i,k)>u_max(i)

```

```

36         u(i,k)=u_max(i);
37     end
38
39     if u(i,k)<u_min(i)
40         u(i,k)=u_min(i);
41     end
42 end
43
44 end
45
46 %%
47 if opc==3 % Saturação modificada
48     Delta_U=(inv(Phi_Phi+BarR)*(Phi_R*yr(:,k)-Phi_F*xa));
49     Delta_u=W*Delta_U;
50     u_=u(:,k-1);
51     u(:,k)=u(:,k-1)+Delta_u;
52
53     for i=1:length(u_max)
54
55         if u(i,k)>u_max(i)
56             u(i,k)=u_max(i);
57             Delta_u(i)=u_max(i)-u_(i);
58
59         elseif u(i,k)<u_min(i)
60             u(i,k)=u_min(i);
61             Delta_u(i)=u_min(i)-u_(i);
62         end
63
64     end
65
66 end

```

Código-fonte 4: Modelo discreto Mentor

```

1 % Modelo discreto indentificado do robô manipulador Mentor
2 % Sistema MIMO 3x3
3 % Período de Amostragem de Ts=0.1 segundo
4
5 function Gz= planta_Mentor(Ts);
6 %First column %joint 1
7 Bz11=[-0.00072717 -0.0039167 ];
8 Az11=[1 -1.6042 0.60415];
9 Raiz=roots(Az11);
10 Az11=poly([1 Raiz(2)]);
11 Gz11 = tf(Bz11,Az11,Ts);
12
13 Bz21 = [0 -0.0000000011382 0.000000000076199];

```

```

14 Az21 = [1 -0.49889 -0.49825];
15 Gz21 = tf(Bz21,Az21,Ts);
16
17 Bz31 = [0 -0.00000000017509 -0.00000000014624];
18 Az31 = [1 -0.49889 -0.49825];
19 Gz31 = tf(Bz31,Az31,Ts);
20 %% %Second column %joint 2
21 Bz12 = [0 -0.0000000052074 -0.00000000034674 ];
22 Az12 = [1 -0.28848 -0.70734];
23 Gz12 = tf(Bz12,Az12,Ts);
24
25 Bz22=[0.00023728 -0.0055177 ];
26 Az22=[1 -1.5338 0.53391];
27 Raiz=roots(Az22);
28 A22=poly([1 Raiz(2)]);
29 Gz22 = tf(Bz22,Az22,Ts);
30
31 Bz32 = [0 0.00000030546 -0.00000011658];
32 Az32 = [1 -0.28848 -0.70734];
33 Gz32 = tf(Bz32,Az32,Ts);
34 %% %third column %joint 3
35 Bz13 = [0 5.6358e-09 -5.5471e-09];
36 Az13 = [1 -0.5 -0.5 ];
37 Gz13 = tf(Bz13,Az13,Ts);
38
39 Bz23 = [0 -1.9662e-05 9.6277e-05];
40 Az23 = [1 -0.58558 -0.41422 ];
41 Gz23 = tf(Bz23,Az23,Ts);
42
43 Bz33=[-0.0022775 -0.010483];
44 Az33=[1 -1.535 0.53697];
45 Raiz=roots(Az33);
46 Az33=poly([1 Raiz(2)]);
47 Gz33 = tf(Bz33,Az33,Ts);
48 %%
49 Gz=[Gz11 Gz12 Gz13;
50      Gz21 Gz22 Gz23;
51      Gz31 Gz32 Gz33];

```

Código-fonte 5: Exemplo SISO: Simulação DMPC clássico

