

Tese apresentada à Divisão de Pós-Graduação do Instituto Tecnológico de Aeronáutica como parte dos requisitos para a obtenção do título de “Mestre em Ciência”, na Área de Sistemas e Controle do Curso de Engenharia Eletrônica e Computação.

Renzo Orchiucci Miura

Controle Preditivo de um Sistema de Levitação Magnética

Tese aprovada em sua versão final pelos abaixo assinados

Prof. Roberto Kawakami Harrop Galvão

Orientador

Prof. Homero Santiago Maciel

Chefe da Divisão de Pós-Graduação

Campus de Montenegro

São José dos Campos, SP – Brasil

2003

INSTITUTO TECNOLÓGICO DE AERONÁUTICA

DIVISÃO DE PÓS-GRADUAÇÃO

12228-900 – São José dos Campos

e-mail:posgrad@ita.cta.br

Controle Preditivo de um Sistema de Levitação Magnética

Renzo Orchiucci Miura

Composição da Banca Examinadora:

Takashi Yoneyama

Presidente-ITA

Roberto Kawakami Harrop Galvão

Orientador-ITA

Karl Heinz Kienitz

Membro-ITA

Alberto Adade Filho

Membro-ITA

Edvaldo Assunção

Membro externo-UNESP

ITA, novembro 2003

SUMÁRIO

Índice de Figuras.....	iii
Índice de Tabelas.....	vi
Lista de Símbolos.....	vii
Lista de Siglas.....	x
Resumo.....	xii
Abstract.....	xiv
 Capítulo 1 – Introdução.....	 1
1.1 Generalidades.....	1
1.2 Objetivo da Tese.....	2
1.3 Apresentação da Tese.....	3
 Capítulo 2 – Controle Preditivo baseado em Modelo.....	 5
2.1 Introdução.....	5
2.2 Histórico do Controle Preditivo baseado em Modelo.....	9
2.2.1 IDCOM.....	11
2.2.2 DMC.....	13
2.2.3 Estatísticas sobre publicações em MPC nos últimos trinta e três anos.....	15
2.3 Formulação MPC.....	16
2.4 Modelos.....	19
2.4.1 Resposta ao Impulso.....	19
2.4.2 Resposta ao Degrau.....	21
2.4.3 Função de Transferência.....	22
2.4.4 Espaço de Estados.....	23
2.5 Estratégia de Controle.....	25
2.5.1 Resposta Finita ao Degrau.....	25
2.5.2 Formulação no Espaço de Estados.....	29
2.6 Tratamento do erro em regime permanente no ambiente MPC.....	30
2.6.1 Introdução.....	30
2.6.2 Modelos Independente e Realinhado.....	31
2.6.3 Rastreamento livre de erro em regime para modelos independentes (sem realinhamento).....	33
2.6.4 Tratamento do Erro em Regime para modelos instáveis em malha aberta (com realinhamento).....	35

2.7 MPC Restrito.....	38
2.7.1 Introdução.....	38
2.7.2 MPC e Restrições.....	38
2.7.2.1 Restrição da Taxa de Variação.....	41
2.7.2.2 Restrição da Amplitude de Entrada.....	42
2.7.2.3 Restrição da Amplitude de Saída.....	43
2.7.2.4 Matriz Restrição.....	44
Capítulo 3 – Estratégia MPC adotada para controle do levitador magnético.....	45
3.1 Introdução.....	45
3.2 Sistema de levitação magnética da Feedback Instruments.....	46
3.3 Linearização do modelo.....	51
3.4 Discretização e formulação do modelo em função da variação do controle.....	53
3.5 Tratamento das Restrições.....	55
3.5.1 Determinação das restrições do SLM.....	55
3.5.2 Matrizes de Restrição.....	59
3.6 Descrição do Algoritmo LMPC.....	61
3.7 Função Objetivo.....	64
Capítulo 4 – Resultados & Discussão.....	66
4.1 Introdução.....	66
4.2 Simulações.....	66
4.2.1 Compensador em Avanço.....	67
4.2.2 Controlador Preditivo.....	71
4.3 Avaliação experimental utilizando simulação analógico.....	87
4.4 Experimentos utilizando o SLM.....	92
4.4.1 Correção do erro em regime.....	92
4.4.2 Rastreamento de uma trajetória de operação utilizando MPC.....	93
Capítulo 5 – Conclusão.....	102
5.1 Conclusões Finais.....	102
5.2 Recomendações para trabalhos futuros.....	104
Referência Bibliográfica.....	106
Anexo A – Ferramentas Computacionais.....	111
A.1 S-Function.....	111
A.2 Quadprog.....	122
A.3 xPC Target.....	123
A.4 DMatrix.....	126

Índice de Figuras

Figura 2.1: Estrutura básica do MPC.....	16
Figura 2.2: Restrições no sinal de controle.....	39
Figura 3.1: Foto ilustrativa do S.L.M da Feedback Instruments.....	47
Figura 3.2: Diagrama de blocos do SLM.....	47
Figura 3.3: Gráfico que relaciona corrente no eletroímã pela força de atração exercida sobre a esfera.....	49
Figura 3.4: Gráfico que relaciona a distância com força de atração exercida sobre a esfera.....	51
Figura 3.5: Diagrama de bloco representativo do driver de corrente.....	56
Figura 3.6: Curva representativa da relação U x I.....	58
Figura 3.7: Gráfico que mostra as taxas de crescimento e decaimento do controle aplicado ao driver de corrente.....	59
Figura 3.8: Fluxograma do algoritmo de LMPC.....	62
Figura 3.9: Função objetivo para esfera a 1.67cm do eletroímã.....	65
Figura 3.10: Função objetivo para esfera a 2.89cm do eletroímã.....	65
Figura 3.11: Função objetivo para esfera a 2.3cm do eletroímã.....	65
Figura 4.1: Diagrama de blocos ilustrativo.....	67
Figura 4.2: Formas de onda obtidas na simulação do SLM mais compensador lead.....	70
Figura 4.3: Diagrama de blocos utilizado na simulação do SLM.....	72
Figura 4.4: Forma de onda obtida sem a correção do erro em regime.....	73
Figura 4.5: Forma de onda com correção do erro em regime.....	73
Figura 4.6: Forma de onda obtida a partir de linearização única.....	74
Figura 4.7: Forma de onda obtida a partir da linearização passo à passo.....	75
Figura 4.8: $\alpha = 0.9$	76
Figura 4.9: $\alpha = 0.7$	76

Figura 4.10: $\alpha = 0.5$	76
Figura 4.11: $\alpha = 0.3$	76
Figura 4.12: $\alpha = 0.1$	77
Figura 4.13: $N=7$ e $Nu=5$	81
Figura 4.14: $N=10$ e $Nu=5$	81
Figura 4.15: $N=15$ e $Nu=5$	82
Figura 4.16: $N=20$ e $Nu=5$	82
Figura 4.17: $N=20$ e $Nu = 7$	82
Figura 4.18: $N=20$ e $Nu = 10$	82
Figura 4.19: $N=20$ e $Nu=15$	83
Figura 4.20: $N=20$ e $Nu=20$	83
Figura 4.21: Sistema sem técnica de rejeição com perturbação na saída.....	84
Figura 4.22: Sistema sem técnica de rejeição com perturbação na entrada.....	84
Figura 4.23: Sistema sujeito a perturbação na saída com técnica de rejeição a perturbação.....	85
Figura 4.24: Sistema sujeito a perturbação na entrada com técnica de rejeição a perturbação.....	85
Figura 4.25: Representação do modelo sob a forma de blocos integrativos.....	89
Figura 4.26: Diagrama elétrico equivalente ao modelo linearizado do SLM.....	89
Figura 4.27: Diagrama de blocos utilizado na experimentação.....	90
Figura 4.28: Formas de onda obtida a partir da simulação.....	91
Figura 4.29: Forma de onda sem correção de erro em regime.....	93
Figura 4.30: Forma de onda com correção do erro em regime.....	93
Figura 4.31 $\alpha = 0.9$	95
Figura 4.32: $\alpha = 0.7$	95
Figura 4.33: $\alpha = 0.5$	95

Figura 4.34: $\alpha = 0.3$	95
Figura 4.35: $\delta = 10$ e $\lambda = 10$	96
Figura 4.36: $\delta = 20$ e $\lambda = 10$	96
Figura 4.37: $\delta = 30$ e $\lambda = 10$	96
Figura 4.38: $\delta = 40$ e $\lambda = 10$	96
Figura 4.39: $N = 28$	97
Figura 4.40: $N = 21$	97
Figura 4.41: $N = 14$	97
Figura 4.42: $N = 7$	97
Figura 4.43: $Nu = 28$	98
Figura 4.44: $Nu = 14$	98
Figura 4.45: excursão 0.6V.....	99
Figura 4.46: excursão 0.8V.....	99
Figura 4.47: excursão 1V.....	100
Figura 4.48: excursão 1.15V.....	100
Figura 4.49: Simulação utilizando compensador lead.....	100

Índice de Tabelas

Tabela 3.1: Relação entre tensão de entrada (U) e corrente de saída (I).....	57
Tabela 4.1: Localização dos pólos ao variar a razão δ/λ	79
Tabela 4.2: Localização dos pólos ao variar o horizonte de predição N.....	80
Tabela 4.3: Localização dos pólos ao variar o horizonte de controle Nu.....	80
Tabela 4.4: Tempo médio de execução para caso irrestrito.....	86
Tabela 4.5: Tempo médio de execução utilizando programação quadrática.....	87
Tabela 4.6: Relação entre os respectivos sinal de referência, excursão e figura.....	99

Lista de Símbolos

Os símbolos utilizados neste trabalho estão listados a seguir:

\hat{y} – valor de saída predito

α – coeficiente de amortização da referência

ρ – constante de conversão corrente-tensão

γ – constante de conversão distância-tensão

ϕ - fluxo magnético

λ – peso da variação do controle

δ – peso do erro de rastreamento

μ_m – permeabilidade magnética do meio

ε_r – constante relativa do dielétrico

μ_r – permeabilidade relativa

Δt – período de amostragem

Δu – variação do controle

Δu_{max} – maior variação de controle possível

Δu_{min} – menor variação de controle possível

\mathbf{A} – matriz de transição de estados

\mathbf{B} – vetor coluna de transferência da entrada

\mathbf{b} – vetor restrição

B_m – densidade de fluxo magnético

\mathbf{C} – vetor linha de transferência da saída

C_x – capacitância do elemento x

e – força eletromotriz

E – esperança matemática

f – resposta livre do sistema

F_m – força de atração magnética

g – aceleração da gravidade (9.8m/s^2)

\mathbf{G} – matriz dinâmica do sistema

$G(s)$ – função de transferência em “s”

g_i – coeficientes da resposta ao degrau amostrada

h_i – coeficientes da resposta ao impulso amostrada

H_m – intensidade de campo magnético

I – corrente

\mathbf{I} – matriz identidade

j – número de saídas do sistema

J – função objetivo

L – indutância

m – massa do objeto a ser levitado

\mathbb{N}^* – conjunto dos números naturais

N – horizonte de predição

N_u – horizonte de controle

p – número de entradas do sistema

R – resistência

\mathbf{Re} – matriz de restrição

$S_m(\infty)$ – ganho em regime do modelo

$S_p(\infty)$ – ganho em regime da planta

U_{max} – maior valor de controle

U_{min} – menor valor de controle

V_x – tensão no ponto x

w – trajetória de referência

y_m – saída do modelo

Y_{max} – maior valor de leitura do sensor ótico

Y_{min} – menor valor de leitura do sensor ótico

y_p – saída da planta

Z – distância entre esfera e eletromagneto

Lista de Siglas

As siglas utilizadas neste trabalho são:

AO – Amplificador Operacional

A/D – Analógico/Digital

AC – Corrente Alternada

CRHPC – Constrained Receding Horizon Predictive Control

D/A – Digital/Analógico

DC – Corrente Contínua

DMC – Dynamic Matrix Control

EDO – Equação Diferencial Ordinária

EE – Espaço de Estado

EHAC – Extended Horizon Adaptive Control

EPSAC – Extended Prediction Self-Adaptive Control

EWB – Eletronic Workbench

FCCU – Fluidised Catalytic Cracking Unit

FIR – Finite Impulse Response

FTMA – Função de Transferência de Malha Aberta

GPC – Generalized Predictive Control

GUI – Graphical User Interface

I/O – Input/Output

IDCOM – IDentification and COMmand

LCP – Linear Complementary Problem

LEV MAG – LEVitação MAGnética

LMPC – Linear Model Predictive Control

LP – Linear Programming

LQ – Linear Quadratic

LQG – Linear Quadratic Gaussian

LQR – Linear Quadratic Regulator

MA – Malha Aberta

MF – Malha Fechada

MPC – Model Predictive Control

MPHC – Model Predictive Heuristic Control

NMPC – Non Linear Model Predictive Control

PC – Personal Computer

PID – Proporcional-Integral-Derivativo

PQ – Programação Quadrática

QDMC – Quadratic Dynamic Matrix Control

SIORHC – Stabilizing Input-Output Receding Horizon Control

SLM – Sistema de Levitação Magnética

SMOC – Shell Multivariable Optimizing Controller

ZOH – Zero Order Hold

Resumo

A utilização de controladores preditivos já está de certa forma consolidada na indústria de processos, porém é aplicada, geralmente, a plantas cuja dinâmica é lenta e estável em malha aberta. A proposta deste trabalho é implementar um algoritmo que realize o controle preditivo de um sistema não-linear, instável em malha aberta e com dinâmica relativamente rápida, e que possua técnicas para correção de erro em regime, rejeição de perturbações, tratamento de restrições de entrada e de saída, e seguimento de referência. Como sistema-exemplo, empregou-se um processo de levitação magnética construído pela Feedback Instruments.

Para possibilitar a utilização de técnicas bem estabelecidas de controle preditivo baseadas em modelos lineares, a estratégia adotada usa versões linearizadas do modelo do levitador. A linearização é refeita a cada período de amostragem, de forma a manter o mais próximo possível o comportamento da planta e do modelo.

O algoritmo de controle foi executado no computador mestre de uma topologia mestre-escravo implementada com a ferramenta xPC Target do aplicativo Matlab. Os computadores se comunicam através da interface serial RS-232, e é no computador escravo, que é inicializado com o kernel xPC Target, que está instalada uma placa de conversão analógica-digital a qual realiza a interface entre o algoritmo e o sistema físico.

As simulações computacionais são resolvidas utilizando-se um algoritmo de programação quadrática, acionado pelo comando “quadprog” do Matlab. Os testes experimentais também são resolvidos através de uma lei de controle obtida a partir dos conceitos de controle preditivo, porém não levam em consideração as restrições do sistema físico, ou seja as simulações levam em consideração o tratamento de restrições, porém no caso experimental a lei de controle é obtido a partir do caso irrestrito.

Ao longo do trabalho são observados, analisados e discutidos resultados obtidos inicialmente por meio de simulação computacional, em seguida através da utilização de um computador analógico e por fim através da implementação do controle junto ao sistema físico. Em todos os casos foi possível não só estabilizar a saída do sistema, como também fazê-la seguir uma determinada trajetória de referência.

Abstract

Although predictive control technology has been widely adopted in industrial frameworks, its use has been mostly focused on slow, open-loop stable processes. In this context, this work investigates the real-time implementation of a model predictive control algorithm for a magnetic levitation system (MLS), which is a nonlinear, open-loop unstable plant with relatively fast dynamics. More specifically, techniques for steady-state error correction, disturbance rejection, reference tracking and input/output restriction management are studied.

To facilitate the use of well-established techniques of predictive control based on linear models, the adopted strategy uses linearized versions of the model of the MLS. The linearization is re-done to each sampling time, in way to maintain the closest possible the behavior of the plant and of the model.

The control algorithm was implemented with two computers in a master-slave topology, employing the xPC Target utility of Matlab Real-Time Workshop. In this configuration, the control tasks are managed by a real-time kernel loaded in the slave computer, which is fitted with a data acquisition board. In the master computer, the control algorithms are compiled into a code that is uploaded to the slave via the serial interface RS-232.

The computational simulations is resolved using an algorithm of quadratic programming, worked by the command "quadprog" of Matlab. The experimental tests are also resolved through a control law obtained from the concepts of predictive control, even so they don't take in consideration the constraints of the physical system.

The performance of the predictive control strategy is initially analysed on the basis of digital and analog simulations. The controller is then tested in the physical system built by Feedback Instruments. In all cases it was possible not only to stabilize the output of the system, as well as to do it to follow a certain reference trajectory.

Agradecimentos

A Deus por sua grandeza que nos auxilia e conforta a todo momento.

A meus pais Luiz Miura e Sônia Orchiucci Miura por vosso amor, carinho, dedicação e por terem feito de suas vidas a realização de seus filhos.

A meus irmãos Luiz Enrico e Paula por vossa companhia e amor fraternal.

Ao meu professor orientador Roberto Kawakami Harrop Galvão pela pessoa e profissional que é, com quem tive a oportunidade e prazer de conviver e trabalhar.

Ao meu amigo Ricardo Kazuo Itikava pela ajuda com as revisões.

Ao Técnico do Laboratório de Controle Digital Eduardo de Almeida Fonseca, que em momento algum, hesitou em me auxiliar.

Ao Alexandre Rodrigues Mesquita pela ajuda com o xPC Target.

Ao Anderson Anjos da Silva pelo auxílio com a biblioteca DMatrix, e com o Visual Studio.

Ao órgão financiador CNPQ.

A todos que de alguma forma contribuíram para a realização desta tese.

Capítulo 1

INTRODUÇÃO

1.1 Generalidades

Atualmente, em consequência da acirrada competitividade industrial promovida principalmente pela globalização, as empresas vêm-se forçadas a reduzir o custo de produção e melhorar a qualidade. A reavaliação e melhora dos processos industriais propiciam incrementos reais de produtividade e asseguram a qualidade a custos mínimos possibilitando assim a fácil aceitação de produtos manufaturados no mercado nacional e internacional a preços competitivos. Mas para que isto seja alcançado, necessita-se de novas idéias e métodos, e assim as técnicas de controle baseadas em otimização vêm ganhando espaço na indústria.

Uma dessas técnicas é o controle preditivo, o qual determina sua lei de controle através da minimização de um determinado índice de desempenho, que pode penalizar gastos demasiados com matéria-prima, consumo excessivo de energia elétrica, e até mesmo quantidade excessiva de peças e/ou material em estoque. Além disso, ainda pode tratar restrições permitindo que os valores de controle, estado e saída sejam mantidos próximos a seus limites de operação, os quais geralmente constituem a região de atuação ótima do sistema.

Pretende-se unir tal estratégia de controle aos fenômenos magnéticos que fascinam não só por seus aspectos científicos, como também por apresentarem inúmeras possibilidades de aplicações práticas. Podem-se citar, por exemplo, a fabricação de motores, alto-falantes,

microfones, memórias, discos rígidos, fitas magnéticas [34], entre outras aplicações mais ousadas, tais como mancais de precisão utilizados em corações artificiais, e trens de levitação magnética que hoje já funcionam no Japão, China e Alemanha.

Entre as técnicas empregadas para controle de sistemas de levitação magnética, podem-se citar controle adaptativo [11,16, 37], controle baseado em caos [19], reguladores Linear Quadrático (LQ, Linear Quadratic) [43], modo deslizante [12] e linearização por realimentação [17,40]. Técnicas de controle preditivo não têm sido amplamente utilizadas neste cenário, muito provavelmente pelo fato da dinâmica dos sistemas de levitação magnética ser bastante rápida quando comparada à dinâmica de processos químicos e petroquímicos, ambiente no qual o Controle Preditivo baseado em Modelo (MPC, Model Predictive Control) já está consolidado.

1.2 Objetivo da Tese

Nos últimos anos houve um aumento tremendo referente à quantidade de informações com que as pessoas, de forma geral, devem lidar. Esta necessidade torna-se ainda maior em ambientes de pesquisa e industrial, surgindo a necessidade do auxílio do uso de um computador.

Na existência de um modelo do processo, a utilização destas máquinas poderosíssimas possibilitou não só o desenvolvimento, como também o ajuste de controladores de forma muito mais rápida e econômica, através de simulações computacionais.

A estratégia de controle preditivo em tempo real utilizada neste trabalho é executada sobre dois computadores na configuração mestre-escravo, onde o computador mestre carrega o algoritmo de controle, e o computador escravo realiza a interação com o sistema físico.

Esta topologia será utilizada no controle de um sistema de levitação magnética, constituído de uma esfera mantida em levitação por um campo magnético gerado por um

eletroímã, sistema este bastante interessante e desafiador, pois trata-se de uma planta não-linear, instável em malha aberta e que possui restrições de entrada e saída. Para o caso apresentado neste trabalho a restrição referente à amplitude de saída é do tipo “hard”, ou seja é uma restrição que não pode ser violada uma vez que causaria a descontinuidade na operação do sistema, pois caso saia da faixa de operação permitida, a esfera ou cairá ou colará no magneto causando então a descontinuidade de operação.

Este é, então, o objetivo deste trabalho: controlar uma planta com características estáticas e dinâmicas complexas, com frequências de interesse relativamente altas (da ordem de 10Hz), utilizando uma estratégia de controle avançada e moderna que é a preditiva.

1.3 Apresentação da Tese

Os capítulos estão dispostos da seguinte forma.

Capítulo 2: Este capítulo é dedicado integralmente a levitação magnética. Inicialmente apresentam-se conceitos e teoremas, comentam-se os métodos de levitação magnética, descreve-se a levitação magnética com maior ênfase na suspensão ferromagnética, apresenta-se a dinâmica da suspensão eletromagnética de forma geral, em seguida é apresentado o Sistema de Levitação Magnética da Feedback Instruments, utilizado neste trabalho.

Capítulo 3: Neste capítulo é realizada uma revisão da literatura sobre controladores preditivos baseados em modelo. Para tal, apresenta-se um breve histórico sobre MPC e seus algoritmos pioneiros IDCOM & DMC , formulação MPC, modelos mais comumente utilizados, correção do erro em regime, e tratamento de restrições.

Capítulo 4: Descreve a estratégia MPC utilizada no controle do levitador magnético, apresentando a metodologia utilizada na linearização e discretização do modelo, o tratamento das restrições, e o algoritmo de Controle Preditivo baseado em Modelo Linear (LMPC, Linear Model Predictive Control) utilizado.

Capítulo 5: Este capítulo apresenta os resultados obtidos em simulação e experimentalmente, além de discussões, interpretações e comparações entre as técnicas utilizadas em simulação e o efeito de cada parâmetro ajustável do controlador preditivo.

4

Capítulo 6: São consideradas as conclusões finais, e sugeridos trabalhos futuros.

Por fim o texto é encerrado com os ANEXOS e a referência bibliográfica utilizada na construção deste trabalho.

Capítulo 2

CONTROLE PREDITIVO BASEADO EM MODELO

2.1 Introdução

O Controle Preditivo baseado em Modelo (MPC, Model Predictive Control) é uma forma de controle realimentado que gera as ações de controle baseadas em um controle ótimo, o qual minimiza uma função objetivo de horizonte finito. Assim as ações de controle são escolhidas baseadas em seu impacto no futuro ao invés de baseadas somente em ações presentes [20].

É interessante destacar que o MPC não se refere a uma estratégia de controle específica, mas a uma quantidade muito grande de métodos de controle que utilizam o modelo do processo para se obter o sinal de controle, que por sua vez otimiza uma determinada função objetivo ou índice de desempenho.

Atualmente, com a comercialização de computadores cada vez mais poderosos, tornou-se viável a implementação em tempo real desta classe de algoritmos de controle, tanto que existem muitas aplicações bem sucedidas do controle preditivo em uso, não só em processos industriais, principalmente das indústrias química e petroquímica, como também em manipuladores robóticos, indústria metalúrgica, processamento de alimentos, servomecanismos, fabricação de papel dentre outros [29].

O MPC apresenta diversas vantagens sobre outros métodos, tais como:

- Pode ser utilizado para controlar uma grande variedade de processos, incluindo sistemas com grande atraso de transporte, fase não mínima ou instáveis;

- O caso multivariável pode ser tratado;
- Introduz no controle uma ação antecipatória que compensa os distúrbios medidos;
- O tratamento de restrições é conceitualmente simples e pode ser incluído durante a fase de projeto.

Contudo uma desvantagem é a necessidade de um modelo adequado do processo. Adicionalmente, sua utilização está restrita a processos cuja dinâmica seja lenta em relação à velocidade de processamento do aparato computacional utilizado em sua implementação, uma vez que cada passo do controle deve obedecer às restrições e à minimização da função objetivo dentro de um determinado horizonte de controle [33]. Devido a esta necessidade de alto poder computacional, o MPC é geralmente implementado no modo supervisor [42], ou seja atua em um nível hierárquico acima ao do controle regulatório.

Uma proposta para tais níveis hierárquicos foi sugerida por Richalet et al. e é caracterizada de acordo com a função exercida [41]:

- Nível 3 – Agendamento de tempo e espaço da produção;
- Nível 2 – Otimização da faixa de atuação do processo, de forma a minimizar custos e assegurar uma produção em quantidade e de qualidade;
- Nível 1 – Controle dinâmico multivariável da planta;
- Nível 0 – Controle regulatório de sistemas subordinados; Controle PID.

Foi apontado também que benefícios significativos não viriam somente da redução de variações da variável controlada por meio de uma melhor dinâmica do controle no nível 1. Os benefícios econômicos reais viriam no nível 2, onde a melhor dinâmica do controle permite que a faixa de atuação da variável controlada seja posta próxima a região de restrição sem violá-la. Este argumento fornece a motivação econômica básica para a utilização da tecnologia MPC. E ainda, segundo Cutler e Perry (1983), apesar da otimização em tempo real

da operação ser uma tarefa difícil e complexa, ela em conjunto com o controle avançado pode incrementar de 6 a 10% a lucratividade da referida unidade. Porém, sabe-se da necessidade da escolha apropriada das restrições do processo para o sucesso de um projeto de otimização em tempo real. Adicionalmente, os benefícios da otimização dependem de alguns fatores, tais como:

- disponibilidade de mercado para aumento da capacidade da unidade;
- diferença de preços entre os produtos;
- magnitude do consumo específico de energia;
- número de variáveis manipuladas e das restrições;
- acurácia do modelo do processo.

A implementação do MPC na indústria não é um problema trivial e é certamente uma tarefa mais difícil do que as técnicas de controle clássico baseadas em controladores PID. Uma aplicação MPC é mais custosa financeiramente, temporalmente e requer pessoas com um melhor treinamento em controle. Como já dito, de início, um modelo do processo deve ser obtido, e isto pode requerer uma quantidade significativa de testes da planta, os quais na maioria dos casos implicam em introduzir excitações que não são usualmente enviadas a planta. Os equipamentos de controle necessitam, em alguns casos, de computadores mais poderosos e melhores instrumentos, além dos pacotes comerciais de MPC serem caros.

Em contrapartida a estas dificuldades o MPC tem provado ser economicamente eficaz através da redução de custos operacionais ou pelo aumento da produção e esta é uma das técnicas de controle avançada de maior sucesso utilizada na indústria. As razões para este sucesso dependem em particular da aplicação, mas estão relacionadas às habilidades do MPC em otimizar funções custo e ao tratamento de restrições. Algumas razões podem ser mencionadas [6]:

- Otimização das condições operacionais: MPC otimiza uma função custo a qual pode ser formulada de forma a minimizar custos operacionais ou qualquer outro objetivo com implicações econômicas.
- Otimização de transições: a função objetivo MPC pode ser formulada de forma a otimizar uma função a qual mede o custo para levar o processo de um ponto de operação a outro.
- Minimização da variância do erro: um MPC pode ser formulado de forma a minimizar a variância do erro de saída. Uma variância menor produzirá benefícios econômicos pelas seguintes razões:
 - Uma variância menor pode aumentar a qualidade do produto, tal como sua uniformidade.
- O tratamento explícito de restrições pode permitir que o processo opere mais próximo de suas restrições e de sua condição ótima de operação, caso o ponto de operação ótimo esteja situado junto a intersecção de alguma das restrições, como geralmente ocorre na maioria dos processos restritos.
- As considerações explícitas das restrições podem reduzir o número de violação de restrições reduzindo, assim, o número de desligamentos emergenciais.

O objetivo deste capítulo é estudar esta classe de controle conhecida como controle preditivo baseado em modelo. A seguir, apresenta-se um breve histórico do MPC, a seção seguinte aborda os dois algoritmos pioneiros na área de controle preditivo, o comando e identificação (IDCOM, IDentification and COMmand) e o controle por matriz dinâmica (DMC, Dynamic Matrix Control), e por fim descreve-se o controle preditivo baseado em modelo atuando em plantas lineares e não-lineares.

2.2 Histórico do Controle Preditivo Baseado em Modelo

Apesar de idéias sobre MPC e técnicas de controle que utilizam horizonte retrocedente (Propoi,1963) terem surgido na década de 1960, o interesse neste campo começou a crescer somente na década de 1980 após as publicações dos algoritmos IDCOM (veja seção 2.2.1) [35] e DMC (veja seção 2.2.2) [10] e de exposições mais claras e concretas sobre controle preditivo generalizado (GPC, Generalized Predictive Control) [7].

Tal como já citado, a gênese do controle preditivo deve ser reportada à década de 1960, quando surgiu o controle ótimo, sendo o MPC obtido através da solução de um problema de controle ótimo padrão, com a diferença de que nos controladores preditivos o problema de controle ótimo deve ser resolvido dentro de um horizonte finito, em oposição ao horizonte infinito usualmente empregado na abordagem de controle linear ótimo. Outra grande diferença é que no MPC o problema de controle é resolvido em tempo real utilizando os estados atuais da planta, ao contrário da estratégia de cálculo “off-line” do ganho de retroalimentação [23].

Surge então o algoritmo DMC que foi concebido para tratar processos/plantas relacionados ao controle multivariável com restrições. Antes do surgimento do DMC estes problemas eram tipicamente tratados utilizando-se um controlador para cada laço do processo/planta.

Outra linha de pesquisa cresceu independentemente, sobre a idéia do controle adaptativo, desenvolvendo estratégias essencialmente para processos monovariáveis formulado com modelos tipo entrada-saída. Esta foi projetada para minimizar um valor esperado de um determinado critério quadrático em um dado horizonte de controle finito ou assintoticamente infinito. Este método conhecido como controle adaptativo com horizonte estendido (EHAC, Extended Horizon Adaptive Control) tenta manter o valor da saída futura próximo ao valor da referência, por um certo período de tempo após o atraso do processo. Já o

controle auto-adaptativo com predição estendida (EPSAC, Extended Prediction Self-Adaptive Control) propõe um sinal de controle constante começando no momento atual enquanto usa um preditor sub-ótimo ao invés de resolver a equação de Diophantine como é feito no EHAC.

O MPC foi também formulado em espaço de estados (EE). Isto não só permite o uso de teoremas conhecidos da teoria de espaço de estados, como também facilita a generalização para casos mais complexos, tais como os de sistemas com perturbações estocásticas e ruídos nas variáveis medidas. As técnicas de estimação de estados surgidas do controle ótimo estocástico podem ser usadas na predição. Esta perspectiva permite o uso de resultados bem estabelecidos de estabilidade e robustez.

Embora os primeiros trabalhos sobre GPC [25] provavam alguns teoremas específicos de estabilidade usando relações de EE e estudavam a influência de filtros polinomiais na melhoria da robustez, a carência original de resultados da estabilidade geral para controladores com janela de horizonte finito era reconhecida como um desafio.

Devido a isto, uma nova linha de trabalho, que garantia a estabilidade dos novos métodos de controle preditivo apareceu na década de 1990. Dois foram estes métodos: o Controle Preditivo Restrito com Horizonte Retrocedente (CRHPC, Constrained Receding Horizon Predictive Control) desenvolvido por Clarck e Scattolini e o Estabilização Entrada-Saída por Controle com Horizonte Retrocedente (SIORHC, Stabilizing Input-Output Receding Horizon Control) desenvolvido por Mosca [4] foram desenvolvidos independentemente e tiveram sua estabilidade provada através da imposição de restrições de igualdade na saída após a janela de horizonte finito.

2.2.1 IDCOM

Este algoritmo foi a primeira aplicação envolvendo MPC, e foi apresentado por Richalet et. al. em uma conferência em 1976 e sumarizado posteriormente em 1978 através de

um artigo publicado na revista Automática [35]. Apesar do aplicativo ser conhecido como IDCOM, um acrônimo de Identificação e Comando, a estratégia de controle foi descrita como Heurística de Controle Preditivo baseado em Modelo (MPHC, Model Predictive Heuristic Control). As características do IDCOM são:

- Modelo linear baseado na resposta ao impulso;
- Função objetivo quadrática;
- Horizonte de predição finito;
- Comportamento futuro da saída da planta especificado por uma trajetória de referência;
- Restrições de entrada e saída incluídas na formulação;
- Controle e Identificação determinados de forma dual.

Obs.: Horizonte de predição das entradas, ou simplesmente, *horizonte de controle*, $N_u \in \mathbb{N}_+$, designa o número de instantes nos quais são otimizadas as ações de controle. Horizonte de predição das saídas, usualmente definido como *horizonte de predição*, $N \in \mathbb{N}_+$, representa o número de instantes ao longo dos quais são calculadas as predições das saídas controladas e/ou estados do modelo [36].

Richalet et al. escolheram uma representação do processo de forma que as entradas influenciam as saídas diretamente. As entradas do processo são divididas em variáveis manipuladas (aquelas ajustadas pelo controle) e variáveis de distúrbio (aquelas não disponíveis para controle), e as saídas do processo são tratadas como variáveis controladas. A relação entre as entradas e saídas do processo é feita usando o modelo discreto da Resposta ao Impulso Finita (FIR, Finite Impulse Response) (detalhes veja seção 2.5.1).

A resposta ao impulso finita é obtida através de dados de teste da planta usando um algoritmo projetado para minimizar a distância estrutural entre a planta e o modelo de

resposta ao impulso, ou seja o algoritmo procura a seqüência de ponderação que conduz ao menor erro de predição do modelo. Então o algoritmo iterativo resultante realiza pequenos ajustes nos coeficientes a cada passo, de forma que a distância estrutural tende a diminuir. O problema de controle é resolvido utilizando o mesmo algoritmo da identificação, uma vez que o problema de controle é dual ao problema de identificação. No problema de identificação, conhecidas as saídas e entradas deseja-se estimar os coeficientes do modelo de resposta ao impulso; no problema de controle, conhecida a trajetória de saída desejada e os coeficientes do modelo, o objetivo é determinar os valores de entrada do processo. Devido à predição da saída aparecer como sendo um produto da entrada pelos vetores de coeficiente, o mesmo algoritmo pode ser usado para encontrar ambos (controle e coeficientes). A natureza iterativa do algoritmo de controle permite que as restrições de entrada e saída sejam checadas enquanto o algoritmo realiza o processo de solução. Como a lei de controle é não-linear e não poderia ser expressa como função de transferência, Richalet et al. referiram-se a esta estratégia como sendo uma heurística, o que no contexto atual é tratado como controle MPC não-linear.

O algoritmo MPHC leva a trajetória da saída futura predita, tão perto quanto possível da trajetória de referência, a partir do valor atual da saída. A velocidade de resposta de malha fechada é ajustada através da constante de tempo da trajetória de referência. Esta observação é importante na prática, pois é a velocidade de resposta desejada que irá determinar a “agressividade” do algoritmo, assim aumentando a constante de tempo o controlador torna-se mais lento, porém menos sensível a ruído de medida.

2.2.2 DMC

No início da década de 1970 engenheiros da Shell desenvolveram sua própria tecnologia MPC [31]. Cutler e Ramaker apresentaram detalhes de um algoritmo de controle multivariável sem restrições em 1979 no encontro Nacional AIChE e em 1980 à Joint

Automatic Control Conference, ao qual chamaram de Dynamic Matrix Control. Em 1980 Prett e Gillette descreveram uma aplicação da tecnologia DMC para um reator de Unidade de Quebra Catalítica de Fluídos (FCCU, Fluidised Catalytic Cracking Unit), no qual o algoritmo tinha sido modificado para tratar não-linearidades e restrições. As características principais do algoritmo de controle DMC incluem:

- Modelo linear baseado na resposta ao degrau;
- Função objetivo quadrática;
- Horizonte de predição finito;
- Comportamento futuro das saídas da planta especificado para tentar seguir a região de operação tão perto quanto possível;
- Entradas ótimas computadas como solução de um problema de mínimos quadrados.

O modelo linear baseado na resposta ao degrau usado no algoritmo DMC relaciona as mudanças na saída do processo à soma dos pesos da variação das entradas passadas (detalhes veja seção 2.6.1).

O objetivo do controlador DMC é guiar a saída o mais próximo possível de uma faixa de operação em um problema de mínimos quadrados, com um termo de penalidade para variações nas variáveis manipuladas. Isto resulta em variações de entrada pequenas e uma resposta da saída menos agressiva. Prett e Gillette [15] formalizaram este conceito matematicamente, definindo fatores de supressão das variações, fatores estes projetados para penalizar mudanças excessivas da entrada.

Cutler e Ramaker [10] exibiram resultados de uma aplicação de um controle de temperatura de uma fornalha para demonstrar a melhoria na qualidade do controle quando se utiliza o algoritmo DMC, em comparação com o compensador Proporcional-Integral-Derivativo (PID) e lead/lag.

Prett e Gillette descreveram uma aplicação da tecnologia DMC para o controle de um reator FCCU. Este sistema de controle foi implementado em uma hierarquia multi-nível, com um modelo de regime não-linear no topo. A cada ciclo de otimização, parâmetros do modelo não-linear eram estimados de forma a fazer coincidir as predições do modelo com os dados operacionais de regime medidos. O modelo não-linear calibrado era então perturbado numericamente para gerar derivadas parciais de cada saída do processo com respeito a cada entrada do mesmo. As derivadas parciais eram então utilizadas em uma Programação Linear (LP, Linear Programming) para se calcular o novo ponto de operação economicamente ótimo para o FCCU, sujeito ainda às restrições de regime do processo. Por fim, os alvos de entrada e saída ótimos do processo eram enviados ao algoritmo DMC para implementação. Tão logo o controlador DMC movia a unidade para seu novo estado de regime, o ciclo de otimização era repetido. Esta separação do sistema de controle em otimização do estado de regime restrito e dinâmica do controle é bastante similar à estrutura descrita por Richalet et al. e tornou-se padrão no projeto de sistemas de controle industrial.

O algoritmo DMC tem como finalidade levar o sistema de um estado ótimo de regime a um outro. Embora a solução da programação linear forneça alvos ótimos para as entradas e saídas do processo, erros dinâmicos podem fazer com que o algoritmo DMC mova as entradas para longe de seu estado de regime. Porém, uma vez que mover uma entrada para longe de sua região ótima custa mais caro, o sistema de controle deve determinar esta estratégia de modo racional. O algoritmo DMC foi modificado para levar em consideração tais estratégias incluindo uma equação adicional para cada entrada do processo. A nova equação requer que a soma de todas as movimentações de uma determinada entrada deve ser igual ao ajuste total requerido para trazê-la de volta ao seu estado ótimo. Isto permite uma certa liberdade para as entradas moverem-se dinamicamente, mas requer que a solução para o estado de regime da

entrada possa ser resolvida através de um problema de mínimos quadrados, com a estratégia determinada através dos pesos apropriados de uma função objetivo.

Prett e Gillette descreveram modificações adicionais ao algoritmo DMC, de forma a prevenir violações das restrições de entrada. Quando uma entrada futura predita chega suficientemente próxima da restrição absoluta, uma equação extra é adicionada ao modelo do processo, de forma a guiá-la de volta a uma região viável. Esta técnica é conhecida como a de restrições variantes no tempo. Devido à decisão de adicionar uma equação ter de ser feita em tempo real, a solução da inversa da matriz tem que ser recalculada a cada execução do controle. Prett e Gillette desenvolveram uma solução na qual a inversa da matriz original pode ser calculada “off-line”, necessitando somente do cálculo em tempo real da inversa da matriz correspondente às restrições ativas variantes no tempo.

Por fim, sabe-se que o surgimento do algoritmo DMC teve um impacto tremendo na indústria, em grande parte devido ao seu algoritmo possuir um identificador de modelos e um otimizador global da planta.

2.2.3 Estatísticas sobre publicações em MPC nos últimos trinta e três anos.

Finalizando esse breve histórico, vale comentar que o MPC tornou-se a estratégia de controle avançado mais aplicada na indústria química, fato que pode ser comprovado pelo número enorme de publicações reportando aplicações. Além disso, a pesquisa acadêmica sobre MPC também tem ocupado espaço significativo na literatura de controle. Essa popularidade, em números, pode ser medida por uma pesquisa à base de dados “Science Direct®”, no mês de Julho de 2003, utilizando-se as palavras-chave “Model Predictive Control”, “Predictive Control”, “Receding Horizon Control”, “Moving Horizon Control” e “Dynamic Matrix Control”. Tal pesquisa, no período de 1970 a 1989 resultou em 27 artigos e no período de 1990 a 2003 o número de artigos selecionados foi de 1096.

2.3 Formulação MPC

Inicialmente apresentar-se-á a estrutura básica de um controlador preditivo baseado em modelo, ilustrada na figura 2.1, para um melhor entendimento do procedimento realizado em um sistema controlado por um MPC.

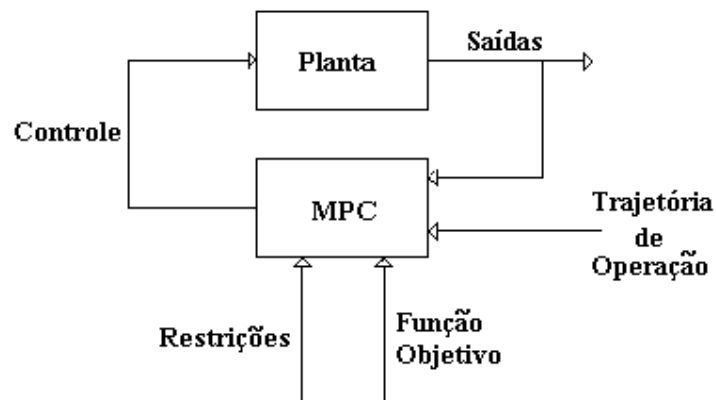


Figura 2.1: Estrutura básica de um MPC.

O MPC é uma forma de controle onde a ação do controle atual é obtida a partir da solução, a cada período de amostragem, de um problema de controle ótimo com horizonte finito, utilizando, na formulação em EE, os estados atuais da planta como condições iniciais. A otimização gera uma seqüência de controle a qual somente o primeiro valor de controle é aplicado à planta [23].

É interessante observar que a abordagem de controle preditivo baseado em modelo linear pode ser estendida a plantas não-lineares através, por exemplo, da linearização em tempo real do modelo não-linear da referida planta. Existem trabalhos [5] que realizam a comparação entre a utilização da técnica de Controle Preditivo baseado em Modelo Não-Linear (NMPC, Nonlinear Model Predictive Control) com a utilização de técnicas de Controle Preditivo baseado em Modelo Linear (LMPC, Linear Model Predictive Control) com

linearização em tempo real, e que concluem um melhor desempenho da técnica NMPC, porém um menor custo computacional para o caso do LMPC com linearização em tempo real.

Existe ainda a possibilidade, para casos onde a dinâmica da planta é extremamente não-linear, de se utilizar múltiplos modelos lineares, onde cada modelo representa uma região de operação da planta. A utilização desta técnica traz a idéia de um controlador adaptativo, que tenha a habilidade de reconhecer qual a atual região de operação da planta [29].

As metodologias de controle preditivo baseadas em modelo utilizam o modelo do processo para prever o comportamento futuro da planta, e assim obter a lei de controle que minimiza um determinado índice de desempenho. Assim, tem-se que o MPC é uma classe de estratégia de controle, cujas implementações diferenciam-se basicamente devido ao modelo de predição e à função objetivo utilizada [8].