```

1
2 clc
3 clear all
4 close all
5 %%
6 Ts=0.1;% Período de amostragem

```

```

7 N_sim=100;% número de interações
8 %%
9 Np=100;% Horizonte de predição
10 Nc=2;% Horizonte de controle
11 Gamma=0.0001; % Esforço de controle
12 %% Referência (Set-point)
13 yr(1,1:N_sim)=4;
14 %% Modelo discreto da Planta
15 Bz11=[-0.00072717 -0.0039167];% B(z^-1)
16 Az11=[1 -1.6042 0.60415];% A(z^-1)
17 Gz11 = tf(Bz11,Az11,Ts);% G(z^-1)
18 %%
19 Gz11=ss(Gz11,'min');% criar o modelo no espaço de estados
20 [A,B,C,D,Ts]=ssdata(Gz11);% Para obter as matrizes A B C
21 %% número de entradas e saídas do sistema
22 [n_out,n1]=size(C);
23 [n_xm,n_in]=size(B);
24 %%
25 [Phi_Phi,Phi_F,Phi_Rs,BarR,Aa,Ba,Ca]= mpc_classico_ganhos(
        A,B,C,Nc,Np,Gamma);
26 %% Inicialização das variáveis
27 x=ones(n1,1);% variáveis de estado
28 xa=[x;0];% estados aumentados da planta
29 u=zeros(n_in,1);% sinal de controle
30 y=zeros(n_out,1);% saída do sistema
31 %%
32 I=eye(n_in,n_in);
33 W=zeros(n_in,Nc*n_in);
34 W(1:n_in,1:n_in)=I;
35 %% Malha fechada
36 Kmpc_C=W*inv(Phi_Phi+BarR)*Phi_F;
37 eigen_C=Aa-Ba*Kmpc_C;
38 Ky=W*inv(Phi_Phi+BarR)*Phi_Rs;
39 sys_C=ss(Aa-Ba*Kmpc_C,Ba*Ky,Ca,D,Ts);% sistema de malha fechada
40 %% Restrições impostas ao sistema
41 Deltau_max=2;
42 Deltau_min=-2;
43 u_max=5;
44 u_min=-5;
45 %% Opção de controle
46 opc=1;
47 %%
48 for k=2:N_sim;
49
50 [Delta_u,u]= controle_mpc_classico(Deltau_max,Deltau_min,u_max,...
51                                     u_min,Phi_Phi,BarR,Phi_Rs,Phi_F,xa,u,yr,W,k,opc);
52

```

```

53     x_old=x;
54     x=A*x+B*u(:,k);
55     y(:,k)=C*x_old;
56     xa=[x-x_old;y(:,k)];
57     du(:,k)=Delta_u;
58
59 end
60
61 k=0:Ts:(N_sim)*Ts-Ts;
62 figure (1)
63 subplot(311)
64 plot(k,y,'b')
65 legend('saída')
66 ylabel('Volts')
67 grid on
68 subplot(312)
69 plot(k,u,'r')
70 xlabel('Instante de tempo')
71 legend('Sinal de controle')
72 ylabel('Volts')
73 grid on
74 subplot(313)
75 plot(k,du,'r')
76 xlabel('Instante de tempo')
77 legend('Sinal de controle incremental')
78 ylabel('\DeltaVolts')
79 grid on
80 %% Polos de malha fechada
81 figure(2)
82 plot(eig(eigen_C),'b*')
83 title('Polos de malha fechada')
84 grid on;

```

Código-fonte 6: Exemplo MIMO: Simulação DMPC clássico

```

1
2 % Eng. Tarcisio Pinheiro (13/12/2017)
3 % DMPC clássico , exemplo MIMO 3x3
4 clear all
5 close all
6 clc
7 format long
8 %%
9 Ts=0.1;
10 N_sim=50;%tamanho do vetor
11 %%
12 Gz=planta_Mentor(Ts);

```

```

13 Gzmin=ss(Gz,'min');% Para criar o modelo no espaço de estados
14 [A,B,C,D,Ts]=ssdata(Gzmin);% Para obter as matrizes A B C
15 %%
16 Np=100;%Prediction Horizon
17 Nc=10;%Control Horizon
18 Gamma=0.00001;% Esforço de controle
19 %%
20 [Phi_Phi,Phi_F,Phi_R,BarR,Aa,Ba,Ca]= mpc_classico_ganhos(
        A,B,C,Nc,Np,Gamma);
21 %% Referência (Set-point)
22 yr1(1,1:N_sim)=1;
23 yr2(1,1:N_sim)=2;
24 yr3(1,1:N_sim)=3;
25 yr=[yr1;yr2;yr3];
26 %% Restrições impostas ao sistema
27 Deltau_max=[2.5 2.5 2.5];
28 Deltau_min=[-2 -2 -2];
29 u_max=[10 10 10];
30 u_min=[-10 -10 -10];
31 %%
32 [n1,n_in]=size(B);
33 [n_out,n1]=size(C);
34 %%
35 u=zeros(n_out,1);
36 y=zeros(n_out,1);
37 x=zeros(n1,1);
38 xa=[x;y];
39 %%
40 I=eye(n_out,n_out);
41 W=zeros(n_out,Nc*n_out);
42 W(1:n_out,1:n_out)=I;
43 %% Malha fechada
44 Kmpc_C=W*inv(Phi_Phi+BarR)*Phi_F;
45 eigen_C=Aa-Ba*Kmpc_C;
46 Ky=W*inv(Phi_Phi+BarR)*Phi_R;
47 sys_C=ss(Aa-Ba*Kmpc_C,Ba*Ky,Ca,D,Ts);% sistema de malha fechada
48 %%
49 opc=1;
50 %%
51 for k=2:N_sim;
52
53 [Delta_u,u]= controle_mpc_classico(Deltau_max,Deltau_min,u_max,...
54         u_min,Phi_Phi,BarR,Phi_R,Phi_F,xa,u,yr,W,k,opc);
55
56     x_old=x;
57     x=A*x+B*u(:,k);
58     y(:,k)=C*x_old;

```

```
59     xa=[x-x_old;y(:,k)];
60     du(:,k)=Delta_u;
61 end
62 %%
63 k=0:Ts:(N_sim)*Ts-Ts;
64 figure (1)
65 subplot(311)
66 plot(k,y(1,:), 'r',k,y(2,:), 'b',k,y(3,:), 'm')
67 legend('saida1', 'saida2', 'saida3')
68 ylabel('Volts')
69 grid on
70 %%
71 subplot(312)
72 plot(k,u(1,:), 'r',k,u(2,:), 'b',k,u(3,:), 'm')
73 ylabel('Volts')
74 legend('entrada_1', 'entrada_2', 'entrada_3')
75 grid on
76 %%
77 subplot(313)
78 plot(k,du(1,:), 'r',k,du(2,:), 'b',k,du(3,:), 'm')
79 xlabel('Instante de tempo')
80 ylabel('\DeltaVolts')
81 legend('Sinal de controle incremental')
82 grid on
83 %% Polos de malha fechada
84 figure(2)
85 plot(eig(eigen_C), 'b*')
86 grid on;
```

CÓDIGOS DO DMPC USANDO FUNÇÕES DE LAGUERRE

Código-fonte 7: Função lagd

```

1
2 % Função retirada do Livro Model Predictive Control System Design and
   Implementation
3 % Contribuição de Liuping Wang
4 % Neta função é calculado: a matriz M de 4.26 relacionandas com o
   sinal% incremetal de controle.
5 % L0 da equação 3.6
6 % A1 da equação 3.9
7
8
9
10
11 function [A1,L0]=lagd(a,N)
12 %a and N are for the Laguerre functions
13 %n_in is the number of inputs
14 v(1,1)=a;
15 L0(1,1)=1;
16 for k=2:N
17 v(k,1)=(-a).^ (k-2)*(1-a*a);
18 L0(k,1)=(-a).^ (k-1);
19 end
20 L0=sqrt((1-a*a))*L0;
21 A1(:,1)=v;
22 for i=2:N
23 A1(:,i)=[zeros(i-1,1);v(1:N-i+1,1)];
24 end

```

Código-fonte 8: Função dmPC