Desse modo a metodologia de todos os controladores pertencentes à família MPC é caracterizada pela necessidade inicial de se determinar o modelo do processo, suas restrições, e uma função objetivo adequada. Torna-se, então, possível escrever um algoritmo o qual irá a cada instante t determinar os valores de saídas futuras utilizando o modelo do processo/planta. Porém para prever, de modo correto, o comportamento da planta é necessário o conhecimento do histórico do processo, histórico esse que depende também do modelo a ser utilizado. Por exemplo, para o caso de modelos em espaço de estado basta conhecer os valores dos estados atuais e do último controle aplicado, já para modelos do tipo FIR é necessário o conhecimento de uma sequência de valores de entrada e saída do modelo.

A predição de um sistema caracterizado por uma equação diferencial não-homogênea, necessita também da determinação prévia da sequência de controle a ser aplicada em tal sistema. Essa sequência de controle deve manter a saída do processo tão próximo quanto possível da trajetória de referência, além de minimizar uma função objetivo composta, geralmente, por uma função quadrática dos erros entre o sinal de saída predito e a trajetória de

referência predita, e do esforço do controle. Uma solução explícita, para tal seqüência de controle, pode ser obtida se o critério for quadrático, o modelo linear e não existirem restrições, caso contrário algum método iterativo de otimização pode ser utilizado. Por fim, obtida a seqüência de controle, aplica-se somente o primeiro elemento ao processo, e realiza-se um novo passo.

Na maioria dos casos o valor de controle entre dois instantes de amostragem é considerado constante, isso é supõe-se a existência de um segurador de ordem zero na entrada da planta [20].

É interessante destacar a diferença existente entre trajetória comandada e trajetória de referência. A primeira é gerada externamente ao algoritmo de MPC, por exemplo uma onda quadrada, ou ainda uma senóide. Já a segunda é determinada internamente pelo controlador preditivo utilizando-se técnicas de suavização de referência que determinam qual o caminho a ser seguido pela saída da planta até encontrar-se com a trajetória comandada, tal como mostra a equação abaixo [6].

$$r(t+k) = s(t+k) - \alpha^k (y(t) - s(t)) \quad (2.1)$$

onde r é a trajetória de referência;

s é a trajetória comandada;

y é o valor medido da saída;

α é o coeficiente de suavização da trajetória de referência.

Pôde-se observar que um modelo é necessário para prever as saídas futuras da planta, saídas estas determinadas baseadas em valores passados e atuais e em ações de controle que satisfaçam as restrições e a função objetivo. Como o MPC não é uma técnica única, mas sim um conjunto de diferentes métodos, existem diversos tipos de modelos que podem ser utilizados dependendo da formulação.

2.4 Modelos

O coração do MPC está no modelo a ser utilizado para representar a dinâmica do processo, sendo este essencial para se realizar a determinação de entradas, estados e saídas futuras. Os modelos podem ser classificados por várias características, tais como, linear ou não-linear, contínuo ou discreto, determinístico ou estocástico, entrada-saída ou espaço de estados, representado no domínio do tempo ou da frequência e obtido através de princípios físicos ou métodos do tipo caixa-preta (empíricos) [1]. Uma observação importante é que a simulação de um modelo para fins de predição deve ser feita em tempo real, e portanto no ambiente MPC convém utilizar modelos discretos, de forma que a solução seja realizada computacionalmente.

A seguir, são apresentados os principais tipos de modelo de um processo, para o caso MPC [6].

2.4.1 Resposta ao Impulso

Um dos modelos mais utilizados na indústria é o modelo obtido através da resposta ao impulso, que pode ser obtida simplesmente medindo-se a saída do processo quando o mesmo é excitado por um pulso unitário de duração igual à de um período de amostragem na entrada. Este modelo, também conhecido como modelo de sequência de ponderação ou modelo de convolução, é amplamente aceito, pois é bastante intuitivo e pode ser utilizado no caso multivariável. Contudo o maior obstáculo é o grande número de parâmetros necessários e uma outra exigência é a de que o processo seja estável em malha aberta. Este modelo prediz a saída em um dado instante de tempo através da combinação linear de valores passados de entrada.

- Modelo Geral

$$y(t) = \sum_{i=1}^{\infty} h_i \cdot u(t-i) \quad (2.2)$$

onde h_i representam os pesos ou coeficientes da resposta ao impulso amostrada.

- Modelo Truncado

$$y(t) = \sum_{i=1}^M h_i \cdot u(t-i) \quad (2.3)$$

Obs.: Esta representação pode ser utilizada somente em processos assintoticamente

estáveis, pois $h_i \xrightarrow{i \rightarrow \infty} 0$ e assim para um certo M os termos com $i > M$ podem ser truncados,

ou seja o valor de M deve ser tal que $h_i \approx 0, i > M$, ou seja $\sum_{i=M+1}^{\infty} h_i u(t-i) \ll \sum_{i=1}^M h_i u(t-i)$.

Como mencionado anteriormente, uma inconveniência deste modelo é a necessidade de um número muito grande de parâmetros, pois M pode ser um número grande.

- Modelo de Predição

$$\hat{y}(t+k | t) = \sum_{i=1}^M h_i \cdot u(t+k-i | t) \quad (2.4)$$

onde $\hat{y}(t+k | t)$ é o valor estimado de y no instante $t+k$, conhecidas informações até o instante t .

$u(t+k-i | t)$ é o valor do controle no instante $t+k-i$, conhecidas informações até o instante t .

- Modelo Multivariável

Para um processo com j saídas e p entradas

$$y_q(t) = \sum_{l=1}^p \sum_{i=1}^M h^{l,q}_i \cdot u^l(t-i) \quad (2.5)$$

onde y_q é a q-ésima saída para $q = 1, 2, 3, \dots, j$

u^l é a l-ésima entrada para $l = 1, 2, 3, \dots, p$

h^{lq}_i representam os coeficientes da resposta impulso amostrada observadas na saída q quando o sistema é excitado pela entrada l .

Por fim, a grande vantagem deste método de modelagem está na não-necessidade de conhecimento das leis físicas que regem a planta, tornando o processo de identificação bastante intuitivo e ao mesmo tempo sendo capaz de identificar dinâmicas complexas, tais como fase não-mínima e atraso de transporte.

2.4.2 Resposta ao Degrau

Outro modelo bastante utilizado é o obtido através da resposta a um degrau. Este método de modelamento é bastante parecido com o descrito anteriormente, exceto pelo fato que aqui o processo deve ser excitado por um degrau. É utilizado no algoritmo DMC.

Matematicamente, a resposta ao degrau pode ser definida como a integral da resposta ao impulso, assim dado o modelo em uma dessas formas pode-se facilmente determinar o modelo na forma adjacente. Múltiplas saídas são tratadas por superposição. Através do uso do modelo baseado na resposta ao degrau pode-se escrever as mudanças nas saídas futuras como uma combinação linear das variações das entradas futuras. A matriz que relaciona essas variações da entrada com a saída é chamada de Matriz Dinâmica. Usando esta representação é possível calcular o vetor de transferência ótimo analiticamente, através da solução de um problema de mínimos quadrados. Na prática, a inversa da matriz requerida pode ser calculada “off-line” de forma a economizar esforço computacional. Somente a primeira coluna da matriz final de ganho precisará ser armazenada porque somente a primeira mudança necessita ser computada.

- Modelo Truncado

$$y(t) = y_0 + \sum_{i=1}^M g_i \cdot \Delta u(t-i) \quad (2.6)$$

onde g_i representam os pesos ou coeficientes da resposta ao degrau amostrada

y_0 representa o valor da saída em sua condição inicial.

Obs.: Esta representação pode ser utilizada somente em processos assintoticamente estáveis pois $g_{i+1} - g_i \xrightarrow{i \rightarrow \infty} 0$ assim para um certo M os termos com $i > M$ podem ser truncados, ou seja o valor de M deve ser tal que $g_{i+1} - g_i \approx 0, i > M$.

- Modelo de Predição

$$\hat{y}(t+k | t) = \sum_{i=1}^M g_i \cdot \Delta u(t+k-i | t) \quad (2.7)$$

Obs.: Nesta representação o valor de y_0 pode ser considerado como sendo 0 (zero) sem perda de generalidade.

As vantagens e desvantagens são as mesmas do modelo anterior.

2.4.3 Função de Transferência

O modelo de função de transferência é mais difundido na comunidade acadêmica e é utilizado na maioria dos métodos de projeto, pois é uma representação que requer alguns poucos parâmetros e é válido para processos estáveis e instáveis, porém algum conhecimento antecipado do processo é fundamental, especialmente a ordem dos polinômios A e B na equação abaixo.

- Modelo Geral

$$A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t) \quad (2.8)$$

onde $A(q^{-1}) = 1 + a_1 q^{-1} + a_2 q^{-2} + \dots + a_n q^{-n}$ (2.9)

$$B(q^{-1}) = 1 + b_1 q^{-1} + b_2 q^{-2} + \dots + b_m q^{-m} \quad (2.10)$$

- Modelo de Predição

$$\hat{y}(t+k | t) = \frac{B(q^{-1})}{A(q^{-1})} \cdot u(t+k | t) \quad (2.11)$$

2.4.4 Espaço de Estados

Este modelo permite o tratamento fácil para o caso multivariável, além da lei de controle ser a realimentação de uma combinação linear dos estados. Porém quando não se possui acesso a todos os estados, talvez haja a necessidade de se implementar um observador de estados, utilizando-se Filtro de Kalman, por exemplo.

Considere, então, um processo multivariável com j saídas e p entradas descrito através do seguinte sistema linear no domínio do tempo discreto:

$$\begin{aligned} \mathbf{x}(t+1) &= \mathbf{A} \cdot \mathbf{x}(t) + \mathbf{B} \cdot \ddot{\mathbf{u}}(t); \\ \mathbf{y}(t) &= \mathbf{C} \cdot \mathbf{x}(t) \end{aligned} \quad (2.12)$$

onde $\mathbf{x}(t)$ é o vetor de estados, e $\Delta \mathbf{u}(t) = \mathbf{u}(t) - \mathbf{u}(t-1)$.

A saída do modelo para o instante $t+k$, assumindo que os estados no instante t e os incrementos futuros do controle sejam conhecidos, pode ser calculada recursivamente com o auxílio das equações acima:

$$\mathbf{y}(t+k) = \mathbf{C} \cdot \mathbf{A}^j \cdot \mathbf{x}(t) + \sum_{i=0}^{j-1} \mathbf{C} \cdot \mathbf{A}^{j-i-1} \cdot \mathbf{B} \cdot \ddot{\mathbf{u}}(t+i) \quad (2.13)$$

Então a predição ótima k passos a frente é dada por:

$$\hat{\mathbf{y}}(t+k | t) = E[\mathbf{y}(t+k) | t] = \mathbf{C} \cdot \mathbf{A}^k \cdot E[\mathbf{x}(t) | t] + \sum_{i=0}^{k-1} \mathbf{C} \cdot \mathbf{A}^{k-i-1} \cdot \mathbf{B} \cdot \ddot{\mathbf{u}}(t+i) \quad (2.14)$$

Considerando um conjunto com predição de N passos:

$$\begin{bmatrix} \hat{\mathbf{y}}(t+1|t) \\ \hat{\mathbf{y}}(t+2|t) \\ \vdots \\ \hat{\mathbf{y}}(t+N|t) \end{bmatrix} = \begin{bmatrix} \mathbf{C} \cdot \mathbf{A} \cdot E[\mathbf{x}(t)|t] + \mathbf{C} \cdot \mathbf{B} \cdot \ddot{\mathbf{u}}(t) \\ \mathbf{C} \cdot \mathbf{A}^2 \cdot E[\mathbf{x}(t)|t] + \sum_{i=0}^1 \mathbf{C} \cdot \mathbf{B}^{1-i} \cdot \ddot{\mathbf{u}}(t+i) \\ \vdots \\ \mathbf{C} \cdot \mathbf{A}^N \cdot E[\mathbf{x}(t)|t] + \sum_{i=0}^{N-1} \mathbf{C} \cdot \mathbf{B}^{N-1-i} \cdot \ddot{\mathbf{u}}(t+i) \end{bmatrix} \quad (2.15)$$

a qual pode ser expressa como:

$$\hat{\mathbf{y}} = \mathbf{F} \cdot \hat{\mathbf{x}}(t) + \mathbf{H} \cdot \ddot{\mathbf{u}} \quad (2.16)$$

onde $\hat{\mathbf{x}}(t) = E[\mathbf{x}(t)|t]$, \mathbf{H} é uma matriz triangular inferior com seus elementos não nulos definidos por:

$$(\mathbf{H})_{ij} = \mathbf{C} \cdot \mathbf{A}^{i-j} \cdot \mathbf{B} \quad (2.17)$$

e a matriz \mathbf{F} é definida como:

$$\mathbf{F} = \begin{bmatrix} \mathbf{C} \cdot \mathbf{A} \\ \mathbf{C} \cdot \mathbf{A}^2 \\ \vdots \\ \mathbf{C} \cdot \mathbf{A}^N \end{bmatrix} \quad (2.18)$$

Considere, agora, um conjunto de previsões de k passos afetando a função objetivo:

$$\hat{\mathbf{y}}_{N_1, N_2} = \begin{bmatrix} \hat{\mathbf{y}}(t+N_1|t) \\ \hat{\mathbf{y}}(t+N_1+1|t) \\ \vdots \\ \hat{\mathbf{y}}(t+N_2|t) \end{bmatrix}, \quad (2.19)$$

onde $t+N_1$ é o primeiro passo da previsão, e $t+N_2$ é o último passo da previsão, assim $N = N_2 - N_1$.

$$\ddot{\mathbf{u}}_{Nu} = \begin{bmatrix} \ddot{\mathbf{u}}(t) \\ \ddot{\mathbf{u}}(t+1) \\ \vdots \\ \ddot{\mathbf{u}}(t+Nu-1) \end{bmatrix} \quad (2.20)$$

onde Nu é o horizonte de controle.

Então o modelo de predição pode ser escrito como:

$$\mathbf{y}_{(Nxj,1)} = \mathbf{F}_{(Nxj,l)} \cdot \hat{\mathbf{x}}(k)_{(l,1)} + \mathbf{H}_{(N,Nu)} \cdot \ddot{\mathbf{A}}\mathbf{u}_{(Nuxp,1)} \quad (2.21)$$

2.5 Estratégia de Controle

Nesta seção será apresentada a obtenção da estratégia de controle preditivo para dois modelos diferentes. O primeiro modelo apresentado é o da resposta finita ao degrau que é utilizado pelo algoritmo DMC, portanto aplicável somente a processos estáveis, já o segundo é o EE, que será utilizado ao longo do trabalho.

2.5.1 Resposta Finita ao Degrau

A seguir, estão descritas as formas como são feitos a predição, e o algoritmo de controle do DMC [6].

- Predição

$$\begin{aligned} \hat{\mathbf{y}}(t+k|t) &= \sum_{i=1}^{\infty} g_i \cdot \Delta u(t+k-i|t) + \hat{n}(t+k|t) = \\ &= \sum_{i=1}^k g_i \cdot \Delta u(t+k-i|t) + \sum_{i=k+1}^{\infty} g_i \cdot \Delta u(t+k-i|t) + \hat{n}(t+k|t) \end{aligned} \quad (2.22)$$

onde $\hat{n}(t+k|t)$ é a estimativa do erro de predição.

Supondo que a perturbação seja constante ao longo do horizonte de predição, tem-se:

$$\hat{n}(t+k|t) = \hat{n}(t|t) = y_p(t) - \hat{\mathbf{y}}(t|t) \quad (2.23)$$

onde: y_p é o valor medido da saída da planta.

Assim:

$$\begin{aligned}
\hat{y}(t+k | t) &= \sum_{i=1}^k g_i \cdot \Delta u(t+k-i) + \sum_{i=k+1}^{\infty} g_i \cdot \Delta u(t+k-i) + y_m(t) - \sum_{i=1}^{\infty} g_i \cdot \Delta u(t-i) = \\
&= \sum_{i=1}^k g_i \cdot \Delta u(t+k-i) + f(t+k)
\end{aligned} \tag{2.24}$$

onde $f(t+k)$ dado por:

$$f(t+k) = y_p(t) + \sum_{i=1}^{\infty} (g_{k+i} - g_i) \cdot \Delta u(t-i) \tag{2.25}$$

é a resposta livre do sistema (parte da resposta que não depende das ações futuras do controle).

Se o processo for assintoticamente estável, os coeficientes g_i da resposta ao degrau tendem a um valor constante após M períodos de amostragem, então pode-se considerar que:

$$g_{k+i} - g_i \approx 0, \quad i > M \tag{2.26}$$

de forma que a resposta livre pode ser calculada como:

$$f(t+k) = y_p(t) + \sum_{i=1}^M (g_{k+i} - g_i) \cdot \Delta u(t-i) \tag{2.27}$$

É interessante observar que para processos que não sejam assintoticamente estáveis, M não existe de forma que $f(t+k)$ não pode ser calculada.

Agora é possível realizar a predição da saída do processo para um horizonte de predição ($k = 1, \dots, N$), considerando Nu passos de controle.

$$\hat{y}(t+1 | t) = g_1 \cdot \Delta u(t) + f(t+1) \tag{2.28}$$

$$\hat{y}(t+2 | t) = g_2 \cdot \Delta u(t) + g_1 \cdot \Delta u(t+1) + f(t+2) \tag{2.29}$$

$$\hat{y}(t+3 | t) = g_3 \cdot \Delta u(t) + g_2 \cdot \Delta u(t+1) + g_1 \cdot \Delta u(t+2) + f(t+3) \tag{2.30}$$

\vdots

$$\hat{y}(t+Nu | t) = g_{Nu} \cdot \Delta u(t) + g_{Nu-1} \cdot \Delta u(t+1) + \dots + g_1 \cdot \Delta u(t+Nu-1) + f(t+Nu) \tag{2.31}$$

$$\hat{y}(t+Nu+1 | t) = g_{Nu+1} \cdot \Delta u(t) + g_{Nu} \cdot \Delta u(t+1) + \dots + g_1 \cdot \Delta u(t+Nu-1) + f(t+Nu+1) \tag{2.32}$$

$$\vdots$$

$$\hat{y}(t+N | t) = \sum_{i=N-Nu+1}^N g_i \cdot \Delta u(t+N-i) + f(t+N) \quad (2.33)$$

Definindo a matriz dinâmica do sistema \mathbf{G} , como sendo:

$$\mathbf{G} = \begin{bmatrix} g_1 & 0 & \cdots & 0 \\ g_2 & g_1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ g_{Nu} & g_{Nu-1} & \cdots & g_1 \\ \vdots & \vdots & \ddots & \vdots \\ g_N & g_{N-1} & \cdots & g_{N-Nu+1} \end{bmatrix} \quad (2.34)$$

pode-se reescrever o modelo de predição, como:

$$\hat{\mathbf{y}} = \mathbf{G} \cdot \ddot{\mathbf{A}}\mathbf{u} + \mathbf{f} \quad (2.35)$$

$$\text{onde: } \hat{\mathbf{y}} = \begin{bmatrix} \hat{y}_1(t+1) \\ \hat{y}_2(t+2) \\ \vdots \\ \hat{y}_N(t+N) \end{bmatrix}$$

$$\ddot{\mathbf{A}}\mathbf{u} = \begin{bmatrix} \ddot{A}u(t+1) \\ \ddot{A}u(t+2) \\ \vdots \\ \ddot{A}u(t+Nu) \end{bmatrix}$$

$$\mathbf{f} = \begin{bmatrix} f_1(t+1) \\ f_2(t+2) \\ \vdots \\ f_N(t+N) \end{bmatrix}$$

Observe que o número de colunas de \mathbf{G} está vinculado ao horizonte de controle, enquanto o número de linhas de \mathbf{G} está relacionado com o horizonte de predição. Assim o vetor $\Delta \mathbf{u}$ (incremento do controle) deve também possuir dimensão Nu (número de colunas de \mathbf{G}), e \mathbf{f} deve possuir N linhas.

- Algoritmo de Controle

O objetivo do controlador DMC é guiar a saída do processo tão perto quanto possível de sua trajetória de referência, no sentido de mínimos quadrados.

Considere a função objetivo:

$$J = \sum_{j=1}^N [\hat{y}(t+j|t) - r(t+j)]^2 + \sum_{j=1}^{Nu} \lambda \cdot [\Delta u(t+j-1)]^2, \lambda > 0 \quad (2.36)$$

Se não há restrições, a solução para a minimização da função objetivo pode ser obtida analiticamente, igualando-se a zero o gradiente de J em relação a Δu , o que gera o seguinte resultado:

$$\ddot{\mathbf{A}}\mathbf{u} = (\mathbf{G}^T \cdot \mathbf{G} + \ddot{\mathbf{e}} \cdot \mathbf{I})^{-1} \cdot \mathbf{G}^T \cdot (\mathbf{r} - \mathbf{f}) \quad (2.37)$$

Vale lembrar que em todas as estratégias preditivas, apenas o primeiro elemento do vetor $\Delta \mathbf{u}$ é realmente enviado à planta. Não é prudente implementar toda a sequência até o próximo intervalo Nu , porque é impossível estimar perfeitamente o vetor perturbação, e é também impossível antecipar precisamente perturbações que façam com que a saída difira da predição que foi usada para se calcular a sequência das ações do controle. Há também o descasamento entre planta e modelo, e o fato de que o ponto de operação pode se alterar ao longo deste intervalo de Nu passos.

Agora com relação às restrições, tem-se que, apesar de exigir um esforço computacional maior, a capacidade de lidar com estas torna o MPC um método bastante atrativo para aplicações práticas. Por razões de segurança, é necessário manter uma zona de segurança ao redor do ponto de operação, pois o efeito de uma perturbação pode fazer com que as restrições do processo sejam violadas. Esta zona pode ser reduzida e assim melhorar o lucro econômico, é claro, se o controlador estiver apto a lidar com estas restrições.

Contudo o problema, agora, é da forma padrão de uma formulação de programação quadrática, cuja otimização é feita numericamente por causa da presença das restrições.

Comercialmente este método é conhecido como Controle por Matriz Dinâmica utilizando Programação Quadrática (QDMC, Quadratic Dynamic Matrix Control).

2.5.2 Formulação no Espaço de Estados

Dado o modelo de predição descrito em 2.4.4, considere a função objetivo:

$$J = \ddot{\mathbf{a}} \cdot \left\| \hat{\mathbf{y}} - \mathbf{r} \right\|^2 + \ddot{\mathbf{e}} \cdot \left\| \ddot{\mathbf{A}}\mathbf{u} \right\|^2 \quad (2.38)$$

Utilizando (2.16) a função objetivo pode ser reescrita como:

$$J = \ddot{\mathbf{a}} \cdot \left[\mathbf{F} \cdot \hat{\mathbf{x}}(k) + \mathbf{H} \cdot \ddot{\mathbf{A}}\mathbf{u} - \mathbf{r} \right]^2 + \lambda \cdot \left\| \ddot{\mathbf{A}}\mathbf{u} \right\|^2 \quad (2.39)$$

e se não existem restrições, a lei de controle ótimo pode ser expressa como:

$$\ddot{\mathbf{A}}\mathbf{u} = (\mathbf{H}^T \cdot \mathbf{H} + \ddot{\mathbf{e}} \cdot \mathbf{I})^{-1} \cdot \ddot{\mathbf{a}} \cdot \mathbf{H}^T \cdot (\mathbf{r} - \mathbf{F} \cdot \hat{\mathbf{x}}(k)) \quad (2.40)$$

Observando o LMPC conclui-se que quando os horizontes de controle e de predição se aproximam do infinito e quando não existem restrições, obtém-se o problema do Regulador Linear Quadrático (LQR, Linear Quadratic Regulator), que foi estudado exhaustivamente nas décadas de 1960 e 1970.

Já problemas com restrições não permitem uma aplicação prática com horizonte de predição e controle com janela de dimensão infinita. Por outro lado escolhendo estas janelas com dimensão finita o problema pode ser resolvido de forma relativamente fácil e em tempo real a cada passo do sistema.

Uma das questões que nos vem à mente é sobre a aplicabilidade. Existem três pontos essenciais, o primeiro é que dependendo das restrições o problema de otimização pode tornar-se impraticável, o segundo é que uma função objetivo mal escolhida pode inadvertidamente

guiar o sistema em malha fechada para fora da região de operação, e por fim se o sistema for instável, então, em geral, o sistema não pode ser estabilizado globalmente, quando existe saturação da entrada [18, 26]. Por estes motivos antes de se implementar o MPC em algum processo é interessante realizar um pré-estudo do caso em particular.

2.6 Tratamento do erro em regime permanente no ambiente MPC

2.6.1 Introdução

Sabe-se que um controlador livre de erro em regime é aquele que leva as saídas controladas aos seus respectivos valores desejados em regime permanente. No ambiente de controle preditivo baseado em modelo linear, o controle para correção de erro em regime é usualmente alcançado adicionando-se perturbações constantes ao modelo do processo.

O desempenho em malha fechada de um algoritmo de controle baseado em modelo está diretamente relacionado à precisão do modelo. Na prática, erro de modelamento e perturbações não medidas podem levar a erro em regime a menos que precauções sejam tomadas no projeto de controle. A eliminação do erro em regime pode ser consumada por dois caminhos básicos. O primeiro método seria modificar a função objetivo do controlador para incluir a integração do erro de rastreamento. Este método, empregado por algoritmos de controle PID, pode também ser usado no ambiente Linear Quadrático Gaussiano (LQG, Linear Quadratic Gaussian) ou MPC. No ambiente MPC, o termo integral é incorporado através do aumento da dimensão do modelo do processo com os estados de erro de rastreamento. Para sistemas de larga escala, este aumento pode significar um aumento do custo computacional da otimização dinâmica que cresce em proporção ao cubo da dimensão dos estados [30]. Outra desvantagem desta abordagem é a necessidade de um algoritmo “anti-windup” para o termo integral para prevenir uma penalidade de desempenho não necessária [28].

Uma segunda abordagem geral para eliminar erro em regime seria aumentar o modelo do processo para incluir uma perturbação degrau constante. Esta perturbação, que é estimada através das variáveis medidas do processo, é geralmente assumida permanecer constante no futuro e seu efeito nas variáveis controladas é removido deslocando a posição em regime permanente do controlador. Embora este método elimine a necessidade de utilização de um algoritmo “anti-windup”, ele possui a desvantagem de requerer que um modelo de perturbação seja projetado e que a perturbação dos estados seja estimada em separado.

2.6.2 Modelos Independente e Realinhado

Algumas plantas contêm integradores, tipicamente se a saída é o nível de um tanque contendo líquido, ou a posição ou atitude de um objeto inercial como uma aeronave. Estes exemplos não são assintoticamente estáveis, porém um controlador preditivo básico ainda pode ser aplicado a plantas que sejam “marginalmente estáveis”.

Entretanto, se a planta é instável (para o caso contínuo deve haver pólos no semi-plano direito, e para o caso discreto deve haver pólos fora do círculo de raio unitário), sua resposta ao impulso ou ao degrau não é limitada, então um controlador preditivo básico já não funciona. O problema é precisamente o uso do modelo independente, porque isto significa que o modelo está sendo “rodado” em malha aberta. Se o modelo é instável e corre em malha aberta, então seu comportamento irá divergir rapidamente amplificado pela instabilidade, e seu valor como preditor do comportamento da planta será perdido [20].

A única solução neste caso é estabilizar o modelo de alguma forma, e isto requer que a independência do modelo seja deixada de lado. A maneira mais fácil de estabilizar o modelo é realinhá-lo à saída da planta. Isto é, quando simulado para gerar as previsões do comportamento da planta, as condições iniciais levadas em conta devem ser as mesmas da

planta, e não as resultantes do passo anterior da simulação. Suponha que o modelo instável corresponda à equação a diferenças:

$$y_m(k) = -\sum_{i=1}^M a_i \cdot y_m(k-i) + \sum_{i=1}^M b_i \cdot u(k-i) \quad (2.41)$$

então o modelo realinhado gera previsões de acordo com:

$$\hat{y}(k+1|k) = -\sum_{i=1}^M a_i \cdot y_p(k+1-i) + b_1 \cdot \hat{u}(k|k) + \sum_{i=2}^M b_i \cdot u(k+1-i) \quad (2.42)$$

$$\begin{aligned} \hat{y}(k+2|k) = & -a_1 \cdot \hat{y}(k+1|k) - \sum_{i=2}^M a_i \cdot y_p(k+2-i) + b_1 \cdot \hat{u}(k+1|k) + \\ & + b_2 \cdot \hat{u}(k|k) + \sum_{i=3}^M b_i \cdot u(k+2-i) \end{aligned} \quad (2.43)$$

Isto é, os últimos valores de entrada e saída da planta são utilizados sempre que estiverem disponíveis. Este procedimento sempre estabiliza o modelo, e pode ser interpretado como a implementação de um observador “deadbeat” para a planta, assim o modelo não diverge muito da planta. Porém este procedimento nem sempre garante estabilidade em malha fechada. Note que (2.42) introduz uma realimentação no controlador preditivo.

Agora um obstáculo para implementação do modelo realinhado torna-se aparente, pois o tratamento do erro em regime não pode ser feito simplesmente introduzindo-se um termo referente a perturbação no modelo de predição, e a razão é que a afirmação:

$$y_{m\infty} - f = 0 \quad (2.44)$$

não é mais válida para esta situação, porque agora o modelo não é mais guiado somente pela entrada $u(k)$, mas também pela saída da planta $y_p(k)$, que age como uma entrada para o modelo.

Mas ainda é possível obter um rastreamento sem erro em regime com um modelo realinhado, através da construção explícita de uma ação integral no controlador.

2.6.3 Rastreamento livre de erro em regime para modelos independentes (sem realinhamento)

Se o modelo não é o mesmo da planta, em particular se o ganho em regime permanente do modelo não está correto, então a saída da planta irá atingir um valor final incorreto.

Felizmente, é fácil modificar o controlador preditivo para fazê-lo insensível a erros de ganho de regime permanente. Para isso é necessário medir a discrepância entre a última saída da planta e a última saída do modelo, e então subtrair esta diferença da trajetória de referência nos pontos coincidentes [6]. Isto é, definindo:

$$d(k) = y(k) - \hat{y}(k | k-1) \quad (2.45)$$

Então a antiga lei de controle:

$$\ddot{\mathbf{A}}\mathbf{u} = \mathbf{K}(\mathbf{r} - \mathbf{f}) \quad (2.46)$$

onde $\mathbf{K} = (\mathbf{G}^T \cdot \mathbf{G} + \ddot{\mathbf{e}} \cdot \mathbf{I})^{-1} \cdot \mathbf{G}^T$

torna-se

$$\ddot{\mathbf{A}}\mathbf{u} = \mathbf{K}(\mathbf{r} - \mathbf{d} - \mathbf{f}) \quad (2.47)$$

quando levada em consideração a perturbação.

Este procedimento funciona para plantas e modelos assintoticamente estáveis pela seguinte razão. Seja o ganho em regime da planta $S_p(\infty)$, e o ganho em regime do modelo $S_m(\infty)$. Também assuma que a malha fechada é assintoticamente estável, e que o ponto de operação é uma constante $s(k) = s_\infty$. Então a entrada, e ambas as saídas da planta e do modelo são valores constantes, ou seja $u(k) \rightarrow u_\infty$, $y_p(k) \rightarrow y_{p\infty}$ e $y_m(k) \rightarrow y_{m\infty}$, onde y_p e y_m representam as saídas da planta e do modelo respectivamente. Então $y_{p\infty} = S_p(\infty) \cdot u_\infty$ e $y_{m\infty} = S_m(\infty) \cdot u_\infty$, e

$$d(k) \rightarrow d_\infty = [S_p(\infty) - S_m(\infty)] \cdot u_\infty \quad (2.48)$$

Em condições de regime, uma vez que a trajetória comandada e a saída da planta são constantes, a trajetória de referência não depende de k , isto é $r(k+i|k) = r_i$. Então o vetor referência é um vetor constante. Além disso, o vetor da resposta livre também é um vetor constante, já que a resposta livre é sempre calculada a partir da mesma condição inicial, uma vez alcançada a condição de regime.

Sendo:

$$\mathbf{G} \cdot \ddot{\mathbf{u}} = \mathbf{y} - \mathbf{f} \quad (2.49)$$

calcula-se $\Delta \mathbf{u}$ substituindo-se “ \mathbf{y} ” por “ $\mathbf{r} - \mathbf{d}$ ”, ou seja considera-se a saída como sendo a referência deslocada de um certo distúrbio, então tem-se em regime que:

$$\mathbf{G} \cdot \ddot{\mathbf{u}} = \mathbf{r} - \mathbf{d} - \mathbf{f} = \mathbf{r} - \mathbf{y}_{p\infty} + \mathbf{y}_{m\infty} - \mathbf{f} \quad (2.50)$$

Mas note que:

$$\mathbf{y}_{m\infty} - \mathbf{f} = \mathbf{0} \quad (2.51)$$

Pois estamos assumindo condição de regime permanente, então os valores iniciais da resposta livre permanecem constantes. Portanto conclui-se que:

$$\mathbf{G} \cdot \ddot{\mathbf{u}} = \mathbf{r} - \mathbf{y}_{p\infty} \quad (2.52)$$

Mas em regime $\Delta \mathbf{u} = \mathbf{0}$, portanto $\mathbf{r} = \mathbf{y}_{p\infty}$. Então a trajetória de referência é constante, com valor $y_{p\infty}$. Mas de (2.1) $r_i = s_{\infty} - \alpha^i * (s_{\infty} - y_{p\infty})$, portanto conclui-se que:

$$y_{p\infty} = s_{\infty} \quad (2.53)$$

que mostra que a saída da planta atinge o valor de operação, mesmo na presença do erro de modelamento.

Note que esta análise assume malha fechada estável, algo que nem sempre é garantido.

Este mesmo procedimento também fornece rastreamento em regime sem erro para um ponto de operação constante, se existir uma perturbação desconhecida que seja aditiva e constante, e que atue ou na entrada ou na saída da planta, desde que ambas tenham o mesmo

efeito como um ganho de regime incorreto do modelo. De fato, o uso de (2.47) pode corrigir o erro entre as saídas atuais da planta e do modelo quando existir uma perturbação agindo sobre a saída da planta, e assumindo que esta perturbação continuará ao longo do horizonte de predição. Este método para se obter rastreamento livre de erro em regime é utilizado na maioria dos aplicativos comerciais de controle preditivo. E de maneira simples, pode-se dizer que este erro em regime é eliminado pelo fato de se inserir um integrador discreto junto à lei de controle.

O uso da perturbação estimada (2.45) é de fato o único meio de realimentação da lei de controle preditivo, e usando-a em conjunto com (2.47) corresponde a assumir que a estimativa da perturbação atual irá permanecer no mesmo nível no futuro. Esta não é a única suposição que pode ser feita; por exemplo, pode-se assumir que a perturbação morre de forma exponencial com uma constante de tempo conhecida, apesar de que para este caso a propriedade de rastreamento livre de erro em regime seria perdida.

2.6.4 Tratamento do erro em regime para modelos instáveis em malha aberta (com realinhamento)

As implementações industriais de MPC mais comumente encontradas na indústria, tais como IDCOM, DMC, QDMC e IDCOM-M usam um modelo de perturbação constante para eliminar o erro em regime do controle. Porém, apesar deste método ser aceitável para plantas estáveis, ele não funciona para sistemas instáveis. E até mesmo para sistemas estáveis, este método pode levar a um desempenho inaceitável se os sensores são ruidosos ou se as perturbações entram na planta por outro lugar. Este caso levou ao desenvolvimento do algoritmo de MPC industrial, denominado Controlador Otimizador Multivariável da Shell (SMOC, Shell Multivariable Optimizing Controller), que permite modelos de perturbação em espaço de estados [28].

– Modelo de perturbação não medida

Um modelo com perturbação constante na saída pode ser escrito utilizando o seguinte modelo em EE:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{p}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{p}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{u}_k, \\ \mathbf{y}_k &= \begin{bmatrix} \mathbf{C} & \mathbf{G}_p \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{p}_k \end{bmatrix} \end{aligned} \quad (2.54)$$

onde $\mathbf{p} \in \mathbb{R}^{S_p}$, S_p é o número de estados perturbados aumentados, e \mathbf{G}_p determina o efeito destes estados na saída. Em uma implementação industrial padrão de MPC, $\mathbf{G}_p = \mathbf{I}$. O resultado é um controlador deadbeat para os estados perturbados de saída e um observador malha aberta para os estados do modelo.

Este modelo de perturbação na saída é simples de implementar, porém pode levar a um desempenho ruim quando a perturbação entra em outro ponto da malha.

O valor de \mathbf{p}_0 é obtido através da medição do erro em regime, quando o sistema é regulado por um controlador LMPC que não utiliza o modelo de perturbação acima mostrado. Este valor permanece constante até o instante em que o valor da saída estiver suficientemente próximo da referência. A partir deste instante o valor de \mathbf{p}_k passa a ser uma constante mais o erro entre valor medido e valor predito.

Prett e Garcia [15] apresentaram um método alternativo para obter um controle livre de erro em regime utilizando um modelo de espaço de estado na forma de velocidade. Nesta abordagem, o estado e entrada representam a mudança no estado e entrada originais e os estados adicionais acrescentados ao modelo são saídas do sistema original.

$$\begin{aligned} \begin{bmatrix} \ddot{\mathbf{A}}\mathbf{x}_{k+1} \\ \mathbf{z}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{C} \cdot \mathbf{A} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\mathbf{A}}\mathbf{x}_k \\ \mathbf{z}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{C} \cdot \mathbf{B} \end{bmatrix} \cdot \ddot{\mathbf{A}}\mathbf{u}_k, \\ \mathbf{y}_k &= \begin{bmatrix} \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \ddot{\mathbf{A}}\mathbf{x}_k \\ \mathbf{z}_k \end{bmatrix} \end{aligned} \quad (2.55)$$

A principal vantagem deste método é que não há necessidade de se calcular alvos em regime permanente pois $\Delta \mathbf{x}$ e $\Delta \mathbf{u}$ aproximam-se da origem quando em regime (problema de regulação). E a principal desvantagem é que a dimensão dos estados é aumentada na otimização dinâmica o que aumenta o custo computacional. O tempo de computação necessário na otimização dinâmica é sempre um fator limitante em aplicações MPC.

Outra alternativa é aumentar o modelo do sistema com uma perturbação na entrada ou nos estados. Davison e Smith [13] apresentaram uma formulação de perturbação nos estados como uma técnica padrão para remover erro em regime para o regulador linear quadrático. A perturbação no estado pode ser escrita como mostra-se a seguir:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{d}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{G}_d \\ \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{u}_k, \\ \mathbf{y}_k &= [\mathbf{C} \quad \mathbf{0}] \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}_k \end{bmatrix} \end{aligned} \quad (2.56)$$

onde $\mathbf{d} \in \mathbb{R}^{S_d}$, S_d é o número de estados perturbados aumentados, e \mathbf{G}_d determina o efeito da perturbação. Perturbações de entrada podem ser representadas neste ambiente ajustando-se $\mathbf{G}_d = \mathbf{B}$.

Para sistemas lineares, um modelo de perturbação em bloco diagonal geral que inclui ambas perturbações de estado e de saída é:

$$\begin{aligned} \begin{bmatrix} \mathbf{x}_{k+1} \\ \mathbf{d}_{k+1} \\ \mathbf{p}_{k+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{A} & \mathbf{G}_d & \mathbf{0} \\ \mathbf{0} & \mathbf{I} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{I} \end{bmatrix} \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}_k \\ \mathbf{p}_k \end{bmatrix} + \begin{bmatrix} \mathbf{B} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \cdot \mathbf{u}_k, \\ \mathbf{y}_k &= [\mathbf{C} \quad \mathbf{0} \quad \mathbf{G}_p] \cdot \begin{bmatrix} \mathbf{x}_k \\ \mathbf{d}_k \\ \mathbf{p}_k \end{bmatrix} \end{aligned} \quad (2.57)$$

2.7 MPC Restrito

2.7.1 Introdução

Sabe-se que a formulação de um problema de controle sem restrições não é algo muito realístico, pois os atuadores possuem uma faixa limite de operação e uma taxa de variação limitada, além dos sensores que também atuam dentro de uma faixa, gerando limites máximos e mínimos nas variáveis do processo. Como os sistemas de controle normalmente operam próximos a seus limites, violações destas restrições podem muito provavelmente ocorrer. Por isso sistemas de controle, principalmente os com grande horizonte de predição, devem se antecipar a violações de restrição e corrigir a “trajetória” das variáveis do processo para um caminho apropriado. Contudo as restrições de saída, que aparecem geralmente por razões de segurança, devem ser controladas com antecedência levando-se em conta a dinâmica do processo. Vale notar que as variáveis de entrada ou manipuladas podem sempre ser mantidas em seu limite pelo controlador.

2.7.2 MPC e Restrições

Como visto anteriormente, as ações de controle no cenário MPC são calculadas através da determinação do vetor $\Delta \mathbf{u}$ que minimiza uma função objetivo quadrática do tipo (2.38), que de forma expandida origina:

$$J(\Delta \mathbf{u}) = \ddot{\mathbf{A}}\mathbf{u}^T (\mathbf{H}^T \mathbf{H} + \mathbf{I}) \ddot{\mathbf{A}}\mathbf{u} + \ddot{\mathbf{A}}\mathbf{u}\mathbf{H}^T (\mathbf{F}\mathbf{x} - \mathbf{r}) + (\mathbf{F}\mathbf{x})^T (\mathbf{F}\mathbf{x}) - \mathbf{r}^T \mathbf{F}\mathbf{x} + \mathbf{r}^T \mathbf{r} \quad (2.58)$$

ou ainda,

$$J(\Delta \mathbf{u}) = \frac{1}{2} \cdot \ddot{\mathbf{A}}\mathbf{u}^T \cdot \mathbf{S}_c \cdot \ddot{\mathbf{A}}\mathbf{u} + \mathbf{f}_1 \cdot \ddot{\mathbf{A}}\mathbf{u} + \mathbf{f}_0 \quad (2.59)$$

que possui o termo quadrático em $\Delta \mathbf{u}$, outra que pondera o termo simples de $\Delta \mathbf{u}$, e por fim o termo que não é influenciado pela variação do controle, e cuja solução ótima é encontrada resolvendo-se a equação linear:

$$\mathbf{S}_c \cdot \ddot{\mathbf{u}} = -\mathbf{f}_1 \quad (2.60)$$

Porém, se $u(t)$ viola a restrição, seu valor satura, ou pelo programa de controle ou pelo atuador.

Esta forma de operação não garante que o valor ótimo será obtido quando as restrições são violadas. Assim o principal propósito do MPC, que é aplicar o melhor sinal de controle possível que minimize a função objetivo acima, não será alcançado.

Para ilustrar, considere os casos de violação de restrição, mostrado na figura 2.2 que ilustra os valores de maior variação do controle e as curvas de nível da função objetivo J , de um problema MPC com horizonte de controle igual a dois. A figura 2.2a mostra o caso onde $\Delta u(t) > \Delta u_{\max}$. Neste caso o modo normal de operação seria aplicar Δu_{\max} ao processo, ao invés de Δu_c onde o mínimo de J é alcançado quando a restrição é considerada. No caso da figura 2.2b, $\Delta u(t)$ não viola as restrições, e seria aplicado ao sistema ao invés do sinal ótimo Δu_c que deveria ser aplicado quando as restrições são levadas em conta.