```

1
2 % Função retirada do Livro Model Predictive Control System Design and
   Implementation
3 % Contribuição de Liuping Wang
4 % Neta função é calculado: Omega da equação 3.41, Psi da equação 3.42
5
6 function [Omega,Psi]=dmPC(Aa,Ba,a,N,Np,Q,R);
7
8 [n,n_in]=size(Ba);
9 N_pa=sum(N); %comprimento de eta
10 E=zeros(N_pa,N_pa);
11 H=zeros(N_pa,n);
12 R_para=zeros(N_pa,N_pa);
13 n0=1;
14 ne=N(1);
15
16 for i=1:n_in-1;
17 R_para(n0:ne,n0:ne)= R(i,i)*eye(N(i),N(i));
18 n0=n0+N(i);
19 ne=ne+N(i+1);
20 end
21 R_para(n0:N_pa,n0:N_pa)=R(n_in,n_in)*eye(N(n_in),N(n_in));
22
23 S_in=zeros(n,N_pa);
24 [A1,L0]=lagd(a(1),N(1));
25 S_in(:,1:N(1))=Ba(:,1)*L0';
26 In_s=1;
27
28 for jj=2:n_in;
29 [A1,L0]=lagd(a(jj),N(jj));
30 In_s=N(jj-1)+In_s;
31 In_e=In_s+N(jj)-1;
32 S_in(:,In_s:In_e)=Ba(:,jj)*L0';
33 end
34 S_sum=S_in;
35 phi=S_in;
36 E=(phi)'*Q*(phi);
37 H=phi'*Q*Aa;
38
39 for i=2:Np;
40 Eae=Aa^i;
41
42 [A1,L0]=lagd(a(1),N(1));
43 % Função de Laguerre relacionado com a entrada 1
44 S_sum(:,1:N(1))=Aa*S_sum(:,1:N(1))+S_in(:,1:N(1))*(A1^(i-1))';
45

```

```

46 %%
47 In_s=1;
48 for kk=2:n_in;
49 [A1,L0]=lagd(a(kk),N(kk));
50 In_s=N(kk-1)+In_s;
51 In_e=In_s+N(kk)-1;
52 S_sum(:,In_s:In_e)=Aa*S_sum(:,In_s:In_e)+ S_in(:,In_s:In_e)*(A1^(i-1)
    )';
53 end
54 phi=S_sum;
55 E=E+phi'*Q*phi;
56 H=H+phi'*Q*Eae;
57 end
58 E=E+R_para;
59 phi=S_sum;
60 %%
61 Omega=E;
62 Psi=H;

```

Código-fonte 9: Função Mdu

```

1
2 % Função retirada do Livro Model Predictive Control System Design and
    Implementation
3 % Contribuição de Liuping Wang
4 % Neta função é calculado: a matriz M de 4.25 relacionandas com o
    sinal incremental de controle.
5 % Lzerot da equação 3.7
6
7 function [M,Lzerot]=Mdu(a,N,n_in,Nc)
8 %a and N are for the Laguerre functions
9 %n_in is the number of inputs
10 %Nc is the number of constraints
11
12 N_pa=sum(N);
13 M=zeros(n_in,N_pa);
14
15 M_du1=zeros(n_in,N_pa);
16 k0=1;
17 [A1,L0]=lagd(a(k0),N(k0));
18 M_du1(1,1:N(1))=L0';
19 cc=N(1);
20
21 for k0=2:n_in;
22 [A1,L0]=lagd(a(k0),N(k0));
23 M_du1(k0,cc+1:cc+N(k0))=L0';
24 cc=cc+N(k0);

```

```

25 end
26
27 Lzerot=M_du1;
28 M=M_du1;
29
30 for kk=2:Nc
31 k0=1;
32 [A1,L0]=lagd(a(k0),N(k0));
33 L=A1^(kk-1)*L0;
34 M_du1(1,1:N(1))=L';
35 cc=N(1);
36 for k0=2:n_in;
37 [A1,L0]=lagd(a(k0),N(k0));
38 L=A1^(kk-1)*L0;
39 M_du1(k0,cc+1:cc+N(k0))=L';
40 cc=cc+N(k0);
41 end
42 M=[M;M_du1];
43 end

```

Código-fonte 10: Função Mu

```

1
2
3 % Função retirada do Livro Model Predictive Control System Design and
   Implementation
4 % Contribuição de Liuping Wang
5 % Neta função é calculado: a matriz M de 4.26 relacionandas com o
   sinal de controle.
6
7
8
9 function M=Mu(a,N,n_in,Nc)
10 %a and N are for the Laguerre functions
11 %n_in is the number of inputs
12 %Nc is the number of constraints
13 % the matrix M used in the constraint of the control signal
14 N_pa=sum(N);
15 M=zeros(n_in,N_pa);
16 M_du1=zeros(n_in,N_pa);
17 k0=1;
18
19
20 [A1,L0]=lagd(a(k0),N(k0));
21 M_du1(1,1:N(1))=L0';
22 cc=N(1);
23

```

```

24
25 for k0=2:n_in;
26 [A1,LO]=lagd(a(k0),N(k0));
27 M_du1(k0,cc+1:cc+N(k0))=L0';
28 cc=cc+N(k0);
29 end
30
31 M=M_du1;
32 Ms=M_du1;
33
34 for kk=2:Nc
35 k0=1;
36 [A1,LO]=lagd(a(k0),N(k0));
37 L=A1^(kk-1)*L0;
38 M_du1(1,1:N(1))=L';
39 cc=N(1);
40 for k0=2:n_in;
41 [A1,LO]=lagd(a(k0),N(k0));
42 L=A1^(kk-1)*L0;
43 M_du1(k0,cc+1:cc+N(k0))=L';
44 cc=cc+N(k0);
45 end
46 Ms=Ms+M_du1;
47 M=[M;Ms];
48 end

```

Código-fonte 11: Função Programação Quadrática de Hildreth

```

1
2 % Função retirada do Livro Model Predictive Control System Design and
   % Implementation
3 % Contribuição de Liuping Wang
4 % Programação Quadrática de Hildreth
5
6 function eta=QPhild(H,f,A_cons,b);
7 % E=H;
8 % F=f;
9 % M=A_cons;
10 % gamma=b;
11 % eta =x
12
13 %Omega=H
14 %Psi*Xf=f
15 %M=A_cons
16 %gamma=b
17
18 [n1,m1]=size(A_cons);

```

```
19 eta=-H\f;
20 kk=0;
21
22 for i=1:n1
23 if (A_cons(i,:)*eta>b(i))
24     kk=kk+1;
25 else
26 kk=kk+0;
27 end
28 end
29
30 if (kk==0)
31 return;
32 end
33
34 P=A_cons*(H\A_cons');%h
35 d=(A_cons*(H\f)+b);%k
36
37 [n,m]=size(d);
38 x_ini=zeros(n,m);
39 lambda=x_ini;
40 al=10;
41
42 for km=1:100
43 %find the elements in the solution vector one by one
44 % km could be larger if the Lagranger multiplier has a slow
45 % convergence rate.
46 lambda_p=lambda;
47
48 for i=1:n
49 w= P(i,:)*lambda-P(i,i)*lambda(i,1);
50 w=w+d(i,1);
51 la=-w/P(i,i);
52 lambda(i,1)=max(0,la);
53 end
54
55 al=(lambda-lambda_p)'*(lambda-lambda_p);
56
57 if (al<10e-8);
58     break;
59 end
60
61 end
62
63 eta=-H\f -H\A_cons'*lambda;%x
```

Código-fonte 12: Função DMPC Laguerre