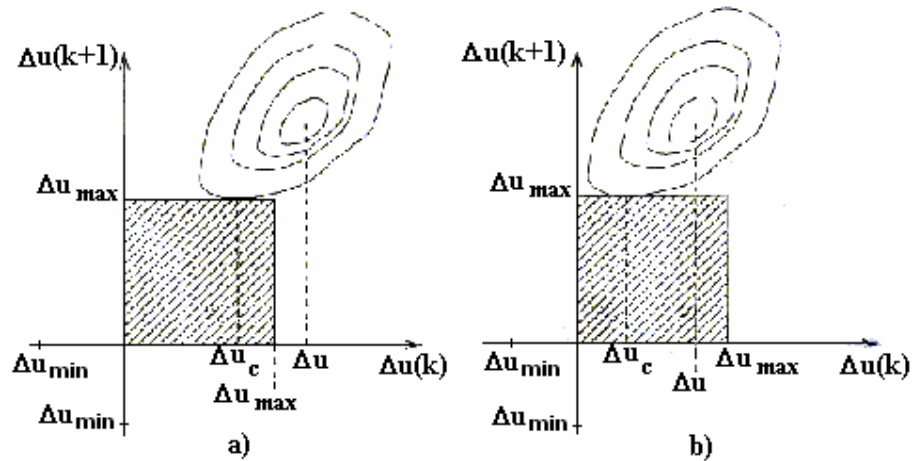


Figura 2.2: Restrições no sinal de controle.

A não-consideração das restrições nas variáveis manipuladas em sua total extensão, pode fazer com que a função objetivo assuma altos valores, gerando um sistema de controle de baixo desempenho. Contudo, as variáveis manipuladas podem quase sempre ser mantidas em seus limites ou pelo programa de controle, ou pelo atuador, e esta não é a principal razão pela qual as restrições devem ser tratadas de um modo apropriado. Violar os limites das variáveis controladas pode ser mais custoso e perigoso, pelo fato que esta violação pode causar danos ao equipamento e perda da produção.

Violações de restrição nas variáveis de saída não são contempladas quando o único modo de tratamento da restrição é através da redução de valores das variáveis de entrada. Uma das principais vantagens do MPC, sua capacidade de predição, não é usada em seu potencial total através deste modo de operação. Sistemas de controle, especialmente os de grande horizonte de predição, deveriam se antecipar a violações de restrição e corrigi-las de modo apropriado.

As restrições de um processo, geralmente, estão relacionadas com o limite de amplitude no sinal de controle, limite de taxa de variação do atuador e limite do sinal de saída, e podem ser descritas respectivamente por:

$$\begin{aligned} U_{\min} &\leq u(t) \leq U_{\max} & \forall t \\ \Delta U_{\min} &\leq u(t) - u(t-1) \leq \Delta U_{\max} & \forall t \\ Y_{\min} &\leq y(t) \leq Y_{\max} & \forall t \end{aligned} \quad (2.61)$$

Para um processo com p entradas, j saídas e restrições agindo sobre um horizonte de predição de tamanho N , estas restrições podem ser expressas como [6]:

$$\begin{aligned} \mathbf{1} \cdot U_{\min} &\leq \mathbf{T} \cdot \ddot{\mathbf{A}}\mathbf{u} + \mathbf{1} \cdot u(t-1) \leq \mathbf{1} \cdot U_{\max} \\ \mathbf{1} \cdot \Delta U_{\min} &\leq \ddot{\mathbf{A}}\mathbf{u} \leq \mathbf{1} \cdot \Delta U_{\max} \\ \mathbf{1} \cdot Y_{\min} &\leq \mathbf{G}\ddot{\mathbf{A}}\mathbf{u} + \mathbf{f} \leq \mathbf{1} \cdot Y_{\max} \end{aligned} \quad (2.62)$$

onde $\mathbf{1}$ é uma matriz $(N \times j) \times p$ formada por N matrizes identidades $p \times p$, e \mathbf{T} é uma matriz bloco triangular inferior cujos blocos não-nulos são matrizes identidades $p \times p$.

Outros tipos de restrição podem ser incluídas nas variáveis manipuladas do processo para forçar a resposta do sistema a possuir determinadas características, e podem ser expressas de maneira similar.

2.7.2.1 Restrição da Taxa de Variação

Devido ao fato do modelo já ser expresso sob a forma da variação do controle, torna-se relativamente simples a formulação desta restrição, basta para isso considerar:

$$\Delta u_{\min} \leq \Delta u(k) \leq \Delta u_{\max} \quad \text{para } Nu = 1 \quad (2.63)$$

$$\mathbf{1} \cdot \Delta u_{\min} \leq \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+Nu-1) \end{bmatrix} \leq \mathbf{1} \cdot \Delta u_{\max} \quad \text{para } Nu \text{ geral} \quad (2.64)$$

ou ainda:

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+Nu-1) \end{bmatrix} \leq \mathbf{1} \cdot \Delta u_{\max} \quad (2.65)$$

e

$$\begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+Nu-1) \end{bmatrix} \geq \mathbf{1} \cdot \Delta u_{\min} \Leftrightarrow (-1) \cdot \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+Nu-1) \end{bmatrix} \leq (-1) \cdot \mathbf{1} \cdot \Delta u_{\min} \quad (2.66)$$

que de forma compacta pode ser escrito como:

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}_{2Nu \times Nu} \ddot{\mathbf{u}}_{Nu \times 1} \leq \begin{bmatrix} \ddot{\mathbf{u}}_{\max} \\ -\ddot{\mathbf{u}}_{\min} \end{bmatrix}_{2Nu \times 1} \quad (2.67)$$

onde \mathbf{I} é matriz identidade de ordem $N \times Nu$;

Δu_{\max} é o maior valor de variação do controle;

Δu_{\min} é o menor valor de variação do controle.

2.7.2.2 Restrição de Amplitude de Entrada

O tratamento da restrição da amplitude de entrada pode ser feito da seguinte maneira.

Deseja-se que o valor de controle esteja restrito a uma faixa de atuação dada por U_{\max} e U_{\min} .

Assim matematicamente, tem-se:

$$U_{\min} \leq u(k) \leq U_{\max} \quad \text{para } Nu = 1 \quad (2.68)$$

$$\mathbf{1} \cdot U_{\min} \leq \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+Nu-1) \end{bmatrix} \leq \mathbf{1} \cdot U_{\max} \quad \text{para } Nu \text{ geral} \quad (2.69)$$

Porém as restrições devem ser escritas com relação à variação do controle (Δu). Então:

$$\begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+Nu-1) \end{bmatrix} = \begin{bmatrix} u(k) - u(k-1) + u(k-1) \\ u(k+1) - u(k) + u(k) \\ \vdots \\ u(k+Nu-1) - u(k+Nu-2) + u(k+Nu-2) \end{bmatrix} \quad (2.70)$$

$$= \begin{bmatrix} \Delta u(k) + u(k-1) \\ \Delta u(k+1) + u(k) \\ \vdots \\ \Delta u(k+Nu-1) + u(k+Nu-2) \end{bmatrix} \quad (2.71)$$

Portanto:

$$\mathbf{1} \cdot U_{\min} \leq \begin{bmatrix} \Delta u(k) + u(k-1) \\ \Delta u(k+1) + u(k) \\ \vdots \\ \Delta u(k+Nu-1) + u(k+Nu-2) \end{bmatrix} \leq \mathbf{1} \cdot U_{\max} \quad (2.72)$$

subtraindo-se $u(k-1)$ em todos os termos, tem-se:

$$\mathbf{1} \cdot [U_{\min} - u(k-1)] \leq \begin{bmatrix} \Delta u(k) \\ u(k+1) - u(k-1) \\ \vdots \\ u(k+Nu-1) - u(k-1) \end{bmatrix} \leq \mathbf{1} \cdot [U_{\max} - u(k-1)] \quad (2.73)$$

Como:

$$u(k+1) - u(k-1) = \Delta u(k) + \Delta u(k+1) = u(k) - u(k-1) + u(k+1) - u(k) \quad (2.74)$$

$$u(k+2) - u(k-1) = \Delta u(k) + \Delta u(k+1) + \Delta u(k+2) \quad (2.75)$$

Pode-se, então, reescrever a restrição como:

$$\mathbf{1} \cdot [U_{\min} - u(k-1)] \leq \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \Delta u(k+2) \end{bmatrix} \leq \mathbf{1} \cdot [U_{\max} - u(k-1)] \quad (2.76)$$

ou ainda, de forma compacta:

$$\mathbf{1} \cdot [U_{\min} - u(k-1)] \leq \mathbf{T} \cdot \ddot{\mathbf{u}} \leq \mathbf{1} \cdot [U_{\max} - u(k-1)] \quad (2.77)$$

Por fim

$$\begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix}_{2NxNu} \ddot{\mathbf{u}}_{Nu \times 1} \leq \begin{bmatrix} \mathbf{U}_{\max} - \mathbf{u}(k-1) \\ \mathbf{U}_{\min} - \mathbf{u}(k-1) \end{bmatrix}_{2Nx1} \quad (2.78)$$

onde: \mathbf{T} é matriz triangular, de ordem $N \times Nu$, com a parte inferior a diagonal principal cheia

de 1's e a parte superior cheia de 0's;

U_{\max} é o maior valor de controle;

U_{\min} é o menor valor de controle;

2.7.2.3 Restrição da Amplitude de Saída

O tratamento referente às restrições de amplitude da saída pode ser feito da seguinte forma. Dado o modelo de predição (2.16), deseja-se que:

$$Y_{\min} \leq \hat{\mathbf{y}} \leq Y_{\max} \quad (2.79)$$

ou ainda

$$Y_{\min} \leq \mathbf{F} \hat{\mathbf{x}}(t) + \mathbf{H} \ddot{\mathbf{A}} \mathbf{u} \leq Y_{\max} \quad (2.80)$$

$$Y_{\min} - \mathbf{F} \hat{\mathbf{x}}(t) \leq \mathbf{H} \ddot{\mathbf{A}} \mathbf{u} \leq Y_{\max} - \mathbf{F} \hat{\mathbf{x}}(t) \quad (2.81)$$

então

$$\begin{bmatrix} \mathbf{H} \\ -\mathbf{H} \end{bmatrix}_{2N \times Nu} \ddot{\mathbf{A}} \mathbf{u}_{Nu \times 1} \leq \begin{bmatrix} Y_{\max} - \mathbf{F} \hat{\mathbf{x}}(t) \\ Y_{\min} - \mathbf{F} \hat{\mathbf{x}}(t) \end{bmatrix}_{2N \times 1} \quad (2.82)$$

onde: \mathbf{H} é matriz de predição do sistema, de ordem $N \times Nu$;

Y_{\max} é o maior valor de saída;

Y_{\min} é o menor valor de saída.

2.7.2.4 Matriz Restrição

De forma resumida temos que todas as restrições podem, então, ser expressas na forma de restrição de desigualdade:

$$\underbrace{\begin{bmatrix} \mathbf{H} \\ -\mathbf{H} \\ \mathbf{I} \\ -\mathbf{I} \\ \mathbf{T} \\ -\mathbf{T} \end{bmatrix}}_{\mathbf{Re} \quad (4N+2Nu) \times Nu} \ddot{\mathbf{A}} \mathbf{u}_{Nu \times 1} \leq \underbrace{\begin{bmatrix} Y_{\max} - \mathbf{F} \hat{\mathbf{x}}(t) \\ Y_{\min} - \mathbf{F} \hat{\mathbf{x}}(t) \\ \ddot{\mathbf{A}} \mathbf{u}_{\max} \\ -\ddot{\mathbf{A}} \mathbf{u}_{\min} \\ \mathbf{U}_{\max} - \mathbf{u}(k-1) \\ \mathbf{U}_{\min} - \mathbf{u}(k-1) \end{bmatrix}}_{\mathbf{b} \quad (4N+2Nu) \times 1} \quad (2.83)$$

ou ainda, de forma compacta:

$$\mathbf{Re} \cdot \ddot{\mathbf{A}} \mathbf{u} \leq \mathbf{b} \quad (2.84)$$

Capítulo 3

Estratégia MPC adotada para controle do levitador magnético

3.1 Introdução

O objetivo do controle é estabilizar a saída da planta, e fazer com que a mesma siga uma determinada trajetória de referência do tipo degrau, sem apresentar erro em regime. Neste contexto, dada a planta e especificados os índices de desempenho propõe-se uma solução fundamentada nos princípios de controle preditivo, objetivando as considerações de estabilidade e de regime permanente, não considerando especificações de transitório.

Nesta seção será apresentado inicialmente o tratamento matemático aplicado ao modelo não-linear da planta, para sua utilização de forma adequada no ambiente dos controladores preditivos. É interessante lembrar que o modelo utilizado neste trabalho foi obtido a partir dos princípios físicos envolvidos no processo de levitação magnética [17].

O primeiro passo é o de linearizar o modelo, para então escrevê-lo na forma de espaço de estados contínuos, e em seguida aplicar-se algum método de discretização. O modelo deve também ser escrito em função da variação de controle Δu .

Por fim são apresentados os tratamentos das restrições, e as superfícies de algumas funções objetivo, onde é possível observar a mudança na característica de estabilidade entre os sistemas em malha aberta, e com o controlador LMPC.

O controlador preditivo minimiza uma função objetivo quadrática, composta pelo quadrado da diferença entre o valor predito e a referência mais o quadrado da variação do

controle, sendo ambos os termos influenciados por pesos, delta (δ) o peso referente ao erro de predição e lambda (λ) o peso referente a variação do controle:

$$J = \min_{\Delta u(.)} \left[\ddot{\mathbf{a}} \cdot \left\| \hat{\mathbf{y}} - \mathbf{r} \right\|^2 + \ddot{\mathbf{e}} \cdot \left\| \Delta \mathbf{u} \right\|^2 \right] \quad (3.1)$$

onde \mathbf{r} é um vetor com N elementos referente trajetória de referência;

$\hat{\mathbf{y}}$ é a predição de uma sequência de saída, com N elementos, a partir do instante t ;

$\Delta \mathbf{u}$ é um vetor com Nu elementos referente às variações do controle.

A expressão matemática utilizada para se determinar a trajetória de referência está descrita na seção 2.3, equação (2.1).

3.2 Sistema de Levitação Magnética da Feedback Instruments

A figura 3.1 ilustra o sistema de levitação magnética da Feedback Instruments, e a figura 3.2 mostra o diagrama de blocos deste mesmo sistema. O SLM pode ser sub-dividido basicamente em três blocos, sendo dois deles lineares (amplificador de corrente e sensor ótico) e somente um terceiro não-linear (eletroímã). Na figura 3.2 existe ainda a referência a um quarto bloco, que representa o controlador preditivo, cujas características principais são discutidas no capítulo seguinte.



Figura 3.1: Foto ilustrativa do SLM da Feedback Instruments.

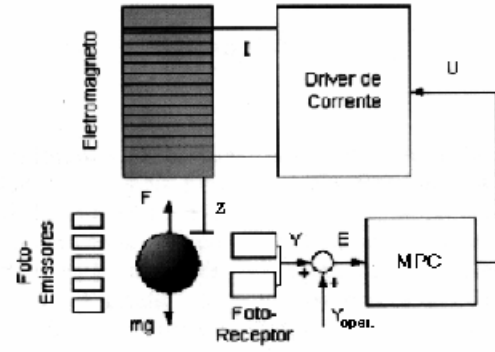


Figura 3.2: Diagrama de blocos do SLM [21].

Sabe-se do manual do fabricante [21] que a equação que rege a magnitude da força de atração magnética de nosso sistema de levitação magnética (SLM) é:

$$F_m = k_c \cdot \frac{I^2}{Z^2} \quad (3.2)$$

com [17]:

$$k_c = 2.78 \cdot 10^{-4} \cdot \exp(-32.5 \cdot Z) \quad (3.3)$$

onde k_c é o coeficiente de conversão eletromecânica (Nm^2/A^2);

I é a corrente no eletroímã (A);

Z é a distância da esfera ao eletroímã (m).

De acordo com a segunda lei de Newton aplicada à resultante das forças na direção vertical, tem-se que:

$$m \cdot a = \sum F \quad (3.4)$$

cuja somatória de forças verticais do sistema de levitação magnética é:

$$\sum F = P - F_m = mg - k_c \frac{I^2}{Z^2} \quad (3.5)$$

o que nos leva a equação diferencial de segunda ordem:

$$m \cdot \frac{d^2 Z}{dt^2} = mg - k_c \frac{I^2}{Z^2} \quad (3.6)$$

onde m é a massa da esfera, cujo valor é 0.0212kg;

g é a aceleração da gravidade, cujo valor utilizado foi 9.8m/s².

A eq.(3.6) pode ser reescrita como:

$$m \ddot{Z} = mg - k_c \frac{I^2}{Z^2} \quad (3.7)$$

ou ainda:

$$f(Z, I) = \ddot{Z} = g - \frac{k_c I^2}{m Z^2} \quad (3.8)$$

A característica do foto sensor infravermelho é considerada linear na faixa de operação requerida, tal como mostra a equação (3.9), que relaciona a tensão de saída Y com a distância Z entre a esfera e o eletroímã.

$$Y = \gamma Z + \bar{Y} \quad (3.9)$$

onde Y é a tensão relacionada à distância Z ;

\bar{Y} é o valor da tensão no sensor quando a altura é 0m, e vale -7.472V;

γ é constante de conversão tensão-distância, e vale 328.1V/m.

A relação entre a tensão U entregue ao amplificador de corrente e o valor de corrente I fornecida à bobina é também linear, e está descrita na equação (3.10):

$$I = \rho U + \bar{I} \quad (3.10)$$

onde: I é a corrente fornecida à bobina relacionada à tensão U ;

\bar{I} é o valor da corrente na bobina do eletroímã quando a tensão enviada é 0V, e vale 0.514A;

ρ é constante de conversão corrente-tensão, e vale 0.166A/V.

A seguir serão discutidos alguns aspectos referentes às características de não linearidade e instabilidade do eletroímã do SLM da Feedback Instruments, ou seja serão apresentadas curvas que mostram como a força magnética, gerada pelo campo induzido no eletroímã, age sobre a esfera metálica, quando varia-se a posição da esfera e a corrente no eletroímã. As curvas foram obtidas utilizando-se o aplicativo Matlab e o modelo já apresentado ainda nesta seção.

A figura 3.3 mostra a curva característica da força aplicada à esfera, pela corrente aplicada ao eletroímã em três situações diferentes, isto é em “1” a esfera foi posicionada a uma distância $Z_1 = 0.01\text{m}$ do eletroímã, em “2” o novo valor da distância é $Z_2 = 0.02\text{m}$ e por fim em “3” a distância é $Z_3 = 0.03\text{m}$.

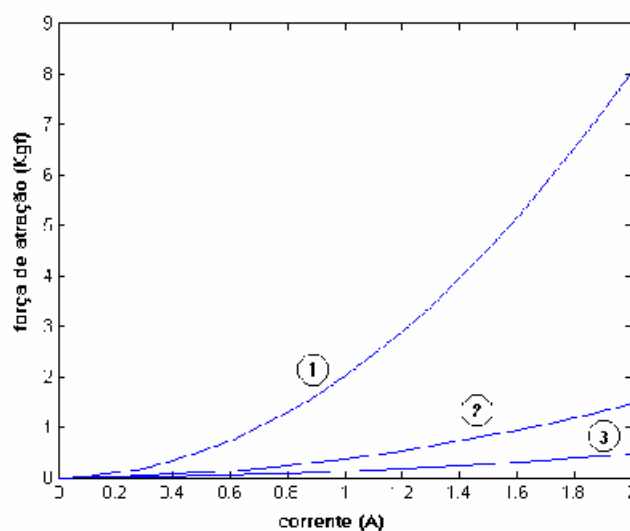


Figura 3.3: Gráfico que relaciona corrente no eletroímã pela força de atração exercida sobre a esfera.

Consegue-se, então, observar que para um mesmo valor de corrente, a esfera se expõe a diferentes valores de forças de atração quando situada em posições diferentes, ou seja esta força de atração é uma função não-linear proporcional ao inverso da distância, tal como já referenciado anteriormente.

É possível observar uma certa região bastante curva (não-linear) na qual o comportamento real da planta pode se afastar com maior facilidade do comportamento do modelo linearizado. Assim deve-se escolher um período de amostragem o menor possível, porém sem comprometer a capacidade operacional dos computadores utilizados.

A seguir é feita mais uma observação sobre o comportamento do SLM, ou seja foram considerados três valores de corrente constantes aplicada ao eletroímã, que são elas $I_1 = 0.43\text{A}$, $I_2 = 0.87\text{A}$ e $I_3 = 1.3\text{A}$, e para cada uma das situações variou-se a posição da esfera partindo de uma distância $Z = 0.01\text{m}$ até $Z = 0.03\text{m}$ do eletroímã. Verificou-se que, como já era esperado, quanto mais próxima a esfera encontra-se do eletroímã maior a força magnética exercida sobre ela, característica esta que determina o sistema como sendo instável, pois a taxa de variação da força com relação à distância é negativa. Apresenta-se aqui o desejo da lei de controle, que é o de inverter a variação negativa das curvas mostradas na figura 3.4, para um valor positivo.

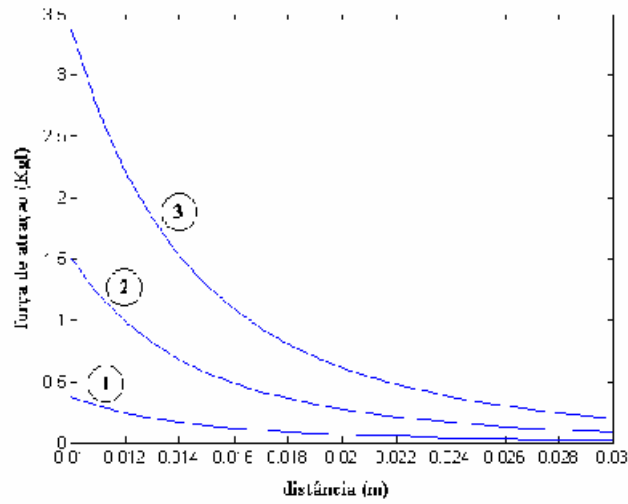


Figura 3.4: Gráfico que relaciona a distância com força de atração exercida sobre a esfera.