```

1
2 % Eg. Tarcisio Pinheiro (13/01/2017)
3 function [Omega, Psi, Lzerot, M, gamma, Aa, Ba, Ca, A, B, C]=mpc_laguerre(
    Gz, Np, R, a, N, Nsum, ...
    Deltau_max, Deltau_min, u_max, u_min);
4
5 Gzmin=ss(Gz, 'min'); % Para criar o modelo no espaço de estados
6 [A, B, C, D, Ts]=ssdata(Gzmin); % Para obter as matrizes A B C
7 %%
8 %Calcular as modelo aumentado no espaço de estados Aa Ba Ca
9 [m1, n1]=size(C);
10 [n1, n_in]=size(B);
11
12 Aa=eye(n1+m1, n1+m1);
13 Aa(1:n1, 1:n1)=A;
14 Aa(n1+1:n1+m1, 1:n1)=C*A;
15
16 Ba=zeros(n1+m1, n_in);
17 Ba(1:n1, :)=B;
18 Ba(n1+1:n1+m1, :)=C*B;
19
20 Ca=zeros(m1, n1+m1);
21 Ca(:, n1+1:n1+m1)=eye(m1, m1);
22
23 Q=Ca'*Ca; %Q matriz peso
24 %%
25 [Omega, Psi]=dmpe(Aa, Ba, a, N, Np, Q, R);
26 %% Matrizes de restricao
27 M0=Mu(a, N, n_in, Nsum);
28 [M1, Lzerot]=Mdu(a, N, n_in, Nsum);
29 %%
30 M=[M0; -M0; M1; -M1];
31 k=1;
32 s=n_in;
33
34 for kk=1:Nsum
35     gamma0(k:s, 1)=(u_max);
36     gamma1(k:s, 1)=(-u_min);
37     gamma2(k:s, 1)=Deltau_max;
38     gamma3(k:s, 1)=-Deltau_min;
39     k=k+n_in;
40     s=s+n_in;
41 end
42 gamma=[gamma0; gamma1; gamma2; gamma3];

```

Código-fonte 13: Função Controle DMPC Laguerre

```

1 % Eng. Tarcisio Pinheiro (13/12/2017)
2 function [Delta_u,u,gamma]=controle_mpc_laguerre(
    Nsum,gamma,M,Omega,Psi,Lzerot,xa,deltau_max,deltau_min,u_max,u_min,u,a
    );
3
4 if a==1 % Sem saturação
5 Kmpc=Lzerot*(Omega\Psi);
6 Delta_u=-Kmpc*xa;
7 u=u+Delta_u;
8 end
9 %%
10 if a==2 % com Saturação
11
12     Kmpc=Lzerot*(Omega\Psi);
13     Delta_u=-Kmpc*xa;
14     for i=1:length(u_max)
15
16         if Delta_u(i,1)>deltau_max(i)
17             Delta_u(i,1)=deltau_max(i);
18         end
19
20         if Delta_u(i,1)<deltau_min(i)
21             Delta_u(i,1)=deltau_min(i);
22         end
23     end
24     u=u+Delta_u;
25     for i=1:length(u_max)
26
27         if u(i,1)>u_max(i)
28             u(i,1)=u_max(i);
29         end
30         if u(i,1)<u_min(i)
31             u(i,1)=u_min(i);
32         end
33     end
34 end
35 %%
36 if a==3 % Restrição com Programação Quadrática de Hildreth
37     eta=QPhild(Omega,Psi*xa,M,gamma);
38     Delta_u=Lzerot*eta;
39     u=u+Delta_u;
40
41     kk=1;
42     s=length(Delta_u);
43
44     for w=1:Nsum

```

```

45     gamma0(kk:s,1)=(u_max-u');
46     gamma1(kk:s,1)=(-u_min+u');
47     kk=kk+length(Delta_u);
48     s=s+length(Delta_u);
49     end
50
51
52     gamma(1:length(Delta_u)*Nsum,1)=gamma0;
53     gamma(length(Delta_u)*Nsum+1:2*length(Delta_u)*Nsum,1)=gamma1;
54 end

```

Código-fonte 14: Exemplo SISO: Simulação DMPC usando Funções de Laguerre

```

1 % Eng. Tarcisio Pinheiro (13/12/2017)
2 % DMPC usando funções de Laguerre , exemplo SISO
3 clc
4 clear all
5 close all
6 %%
7 Ts=0.1;% Período de amostragem
8 N_sim=400;% número de interações
9 %% Parametros de sintonia
10 Np=100;% Horizonte de predição
11 Gamma=0.001; % Esforço de controle
12 N=2;% Ordem das funções de Laguerre
13 Nsum=sum(N);
14 a=0.4;% Polos das funções de Laguerre
15 %% Restrições impostas ao sistema
16 Deltau_max=2;
17 Deltau_min=-2;
18 u_max=5;
19 u_min=-5;
20 %% Referência (Set-point)
21 yr(1,1:2)=0;
22 yr(1,3:N_sim)=4;
23 %% Modelo discreto da Planta
24 Bz11=[-0.00072717 -0.0039167];% B(z^-1)
25 Az11=[1 -1.6042 0.60415];% A(z^-1)
26 Gz11 = tf(Bz11,Az11,Ts);% G(z^-1)
27 %%
28 [Omega,Psi,Lzerot,M,gamma,Aa,Ba,Ca,A,B,C]=mpc_laguerre(
    Gz11,Np,Gamma,a,N,Nsum,...
29     Deltau_max,Deltau_min,u_max,u_min);
30 %%
31 [n_out,n1]=size(C);
32 [n_xm,n_in]=size(B);
33 %% Analise em malha fechada

```

```

34 Kmpc_L=Lzerot*(Omega\Psi);
35 eigen_L=Aa-Ba*Kmpc_L;
36 sys_L=ss(Aa-Ba*Kmpc_L,Ba,Ca,0,Ts);% sistema em malha fechada
37 %% startup variables
38 u=zeros(n_in,1); % sinal de controle
39 y=zeros(n_out,1);% saída
40 %x=ones(n1,1);% estados
41 x=1.0e-23*[0.223888205729579;0.390471883227424];
42 xa=[x;(y(:,1)-yr(:,1))];% estados aumentados
43 %%
44 opc=3;
45 %%
46 for k=1:N_sim;
47
48     [Delta_u,u,gamma]=controle_mpc_laguerre(
49         Nsum,gamma,M,Omega,Psi,Lzerot,xa,...
50         Deltau_max,Deltau_min,u_max,u_min,u,opc);
51
52     x_old=x;
53     x=A*x+B*u;
54     y(1:n_out,k)=C*x_old;
55
56     xa=[x-x_old;(y(1:n_out,k)-yr(:,k))];
57
58     du_L(1:n_in,k)=Delta_u;
59     u_L(1:n_in,k)=u;
60 end
61
62 % Plotar Gráficos
63 k=0:Ts:N_sim*Ts-Ts;
64 figure(1)
65 subplot(3,1,1)
66 plot(k,y,'r')
67 legend('saída')
68 ylabel('Volts')
69 grid on
70 %%
71 subplot(3,1,2)
72 plot(k,u_L(1,1:length(k)),'r')
73 ylabel('Volts')
74 legend('sinal de controle')
75 grid on
76 %%
77 subplot(3,1,3)
78 plot(k,du_L(1,1:length(k)),'r')
79 ylabel('\DeltaVolts')

```