3.3 Linearização do modelo

O método de linearização utilizado será o de Taylor. O primeiro passo para se determinar o modelo linearizado aproximado de um sistema não-linear é o de obter um ponto de equilíbrio. Considere, então, uma corrente constante I_0 aplicada ao sistema, que resulta em uma altura constante Z_0 , de forma que a variação de ambas seja nula. Assim, segundo (3.6) o ponto de equilíbrio é:

$$0 = mg - k_c \frac{I_0^2}{Z_0^2} \quad \Rightarrow \quad Z_0 = I_0 \cdot \sqrt{\frac{k_c}{mg}} \quad (3.11)$$

Dado (3.8) que rege a dinâmica vertical da esfera, tem-se que suas derivadas parciais são:

$$\left. \frac{\partial f}{\partial Z} \right|_{Z=Z_0, I=I_0} = 2k_c \frac{I_0^2}{Z_0^3} = \frac{2mg}{Z_0} \quad (3.12)$$

$$\left. \frac{\partial f}{\partial I} \right|_{Z=Z_0, I=I_0} = -2k_c \frac{I_0}{Z_0^2} = -\frac{2\sqrt{mgk_c}}{Z_0} \quad (3.13)$$

Do método de Taylor, tem-se que o formato da função linearizada é:

$$F_L(\Delta Z, \Delta I) = f(Z_0, I_0) + \left. \frac{\partial f}{\partial Z} \right|_{Z=Z_0, I=I_0} \Delta Z + \left. \frac{\partial f}{\partial I} \right|_{Z=Z_0, I=I_0} \Delta I \quad (3.14)$$

onde: $f(Z_0, I_0) = 0$

$$\Delta Z = (Z - Z_0)$$

$$\Delta I = (I - I_0)$$

Assim o modelo linearizado completo, ou seja, aquele que envolve o amplificador de corrente, eletroímã e sensor ótico do sistema de levitação magnética é:

$$\ddot{y} = \frac{2g}{Z_0} \cdot y - \frac{2 \cdot \rho \cdot \gamma}{Z_0} \cdot \sqrt{\frac{gk_c}{m}} \cdot u \quad (3.15)$$

onde: $y = (Y - Y_0)$

Y é o sinal do sensor equivalente à posição da esfera

Y_0 é o valor do sinal do sensor no ponto de equilíbrio

$$u = (U - U_0)$$

U é o controle enviado ao SLM

U_0 é o valor de controle no ponto de equilíbrio

Considerando $x_1 = y$ e $x_2 = \dot{y}$, tem-se na forma matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ \frac{2 \cdot g}{Z_0} & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2 \cdot \rho \cdot \gamma}{Z_0} \cdot \sqrt{\frac{g \cdot k_c}{m}} \end{bmatrix} \cdot u = \mathbf{A}\mathbf{x} + \mathbf{B}u \quad (3.16)$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \mathbf{C}\mathbf{x}$$

ou ainda fazendo $\mathbf{C}(\mathbf{sI} - \mathbf{A})^{-1} \mathbf{B}$ se obtém a função de transferência de malha aberta do SLM:

$$G(s) = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \left[\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} - \begin{pmatrix} 0 & 1 \\ \frac{2g}{Z_0} & 0 \end{pmatrix} \right]^{-1} \cdot \begin{bmatrix} 0 \\ -\frac{2\rho\gamma}{Z_0} \sqrt{\frac{gk_c}{m}} \end{bmatrix} \quad (3.17)$$

$$G(s) = \frac{-\frac{2\rho\gamma}{Z_0} \sqrt{\frac{gk_c}{m}}}{s^2 - \frac{2g}{Z_0}} = \frac{-\Lambda}{s^2 - \eta} \quad (3.18)$$

$$\text{onde: } \Lambda = \frac{2\rho\gamma}{Z_0} \sqrt{\frac{gk_c}{m}} \quad (3.19)$$

$$\eta = \frac{2g}{Z_0} \quad (3.20)$$

3.4 Discretização e formulação do modelo em função da variação do controle

De forma a conseguir fazer com que um computador digital consiga encontrar a solução aproximada de equações diferenciais, torna-se necessária a utilização de algum método de discretização, que consiste em escrever equações diferenciais sob a forma de equações a diferenças. Uma forma simples de se realizar esta transformação é através do método de Euler. Portanto aplicando tal método para discretização do SLM, tem-se:

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t \\ \frac{2g}{Z_0} \Delta t & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2\rho\gamma\Delta t}{Z_0} \sqrt{\frac{k_c g}{m}} \end{bmatrix} \cdot u(k) \quad (3.21)$$

onde $x_1(k)$ é o valor da variável de estado x_1 , que se refere à posição da esfera no instante k ;

$x_2(k)$ é o valor da variável de estado x_2 , que se refere à velocidade da esfera no instante k ;

$u(k)$ é o valor do controle aplicado à planta no instante k ;

Δt é o período de amostragem utilizado na discretização do modelo.

É interessante lembrar que vários outros métodos de discretização poderiam ser utilizados, como ZOH (Zero Order Hold, Segurador de Ordem Zero) por exemplo, contudo optou-se pelo método de Euler por sua facilidade de implementação em tempo real, isto porque os elementos das matrizes **A** e **B** devem ser recalculados a cada nova iteração devido a sua característica não-linear. E, além disso, como o tempo de computação é algo valioso neste cenário, o método de Euler seria aparentemente o método de discretização mais “instantâneo” quando comparado a heurística necessária na implementação de um discretizador utilizando ZOH.

Devido ao fato da função objetivo da estratégia de controle preditivo aqui adotada penalizar a variação do controle ($\Delta u(k)=u(k)-u(k-1)$) e não o valor de controle ($u(k)$) propriamente dito, torna-se conveniente a escrita da dinâmica do sistema em função de Δu . Adicionalmente, esse procedimento insere um efeito integrativo no modelo de controle da planta [20]. Porém, para escrever o sistema de equações (3.21) em função de Δu , torna-se necessário aumentar a ordem do sistema inserindo-se mais um estado.

Então, seja $x_3(k+1) = u(k)$, que também pode ser escrito como $x_3(k) + \Delta u(k) = u(k)$.

Assim, tem-se que $x_3(k+1) = x_3(k) + \Delta u(k)$.

Na forma matricial:

$$\begin{aligned} \begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & \Delta t & 0 \\ \frac{2g\Delta t}{Z_0} & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 1 \end{bmatrix} \cdot x_3(k) + \\ &+ \begin{bmatrix} 0 \\ -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 1 \end{bmatrix} \cdot \Delta u(k) \end{aligned} \quad (3.22)$$

$$\begin{aligned}
\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} &= \begin{bmatrix} 1 & \Delta t & 0 \\ \frac{2g\Delta t}{Z_0} & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \\
&+ \begin{bmatrix} 0 \\ -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 1 \end{bmatrix} \cdot \Delta u(k)
\end{aligned} \tag{3.23}$$

$$\begin{bmatrix} x_1(k+1) \\ x_2(k+1) \\ x_3(k+1) \end{bmatrix} = \begin{bmatrix} 1 & \Delta t & 0 \\ \frac{2g\Delta t}{Z_0} & 1 & -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x_1(k) \\ x_2(k) \\ x_3(k) \end{bmatrix} + \begin{bmatrix} 0 \\ -\frac{2\rho\gamma\Delta t}{Z_0} \cdot \sqrt{gk_c/m} \\ 1 \end{bmatrix} \cdot \Delta u(k) \tag{3.24}$$

Para completar, deve-se determinar um período de amostragem (Δt) conveniente. No caso do SLM da Feedback Instruments considerando-se que a maior frequência de interesse é de 10Hz [17], adotou-se um valor 20 vezes maior para a frequência de amostragem [14], ou seja 200Hz ($\Delta t = 0.005s$).

3.5 Tratamento das Restrições

Nesta seção serão apresentadas as matrizes de restrição referente a amplitude e taxa de variação do controle, e amplitude de saída. A obtenção física de tais valores é mostrada a seguir no item 3.4.1, portanto o objetivo aqui é de simplesmente escrever as matrizes de restrição em seu formato adequado para utilização no controlador preditivo.

3.5.1 Determinação das restrições do SLM

Esta seção é dedicada à obtenção dos valores de restrição, referentes ao amplificador de corrente do SLM da Feedback Instruments. Assim serão apresentadas medidas, obtidas em laboratório, que correspondem primeiramente ao maior e menor valor de corrente que pode

ser fornecido por este dispositivo e conseqüentemente aos valores máximo e mínimo de tensão que não saturam o equipamento. Em seguida são apresentadas a maior taxa de variação de corrente na subida e a menor taxa de variação da corrente na descida.

→ Obtenção dos valores máximo e mínimo

Para encontrar o maior e menor valor de corrente que pode ser fornecido pelo dispositivo foram inseridos diversos valores de tensão na entrada do amplificador de corrente de forma a verificar seus respectivos valores de corrente na saída, como ilustra a figura 3.5.

Os valores de corrente foram obtidos através da leitura do valor de tensão sobre um resistor de 1Ω em série com a bobina do eletroímã.

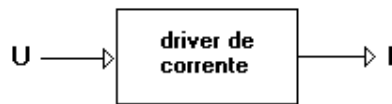


Figura 3.5: Diagrama de bloco representativo do amplificador de corrente

Inicialmente buscou-se inserir um valor de tensão que gerasse o menor valor de corrente, que no caso é de 0A, em seguida os valores de tensão são variados em aproximadamente 0.5V até encontrar o maior valor de corrente que pode ser fornecido pelo equipamento, através da saturação do mesmo, como mostra a tabela 3.1.

Tabela 3.1.: Relação entre tensão de entrada (U) e corrente de saída (I)

U (V)	I (A)
-2.83	0
-2.52	0.04
-2.01	0.13
-1.49	0.22
-1.02	0.3
-0.51	0.4
0	0.51
0.47	0.56
0.99	0.65
1.48	0.73
1.98	0.83
2.52	0.91
3.09	1.02
3.5	1.09
3.98	1.17
4.53	1.27
5	1.29

A seguir, na figura 3.6, é apresentada a curva da tensão de entrada do amplificador de corrente pela corrente de saída do mesmo. É possível verificar um comportamento aproximadamente linear para a faixa de operação contida entre -3V e $+5\text{V}$, ou ainda entre 0A e 1.3A .

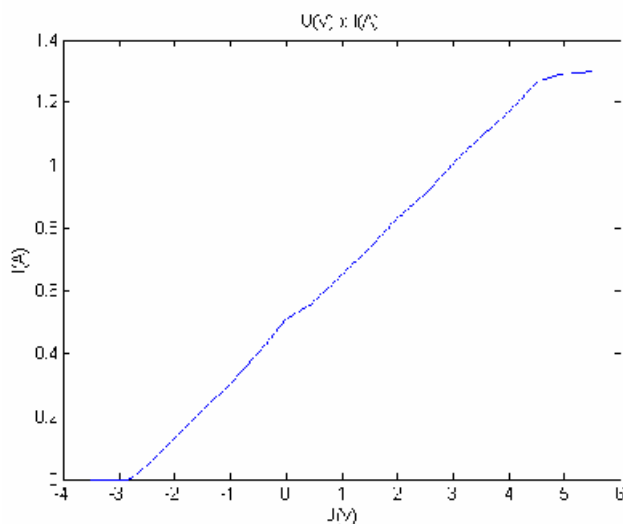


Figura 3.6: Curva representativa da relação U x I.

Observando a tabela 3.1 conclui-se que o maior valor de corrente que pode ser fornecido pelo amplificador é de 1.3 A, que corresponde a 5V. Por outro lado, o menor valor de corrente é de 0A, que corresponde a -3V. Além de obter os valores de máximo e mínimo do dispositivo, este ensaio foi de grande valia, pois mostra que a equação estática que representa a característica do dispositivo se aproxima bastante de seu comportamento real.

É interessante lembrar que inserir valores de tensão acima de +5V no amplificador de corrente não alteram o valor de corrente da saída do equipamento, pois este satura em 1.3A.

→ Obtenção das taxas de variação de aumento e decrescimento da corrente

A taxa de variação da corrente também foi obtida. O procedimento para determinação de tal restrição foi feito aplicando-se um degrau à entrada do amplificador de corrente, e com o auxílio de um osciloscópio foi observada a tensão sobre o resistor de 1Ω mencionado anteriormente. Este valor de tensão é numericamente igual ao valor da corrente na saída do dispositivo. O sinal observado em osciloscópio está representado na figura 3.7.

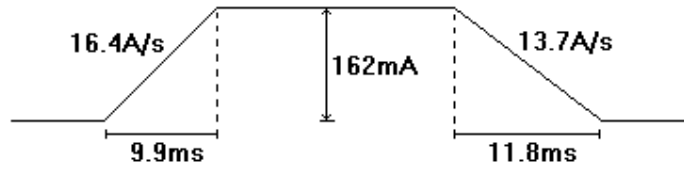


Figura 3.7: Gráfico que mostra as taxas de crescimento e decaimento do controle aplicado ao amplificador de corrente.

Expressando a restrição em termos do sinal de controle u aplicado ao amplificador de corrente, tem-se para $\rho = 0.166 A/V$ e $\Delta t = 0.005s$:

$$-\frac{13.7 A/s}{0.166 A/V} \leq \frac{\Delta u}{\Delta t} \leq \frac{16.4 A/s}{0.166 A/V} \quad (3.25)$$

$$-82.5 V/s \leq \frac{\Delta u}{\Delta t} \leq 98.8 V/s \quad (3.26)$$

É interessante destacar que as taxas de crescimento e decrescimento da corrente mantêm-se as mesmas para degraus de diferentes amplitudes.

3.5.2 Matrizes de Restrição

→ Amplitude de entrada

Substituindo os valores de maior e menor controle em (2.69), tem-se:

$$\mathbf{1} \cdot (-3V) \leq \begin{bmatrix} u(k) \\ u(k+1) \\ \vdots \\ u(k+N_u-1) \end{bmatrix} \leq \mathbf{1} \cdot (+5V) \quad (3.27)$$

Resultando, com uso da equação (2.78):

$$\begin{bmatrix} \mathbf{T} \\ -\mathbf{T} \end{bmatrix}_{2NxNu} \ddot{\mathbf{A}} \mathbf{u}_{Nu \times 1} \leq \begin{bmatrix} 5 - u(k-1) \\ -3 - u(k-1) \end{bmatrix}_{2Nx1} \quad (3.28)$$

→ Taxa de variação do controle

Da mesma forma, substituindo os valores de maior e menor taxa de variação do controle em (2.64), tem-se:

$$\mathbf{1} \cdot (-82.5V \cdot \Delta t / passo) \leq \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+Nu-1) \end{bmatrix} \leq \mathbf{1} \cdot (98.8V \cdot \Delta t / passo) \quad (3.29)$$

Utilizando $\Delta t = 0.005s$, resulta em:

$$\begin{bmatrix} \mathbf{I} \\ -\mathbf{I} \end{bmatrix}_{2Nu \times Nu} \ddot{\mathbf{u}}_{Nu \times 1} \leq \begin{bmatrix} 0.494 \\ -0.4125 \end{bmatrix}_{2Nu \times 1} \quad (3.30)$$

→ Amplitude de saída

O maior e menor valor de leitura do sensor ótico foram obtidos da literatura [17] e são respectivamente:

$$-2V \leq \hat{y} \leq +2V \quad (3.31)$$

que aplicados à equação (2.82), resultam em:

$$\begin{bmatrix} \mathbf{H} \\ -\mathbf{H} \end{bmatrix}_{2NxNu} \ddot{\mathbf{u}}_{Nu \times 1} \leq \begin{bmatrix} 2 - F \hat{x}(t) \\ -2 - F \hat{x}(t) \end{bmatrix}_{2Nx1} \quad (3.32)$$

→ Matriz de Restrições “**Re**”

Por fim reunindo todas as restrições em uma inequação matricial, tem-se:

$$\underbrace{\begin{bmatrix} T \\ -T \\ I \\ -I \\ H \\ -H \end{bmatrix}}_{\text{Re} \quad (4N+2Nu) \times Nu} \Delta u_{Nu \times 1} \leq \underbrace{\begin{bmatrix} 5-u(k-1) \\ -3-u(k-1) \\ 0.494 \\ -0.4215 \\ 2-\hat{F}x(k) \\ -2-\hat{F}x(k) \end{bmatrix}}_b \quad (4N+2Nu) \times 1 \quad (3.33)$$

3.6 Descrição do algoritmo LMPC

Esta seção é dedicada a descrever a sequência de tarefas a serem executadas pelo programa que implementa técnicas de controle preditivo no espaço de estados, de forma a se obter um correto funcionamento do sistema. Portanto será apresentado, a seguir, o fluxograma do algoritmo de controle LMPC.

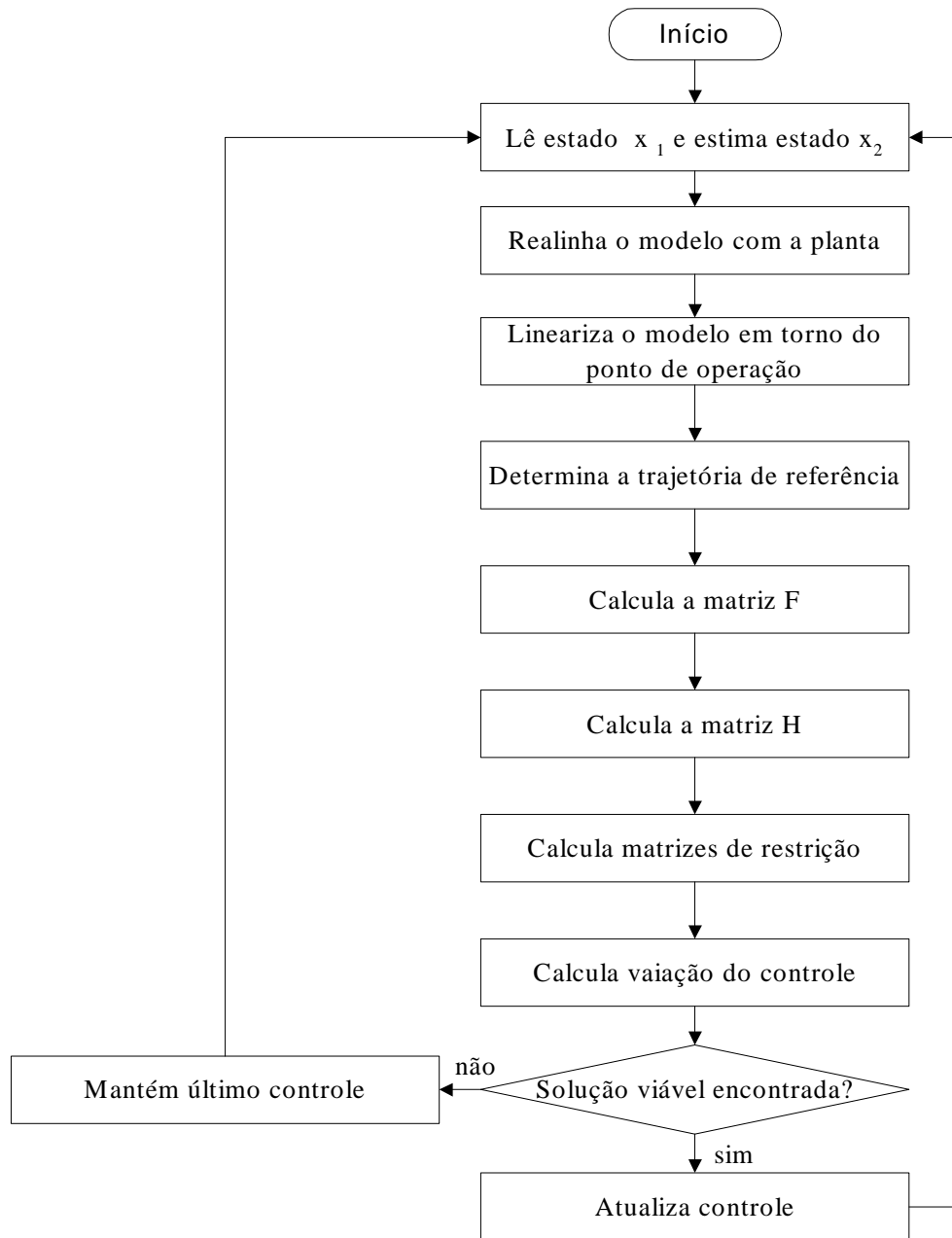


Figura 3.8: Fluxograma do algoritmo de LMPC.

- a) Início: Consiste em determinar a estrutura de dados e fornecer os valores iniciais às variáveis utilizadas;
- b) Lê estado x_1 e estima estado x_2 : Neste ponto a posição da esfera é lida junto ao sensor e em seguida o valor é atribuído ao estado x_1 . O estado x_2 , que se refere à velocidade

da esfera, é estimado efetuando-se a diferença entre as posições da esfera nos instantes atual e anterior dividido pelo período de amostragem Δt ;

- c) Realinha o modelo com a planta: Sendo o SLM instável em malha aberta, é necessário que a cada nova iteração, as condições iniciais do modelo sejam realinhadas com a atual condição do levitador (vide seção 2.6.2);
- d) Lineariza o modelo em torno do ponto atual: Adotou-se a heurística de realizar a linearização do modelo a cada passo de controle, com o intuito de melhor aproximar a dinâmica entre o modelo e o comportamento da planta;
- e) Determina a trajetória de referência: Gera a trajetória de referência através da equação (2.1);
- f) Calcula matriz **F**: Este cálculo é feito tal como já indicado no capítulo 3, equação (2.18);
- g) Calcula matriz **H**: Este cálculo é feito tal como já indicado no capítulo 3, equação (2.17);
- h) Calcula matrizes de restrição: Este cálculo é feito tal como já indicado na seção 3.4, equação (3.33);
- i) Calcula variação do controle: É realizada utilizando-se programação quadrática para o caso restrito, ou através da expressão (2.40) para o caso irrestrito;
- j) Solução viável encontrada?: Quando se utiliza programação quadrática nem sempre é possível determinar qual o valor ótimo para a nova variação do controle, devido a não existência de uma região factível causada, por exemplo, por uma restrição muito severa. Na ocorrência desta situação, o valor do controle é mantido constante;
- k) Retorna para leitura do sensor ótico.