```

80 legend('sinal de controle incremental')
81 xlabel('Instante de tempo')
82 grid on
83 %% Polos de malha fechada
84 figure(2)
85 plot(eig(eigen_L),'b*')
86 grid on;

```

Código-fonte 15: Exemplo MIMO: Simulação DMPC usando Funções de Laguerre

```

1
2 % Eng. Tarcisio Pinheiro (13/12/2017)
3 % DMPC usando funções de Laguerre , exemplo MIMO 3x3
4 clear all
5 close all
6 clc;
7 format long
8 %%
9 Ts=0.1;
10 N_sim=70;%tamanho do vetor
11 %% Referência (Set-point)
12 yr1(1,1:N_sim)=1;
13 yr2(1,1:N_sim)=2;
14 yr3(1,1:N_sim)=3;
15 yr=[yr1;yr2;yr3];
16 %% Restricoes no sinal de controle
17 Deltau_min=[-2 -2 -2];
18 Deltau_max=[2 2 2];
19 u_min=[-5 -5 -5];
20 u_max=[5 5 5];
21 %% Parametros de sintonia
22 Np=100;% Horizonte de predição
23 Gamma=diag([0.008 0.008 0.008]); % Esforço de controle
24 N=[2 2 2];% Ordem das funções de Laguerre
25 Nsum=10;
26 a=[0.3 0.2 0.6];% Polos das funções de Laguerre
27 %%
28 Gz=planta_Mentor(Ts);
29 %%
30 [Omega,Psi,Lzerot,M,gamma,Aa,Ba,Ca,A,B,C]=mpc_laguerre(
    Gz,Np,Gamma,a,N,Nsum,...
31     Deltau_max,Deltau_min,u_max,u_min);
32 %%
33 [n_out,n1]=size(C);
34 [n_xm,n_in]=size(B);
35 %% Analise em malha fechada
36 Kmpc_L=Lzerot*(Omega\Psi);

```

```

37 eigen_L=Aa-Ba*Kmpc_L;
38 sys_L=ss(Aa-Ba*Kmpc_L,Ba,Ca,0,Ts);% sistema em malha fechada
39 %% startup variables
40 u=zeros(n_in,1);
41 y=zeros(n_out,1);
42 x=ones(n1,1);
43 xa=[x;(y(:,1)-yr(:,1))];
44 %%
45 opc=3;
46 %%
47
48 for kk=1:N_sim;
49
50     x_old=x;
51     x=A*x+B*u;
52     y(:,kk)=C*x_old;
53     xa=[x-x_old;(y(:,kk)-yr(:,kk))];
54
55     [DeltaU,u,gamma]=controle_mpc_laguerre(
56         Nsum,gamma,M,Omega,Psi,Lzerot,xa,...
57         Deltau_max,Deltau_min,u_max,u_min,u,opc);
58
59     du_L(1:n_in,kk)=DeltaU;
60     u_L(1:n_in,kk)=u;
61 end
62
63 % Plotar Gráficos
64 k=0:Ts:N_sim*Ts-Ts;
65 figure(1)
66 subplot(3,1,1)
67 plot(k,y(1,:), 'r',k,y(2,:), 'b',k,y(3,:))
68 legend('saída_1','saída_2','saída_3')
69 ylabel('Volts')
70 grid on
71 %%
72 subplot(3,1,2)
73 plot(k,u_L(1,:), 'r',k,u_L(2,:), 'b',k,u_L(3,:))
74 legend('entrada_1','entrada_2','entrada_3')
75 ylabel('Volts')
76 grid on
77 %% Increment signal control
78 subplot(3,1,3)
79 plot(k,du_L(1,:), 'r',k,du_L(2,:), 'b',k,du_L(3,:))
80 ylabel('\DeltaVolts')
81 xlabel('Instante de tempo')
82 grid on

```

```
83  %% Polos de malha fechada
84  figure(2)
85  plot(eig(eigen_L), 'b*')
86  grid on;
```