3.7 Função Objetivo

O objetivo desta seção é o de ilustrar as variações que ocorrem na função objetivo do controlador MPC, quando a esfera encontra-se em posições diferentes solicitando assim valores de controle diferentes.

A função objetivo utilizada neste trabalho penaliza com peso δ o erro de rastreamento, e com peso λ a variação do controle. A seguir ilustra-se a função objetivo, considerando-se um horizonte de controle igual a 2, e um horizonte de predição igual a 50, e em duas situações diferentes. Na primeira a esfera está posicionada a 1.67cm do eletroímã, enquanto na segunda a esfera está a 2.89cm do mesmo. Em ambas as situações o desejo é de que a esfera vá para a posição $Y = 0V$, que é equivalente a uma distância de 2.3cm do eletroímã. Os valores dos parâmetros utilizados foram $\delta = 100$, $\lambda = 70$, $\alpha = 0,9$, e o período de amostragem $\Delta t = 0,005s$.

É possível observar que para a primeira situação (figura 3.9), pelo fato da esfera estar mais próxima ao eletroímã, a função objetivo possui menor custo na região de variações de controle negativa, ao contrário da segunda ocasião (figura 3.10), onde devido a esfera estar longe do eletroímã, valores de variação de controle positiva são quem minimizam o índice de desempenho.

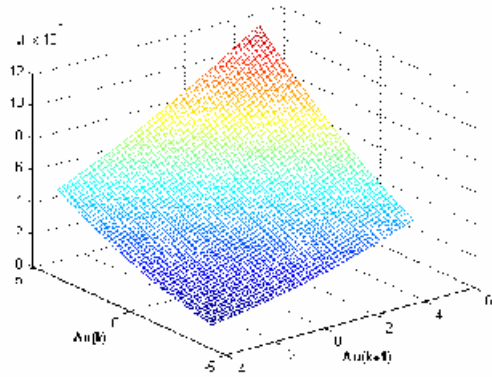


Figura 3.9: Função objetivo para esfera a 1.67cm do eletroímã.

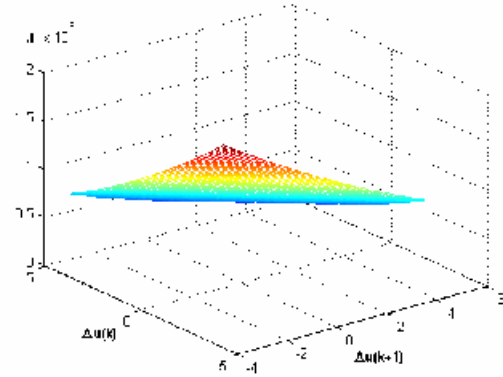


Figura 3.10: Função objetivo para esfera a 2.89cm do eletroímã.

Por fim a figura 3.11 ilustra a ocasião onde a esfera já se encontra sobre a posição desejada, ou seja em $Z = 2.3\text{cm}$ ou ainda $Y = 0\text{V}$.

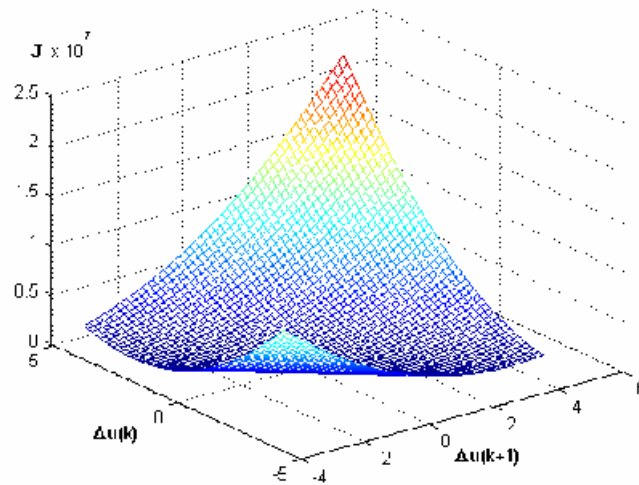


Figura 3.11: Função objetivo para esfera a 2.3cm do eletroímã.

Após observar todas as situações descritas acima é possível concluir que a ação do controlador se contrapõe a característica de instabilidade em malha aberta do SLM, observada na seção 3.2, pois para distâncias menores entre a esfera e o eletroímã, o valor de controle também é menor quando comparado a distâncias maiores.

Capítulo 4

RESULTADOS & DISCUSSÃO

4.1 Introdução

Este capítulo é dedicado a apresentar os resultados obtidos na fase de simulação e na fase experimental, além de discutir assuntos relacionados às dificuldades encontradas na implementação e ao desempenho obtido.

A lei de controle irrestrita é aplicada ao modelo do sistema, de forma a se visualizar a localização dos pólos, e assim poder analisar o comportamento da planta.

Por fim são apresentados os resultados em simulação, e na fase experimental são analisados os efeitos dos parâmetros de ajustes de um controlador LMPC. Foram também realizadas comparações entre diferentes técnicas de geração da lei de controle, com a finalidade de obter conclusões sobre o comportamento do LMPC.

4.2 Simulações

Nesta seção serão apresentados alguns resultados obtidos mediante simulação do modelo, em ambiente Simulink/Matlab. As simulações foram realizadas considerando-se dois métodos de controle: compensador de avanço de fase e controle preditivo. A simulação do mesmo sistema utilizando-se técnicas de geração da lei de controle diferentes foi feita a fim de ilustrar potenciais vantagens da estratégia de controle MPC.

4.2.1 Compensador de Avanço de Fase

O modelo do compensador em avanço utilizado aqui, é o mesmo do SLM da Feedback Instruments. Na simulação do sistema de controle em malha fechada utilizou-se o modelo não-linear do sistema de levitação magnética.

Inicialmente serão determinados os valores de alguns parâmetros a serem ajustados, tais como o valor de offset que leva a saída a 0V quando o sistema estiver sendo excitado por uma referência nula e qual valor de degrau da referência a ser inserido de forma que a esfera excursione por uma região pré-determinada.

→ Cálculo do offset

Considere o diagrama de blocos da figura 4.1:

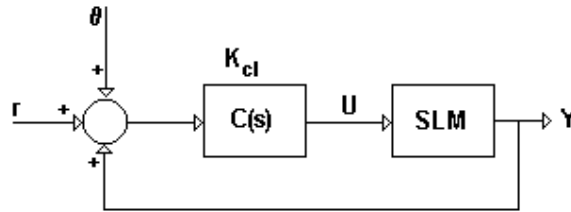


Figura 4.1: Diagrama de blocos ilustrativo.

Vale lembrar o objetivo que é determinar qual o valor de θ , que para um valor de referência $r = 0V$, leve a saída para $Y = 0V$. Para isso torna-se necessária a análise DC do sistema (compensador + planta) apresentado na figura acima.

Com relação ao compensador, tem-se que sua função de transferência é:

$$C_{lead}(s) = K_{cl} \cdot \frac{(0.0221s + 1)}{(0.0001s + 1)}; \quad (4.1)$$

Conclui-se então que sua influência em baixas frequências resume-se ao ganho K_{cl} .

Para analisar o SLM, basta a partir de (3.9), (3.10) e (3.11) escrever:

$$Y = \gamma[\rho U + \bar{I}] \sqrt{\frac{k_c}{mg}} + \bar{Y} \quad (4.2)$$

Como:

$$U = K_{cl}(Y + r + \theta) \quad (4.3)$$

Tem-se que:

$$Y = \gamma[\rho K_{cl}(Y + r + \theta) + \bar{I}] \sqrt{\frac{k_c}{mg}} + \bar{Y} \quad (4.4)$$

que resulta em:

$$\theta = 0.3496V \quad (4.5)$$

Para se obter o valor de θ acima ajustou-se o ganho $K_{cl} = 6.7$ pelo fato deste ser o valor central da faixa de ajuste do controlador da Feedback Instruments.

→ Obtenção da amplitude de referência

O procedimento é o mesmo do item anterior, ou seja, dadas duas variáveis deseja-se encontrar uma terceira. Para o caso aqui apresentado, as variáveis fixas serão $\theta = 0.3496V$, obtido anteriormente, e em primeira instância $Y = 0.2V$ para em seguida determinar qual r deve ser aplicado de forma a obter $Y = -0.2V$.

É interessante lembrar que o valor de k_c deve ser recalculado a cada novo valor desejado para Y .

Para $Y = 0.2V$

$$r = 0.172V \quad (4.6)$$

Para $Y = -0.2V$

$$r = -0.168V \quad (4.7)$$

→ Análise do Lugar das Raízes

De (3.19) e (3.20) obtém-se a FTMA, do modelo linearizado:

$$G(s) = -\frac{\Lambda}{s^2 - \eta} = -\frac{1171.6}{s^2 - 852.17} \quad (4.8)$$

Assim o modelo linearizado do sistema de levitação magnética apresenta os seguintes pólos de malha aberta: $s_1 = -29.19$; $s_2 = 29.19$.

Considere, agora, a função de transferência do compensador lead em conjunto com a planta dando origem a função de transferência de malha aberta:

$$C(s) \cdot G(s) = K_{cl} \cdot \frac{(0.0221s + 1)}{(0.0001s + 1)} \cdot \frac{-2\rho\gamma\sqrt{\frac{k_c g}{m}}}{Z_0 \cdot s^2 - 2g} = K_{cl} \cdot 221 \cdot \frac{(s + 45.25)}{(s + 10000)} \cdot \frac{-1171.6}{s^2 - 852.17} \quad (4.9)$$

$$FTMA_{COMP} = -K_{cl} \cdot \frac{(258920s + 11716000)}{(s^3 + 10000s^2 - 852.17s - 8521700)} \quad (4.10)$$

Os pólos de malha aberta do sistema compensado são então: $s_1 = -10000$; $s_2 = -29.19$; $s_3 = 29.19$.

Foi observado que valores de ganho menores do que 0.7 não são suficientes para estabilizar o sistema. Utilizando o mesmo ganho $K_{cl} = 6.7$, os pólos de malha fechada

posicionam-se em $s_1 = -9.82 \cdot 10^3$, $s_2 = -112$ e $s_3 = -63.5$. O fator de amortecimento é igual a 1, pois todos os pólos são reais e negativos.

A resposta do sistema em malha fechada a uma referência tipo onda quadrada, com $r = \pm 0.17V$ e $\theta = 0.35V$, pode ser observada na figura 4.2.

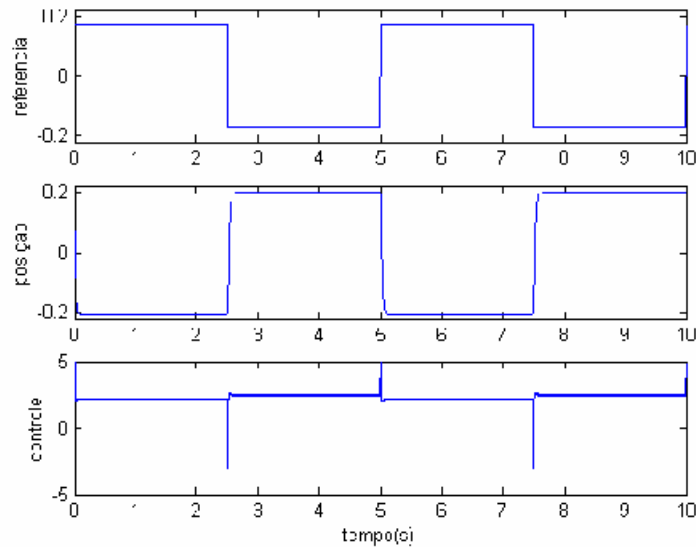


Figura 4.2: Formas de onda obtidas na simulação do SLM mais compensador lead.

O tempo de subida para uma excursão de 0 a 100% do sinal referente a posição da esfera da figura 4.2 é de 0.053s. É possível observar também a existência de uma inversão de fase entre os sinais de referência, e a posição da esfera, ou seja quando os valores da referência sobem, os valores da posição descem. Isto se deve ao fato da FTMA do SLM ser negada, obrigando o uso de realimentação positiva. Porém esta não é uma deficiência do sistema compensado em avanço, pois para sanar tal situação basta trabalhar com a referência negada ao invés da referência bruta. Por outro lado uma deficiência real do compensador lead é o fato deste não conseguir eliminar o erro em regime do levitador, que sob tais circunstâncias ficou em torno dos 15%.

4.2.2 Controlador Preditivo

Nesta seção são apresentados alguns resultados de simulação obtidos a partir da implementação de um controlador preditivo baseado em modelo. A planta utilizada na simulação é não-linear, porém o modelo interno ao controlador, utilizado para realização da predição, é linearizado.

O algoritmo implementado possui técnicas para rejeição a perturbações externas, correção de erro em regime permanente, tratamento de restrições de amplitude e variação da entrada e de amplitude da saída, e suavizador de referência. O problema de programação quadrática é solucionado utilizando-se a função *quadprog* do Matlab que utiliza o método do conjunto ativo similar ao descrito em [27].

Na estratégia de controle preditivo, quando adota-se, por exemplo, um horizonte de predição $N = 20$ e um período de discretização de 0.005s, o modelo permite utilizar predições em uma janela de 0.1s à frente da planta.

Todas as figuras desta seção 4.2.2 apresentarão o comportamento de pelo menos quatro variáveis, a trajetória comandada, a posição da esfera, o controle aplicado ao SLM e sua variação. Isso para tentar mostrar que nenhuma das três restrições (amplitude, taxa de variação da entrada e amplitude da saída) é violada. Como já dito anteriormente, a maior taxa de crescimento do controle é de 98.8V/s, portanto a cada passo Δt a maior taxa de variação do controle deve ser de aproximadamente 0.5V/passo de discretização.

O esquema em diagrama de blocos utilizado na simulação do SLM está apresentado na figura 4.3.

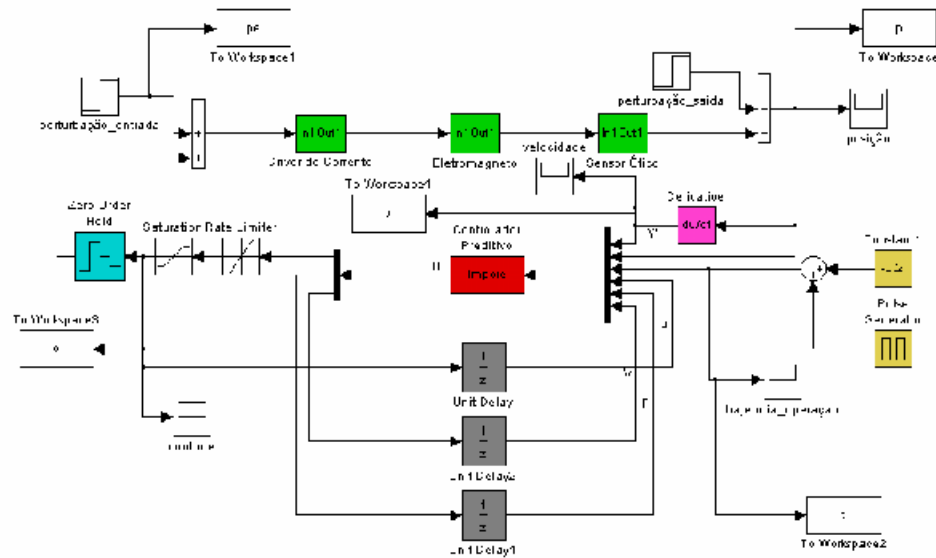


Figura 4.3: Diagrama de blocos utilizado na simulação do SLM.

Vale lembrar que o código LMPC utilizado nesta seção está escrito em linguagem Matlab, e que informações adicionais podem ser encontradas em [24].

→ Correção do erro em regime

Em ambas as simulações desta sub-seção os valores dos parâmetros utilizados são $N = 15$, $N_u = 15$, $\alpha = 0.7$, $\lambda = 10$ e $\delta = 100$.

O objetivo é ajustar a trajetória de referência em 0V e observar a posição final da esfera para dois casos distintos, em um primeiro instante sem a utilização do estado aumentado, e em seguida com sua utilização. Desta forma será possível observar o efeito do estado aumentado na correção do erro em regime do sistema.

A figura 4.4 ilustra a resposta obtida a partir de simulação sem utilização de técnicas de correção do erro em regime. O valor da saída, depois de estabilizada é de 0.895V.

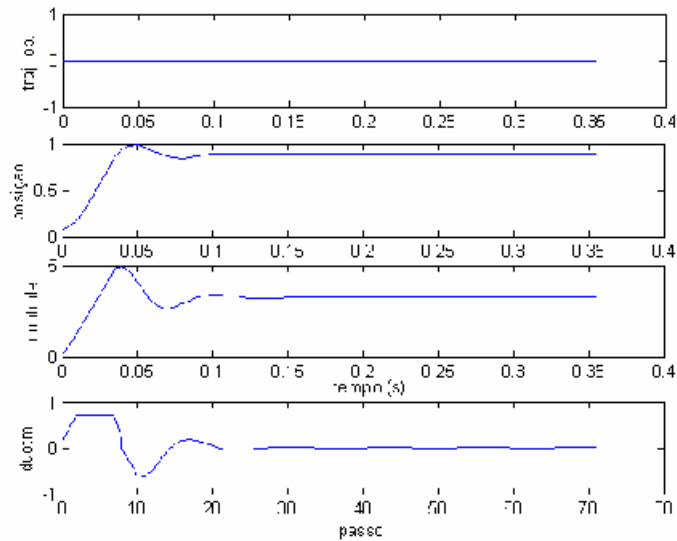


Figura 4.4: Forma de onda obtida sem a correção do erro em regime.

Já a figura 4.5 mostra o comportamento temporal do sistema utilizando o estado aumentado que realiza a correção do erro em regime eq.(2.54), e o valor da saída depois de estabilizada é de 0.005V, o que mostra que a correção do erro em regime ajuda, mas não resolve completamente o problema. O fato do erro em regime não ser corrigido em sua totalidade, deve-se aparentemente ao fato da utilização de um modelo realinhado.

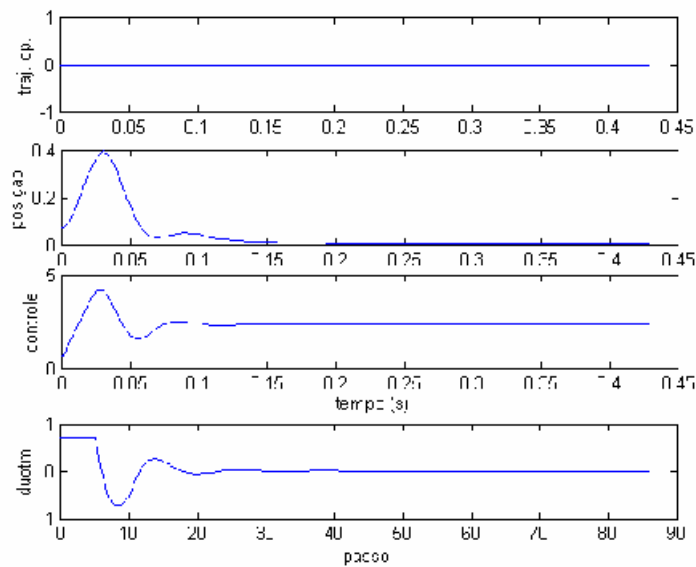


Figura 4.5: Forma de onda com correção do erro em regime.

→ Tempo de subida no rastreamento de uma trajetória

Nas figuras 4.6 e 4.7 são apresentadas as formas de onda referentes ao sistema de levitação magnética sendo comandado por um controlador preditivo baseado em modelo linear. Os valores dos parâmetros utilizados foram $\delta=10$, $\lambda=0.1$, $\alpha=0.7$, $\Delta t=0.005$, $N=15$ e $Nu=15$.

A figura 4.6 ilustra comportamento do sistema quando se utiliza a linearização do modelo em torno de um único ponto, no caso em torno do ponto $Z = 0.023\text{m}$ ou $Y = 0\text{V}$, apresentando um tempo de subida de 0.03s .

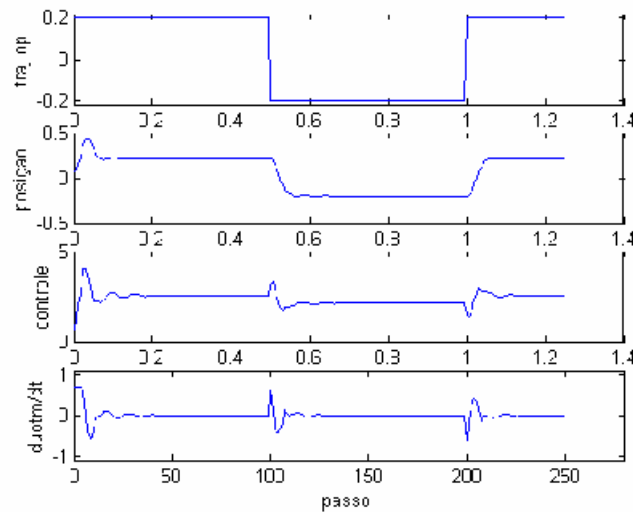


Figura 4.6: Formas de onda obtida a partir de linearização única.

A figura 4.7 ilustra o comportamento do sistema com a linearização sendo realizada a cada período de amostragem, ou seja, os elementos das matrizes **A** e **B** são recalculados a cada passo, através de uma heurística. Isto significa que o modelo interno ao controlador é linearizado a cada novo período de amostragem em torno do ponto atual de operação. Tal procedimento caracteriza o realinhamento do modelo, constituindo-se aqui a realimentação do sistema.

Comparando as figuras 4.6 e 4.7 é possível observar uma semelhança comportamental, de forma que torna-se difícil concluir sobre qual seria o melhor procedimento a ser adotado. Isso se deve ao fato de se ter trabalhado com uma pequena faixa em torno do ponto de operação escolhido. O tempo de subida aqui observado também é de 0.03s.

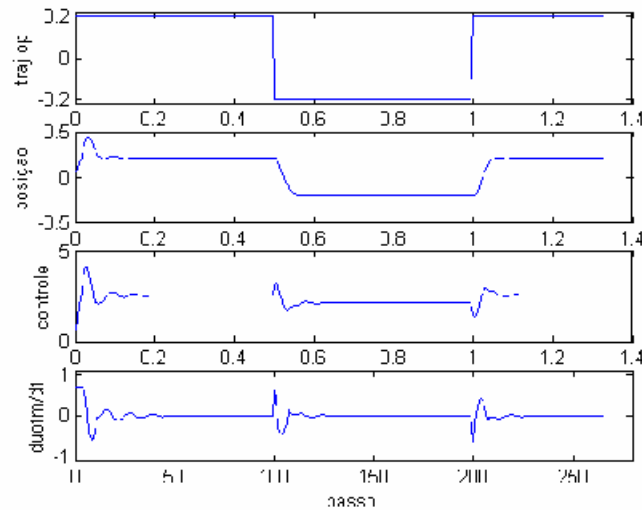


Figura 4.7: Formas de onda obtidas a partir da linearização passo a passo.

Das figuras 4.6 e 4.7 observa-se uma característica interessante e que difere da resposta obtida na simulação utilizando compensador lead, ou seja não existe mais a inversão de fase entre os sinais de referência e de posição da esfera, fato este devido ao controlador MPC minimizar um índice de desempenho em malha aberta, sendo constituída a realimentação somente através do realinhamento dos estados e saída entre a planta e o modelo. É possível observar, também, que as restrições são todas obedecidas, isto é não existe violação das restrições de amplitude de entrada, taxa de variação de entrada e amplitude de saída.

Vale ressaltar que todas as simulações e teste experimentais realizados neste trabalho utilizam a linearização em tempo real, com exceção da simulação da figura 4.6.

→ Efeito de α na trajetória de referência

Nesta seção será mostrado qual a influência do parâmetro α na trajetória de referência e por consequência na posição da esfera.

Os valores dos parâmetros fixos utilizados são $N = 15$, $Nu = 15$, $\delta = 10$, $\lambda = 0.1$ e $\Delta t = 0.005s$.

As figuras 4.8 a 4.12 ilustram a trajetória de referência para diferentes valores de α e mostram também o efeito das alterações deste mesmo parâmetro no posicionamento da saída.

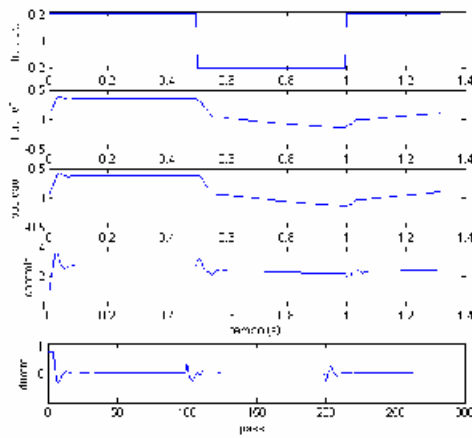


Figura 4.8: $\alpha = 0.9$

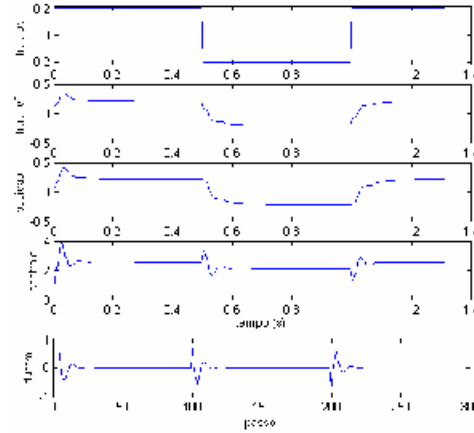


Figura 4.9: $\alpha = 0.7$

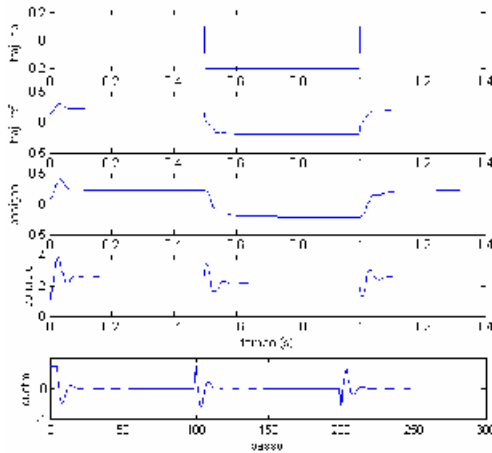


Figura 4.10: $\alpha = 0.5$

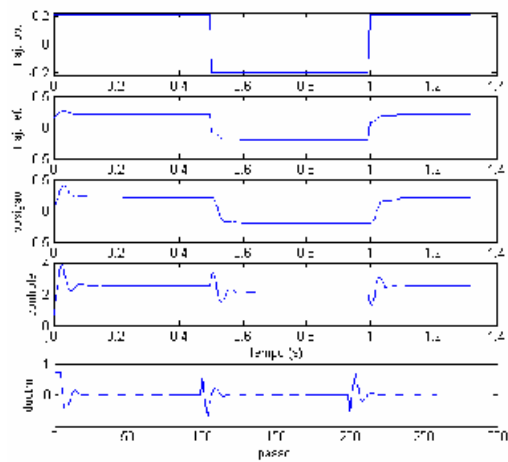


Figura 4.11: $\alpha = 0.3$

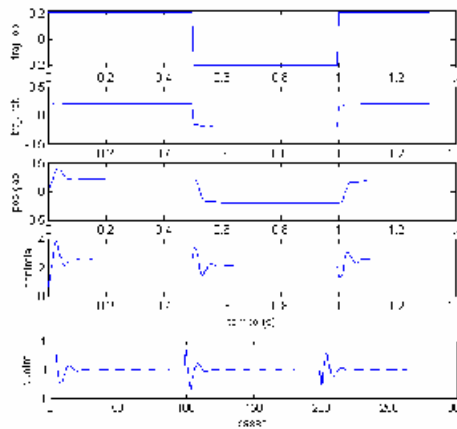


Figura 4.12: $\alpha = 0.1$

→ Análise dos pólos de malha fechada para uma lei de controle irrestrita

Antes de mostrar os resultados de simulação será realizada uma análise qualitativa referente aos efeitos dos parâmetros do controlador MPC sobre os pólos de malha fechada do sistema, desta maneira será possível justificar a escolha dos valores de tais parâmetros. Vale lembrar que existem trabalhos [2,32] que apresentam guias para o ajuste dos parâmetros MPC.

Para realizar a análise proposta considerou-se somente o caso irrestrito, pois aplicando (2.40) ao sistema:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{B}\ddot{\mathbf{u}} \\ \mathbf{y} &= \mathbf{Cx}\end{aligned}\tag{4.11}$$

obtem-se o sistema em malha fechada:

$$\dot{\mathbf{x}} = \mathbf{Ax} + \mathbf{B}(\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \delta \mathbf{H}^T (\mathbf{r} - \mathbf{Fx})\tag{4.12}$$

Considere:

$$\mathbf{M} = (\mathbf{H}^T \mathbf{H} + \lambda \mathbf{I})^{-1} \delta \mathbf{H}^T\tag{4.13}$$

Porém, como se deseja aplicar somente a primeira variação de controle:

$\mathbf{G} = \mathbf{M}(1,:)$, isto é, \mathbf{G} é vetor linha que contém somente a primeira linha de \mathbf{M} .

Então:

$$\begin{aligned}\dot{\mathbf{x}} &= \mathbf{Ax} + \mathbf{BG}(\mathbf{w} - \mathbf{Fx}) \\ \dot{\mathbf{x}} &= (\mathbf{A} - \mathbf{BGF})\mathbf{x} + \mathbf{BGw}\end{aligned}\tag{4.14}$$

Portanto, a análise dos autovalores de $(\mathbf{A} - \mathbf{BGF})$, resultará na sensibilidade referente a critérios de desempenho tais como amortecimento e frequência natural do sistema, por exemplo.

→ Peso do erro de predição (δ) e peso da variação do controle (λ)

Com os parâmetros $N = 30$ e $Nu = 30$ fixados, foi alterada a relação entre δ e λ , e observou-se que razões maiores diminuem a parte real dos pólos, e portanto o controlador torna-se mais “veloz”, e aumentam a parte imaginária dos pólos complexos com diminuição do fator de amortecimento. Os valores de ξ e ω_n mostrados na tabela 4.1, foram obtidos considerando os pólos complexos como sendo dominantes.

Em simulação foi verificado que o aumento da relação δ / λ realiza também um certo deslocamento no valor de saída do sistema, por exemplo para uma relação de 10 o erro em regime era nulo, porém para uma relação de 100, o sistema além de se tornar mais oscilatório, tal como já citado acima, também teve o valor de saída alterado para $-0.55V$.

Tabela 4.1: Localização dos pólos ao variar a razão δ/λ .

Parâmetro	Pólos	ξ e ω_n
$\lambda = 1$ $\delta = 1$	0.8202+0.2071i 0.8202-0.2071i 0.718	$\xi = 0.56$ $\omega_n = 59.72\text{rad/s}$
$\lambda = 1$ $\delta = 2$	0.7977+0.2314i 0.7977-0.2314i 0.6934	$\xi = 0.55$ $\omega_n = 67.58\text{rad/s}$
$\lambda = 1$ $\delta = 3$	0.7831+0.246i 0.7831-0.246i 0.678	$\xi = 0.54$ $\omega_n = 72.56\text{rad/s}$
$\lambda = 1$ $\delta = 4$	0.7719+0.256i 0.7719-0.256i 0.6666	$\xi = 0.54$ $\omega_n = 76.23\text{rad/s}$
$\lambda = 1$ $\delta = 5$	0.7628+0.2651i 0.7628-0.2651i 0.6574	$\xi = 0.538$ $\omega_n = 79.4\text{rad/s}$
$\lambda = 1$ $\delta = 10$	0.7318+0.2914i 0.7318-0.2914i 0.6276	$\xi = 0.533$ $\omega_n = 89.6\text{rad/s}$
$\lambda = 0.1$ $\delta = 10$	0.594+0.3779i 0.594-0.3779i 0.5142	$\xi = 0.526$ $\omega_n = 133.3\text{rad/s}$

→ Horizonte de Predição (N)

Com os parâmetros $Nu = 1$, $\delta = 2$ e $\lambda = 1$, observa-se que um horizonte de predição igual a 1 não é suficiente para trazer todos os pólos de malha fechada para dentro do raio unitário, e conseqüentemente estabilizar o sistema. Já valores de horizonte de predição grandes tornam o sistema superamortecidos porém com frequência natural não amortecida reduzida.

Tabela 4.2: Localização dos pólos ao variar o horizonte de predição N .

Parâmetro	Pólos	ξ e ω_n
$N = 1$	1.145 0.854 1	Instável por possuir pólo fora do círculo unitário.
$N = 10$	0.8507+0.098i 0.8507-0.0980i 0.1241	$\xi = 0.804$ $\omega_n = 38.58\text{rad/s}$
$N = 15$	0.8934+0.0213i 0.8934-0.0213i 0.0127	$\xi = 0.978$ $\omega_n = 22.98\text{rad/s}$
$N = 20$	0.962 0.8589 0.002	$\xi = 1$

→ Horizonte de Controle (Nu)

Para $N = 30$, $\delta = 2$ e $\lambda = 1$, observa-se que o aumento do horizonte de controle aumenta a frequência natural do sistema, além de torná-lo criticamente amortecido considerando que os pólos dominantes sejam os pólos complexos.

Tabela 4.3: Localização dos pólos ao variar o horizonte de controle Nu .

Parâmetro	Pólos	ξ e ω_n
$Nu = 1$	0.991 0.8543 0.0001	$\xi = 1$
$Nu = 10$	0.7931+0.2253i 0.7931-0.2253i 0.6949	$\xi = 0.572$ $\omega_n = 67.49\text{rad/s}$
$Nu = 15$	0.7964+0.232i 0.7964-0.232i 0.6933	$\xi = 0.55$ $\omega_n = 67.91\text{rad/s}$
$Nu = 20$	0.7976+0.2313i 0.7976-0.2313i 0.6934	$\xi = 0.55$ $\omega_n = 67.58\text{rad/s}$
$Nu = 30$	0.7977+0.2314i 0.7977-0.2314i 0.6934	$\xi = 0.55$ $\omega_n = 67.58\text{rad/s}$

Observando a tabela 4.3 conclui-se que não há alteração significativa na localização dos pólos ao se alterar Nu .

→ Análise dos Horizontes de Controle e Predição no rastreamento de uma trajetória

Analisando as figuras 4.13, 4.14, 4.15 e 4.16 é possível observar que inicialmente as variações de controle são conservadoras, tornando-se mais agressivas com o aumento do horizonte de predição. Além disso, considerando como pólos dominantes os pólos complexos conjugados, pode-se observar um aumento do fator de amortecimento e da frequência natural não-amortecida do sistema, ao se aumentar o horizonte de predição, tal como descrito nas figuras a seguir. O sinal referente a posição da figura 4.16 apresentou um pequeno erro em regime em torno de 10%. Em todas as simulações os valores dos parâmetros fixos utilizados são $\alpha = 0.7$, $\lambda = 10$ e $\delta = 100$.

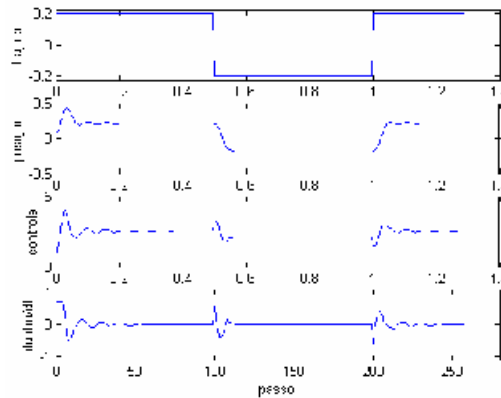


Figura 4.13: $N=7$ e $Nu=5$
 pólos em z : 0.6211 e $0.7625 \pm 0.2958i$
 $\omega_n = 84.23 \text{ rad/s}$
 $\xi = 0.4774$

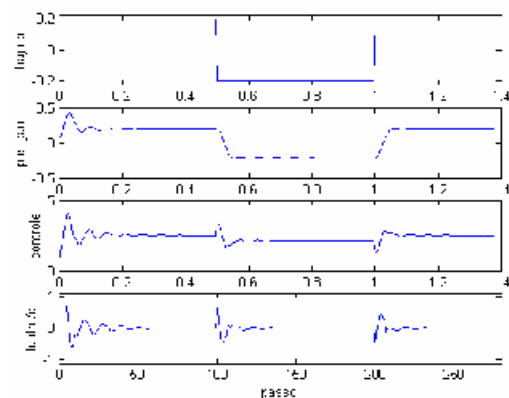


Figura 4.14: $N=10$ e $Nu=5$
 pólos em z : 0.6254 e $0.7466 \pm 0.3012i$
 $\omega_n = 88.1 \text{ rad/s}$
 $\xi = 0.4922$

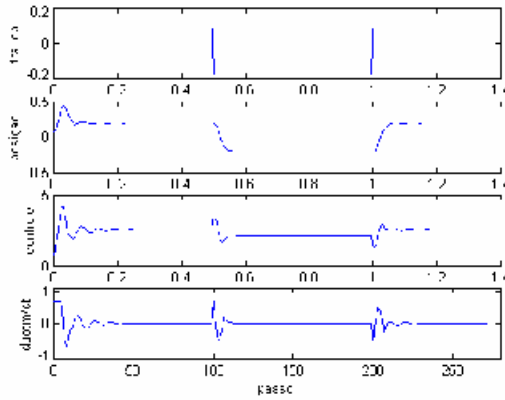


Figura 4.15: N=15 e Nu=5

pólos em z : 0.6472 e $0.683 \pm 0.2815i$
 $\omega_n = 98.9 \text{ rad/s}$
 $\xi = 0.6123$

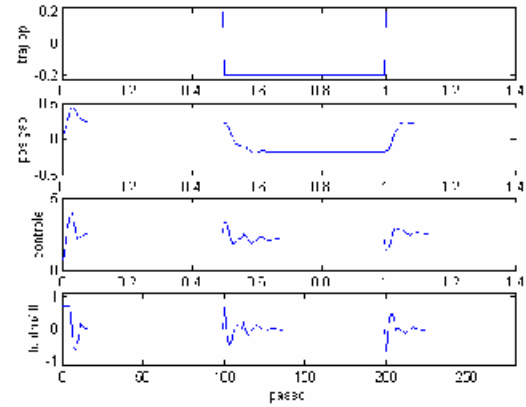


Figura 4.16: N=20 e Nu=5

pólos em z : 0.6706 e $0.6546 \pm 0.2534i$
 $\omega_n = 102.31 \text{ rad/s}$
 $\xi = 0.692$

O aumento do horizonte de controle aumenta o ganho em baixas frequências, pois foi capaz de corrigir o pequeno erro de regime apresentado sobre a situação da figura 4.16. Sobre a localização dos pólos, estes parecem não se alterar de forma significativa, tal como se pode observar nas figuras 4.17, 4.18, 4.19 e 4.20.

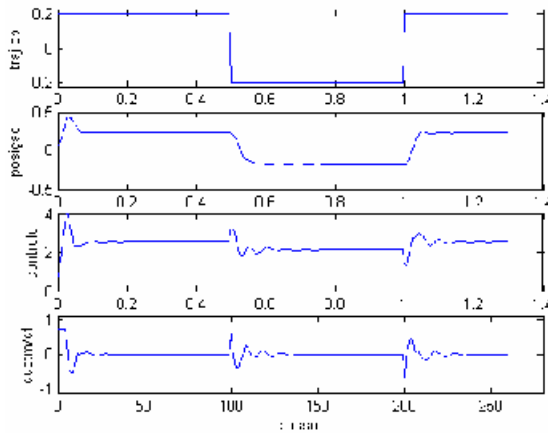


Figura 4.17: N=20 e Nu=7

pólos em z : 0.6326 e $0.7206 \pm 0.2758i$
 $\omega_n = 89.63 \text{ rad/s}$
 $\xi = 0.58$

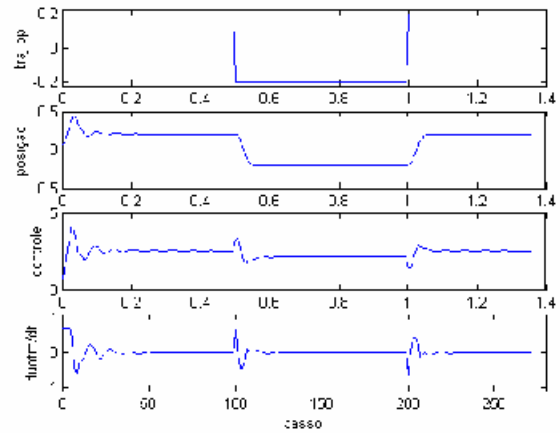


Figura 4.18: N=20 e Nu=10

pólos em z : 0.6276 e $0.7314 \pm 0.2919i$
 $\omega_n = 89.81 \text{ rad/s}$
 $\xi = 0.53$

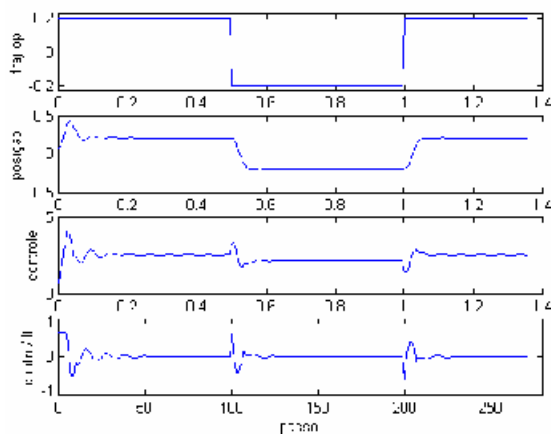


Figura 4.19: N=20 e Nu=15
 pólos em z : 0.6276 e $0.7317 \pm 0.2915i$
 $\omega_n = 89.6 \text{ rad/s}$
 $\xi = 0.53$

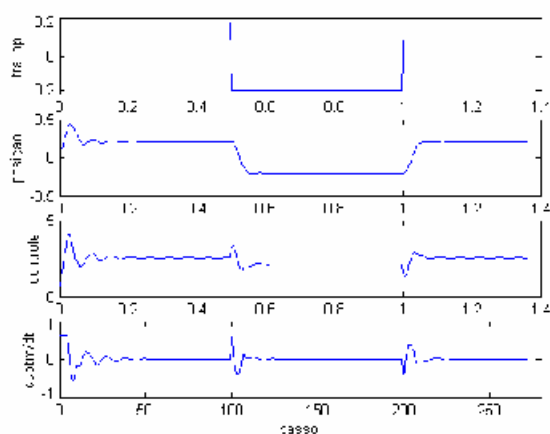


Figura 4.20: N=20 e Nu=20
 pólos em z : 0.6276 e $0.7318 \pm 0.2915i$
 $\omega_n = 89.6 \text{ rad/s}$
 $\xi = 0.53$

→ Análise de rejeição a perturbações

A seguir será apresentado o comportamento do sistema quando sujeito a uma perturbação externa. Primeiramente não haverá utilização de técnicas de rejeição a perturbações externas, e em seguida sim. Em ambos os casos, inicialmente a perturbação será posta junto à saída representando uma perturbação junto ao sensor de leitura da posição da esfera, e em seguida a ocasião onde uma perturbação é inserida junto à entrada do sistema.

Na figura 4.21 a perturbação inserida é do tipo degrau e de valor $0.5V$, que ocasionou um erro de $-0.075V$, resultando em uma rejeição de 85% do valor da mesma. Na figura 4.22 foi utilizado o mesmo sinal de perturbação, porém atuando na entrada do SLM, o qual gerou um erro de $-0.09V$, uma rejeição de 82% do valor da perturbação.

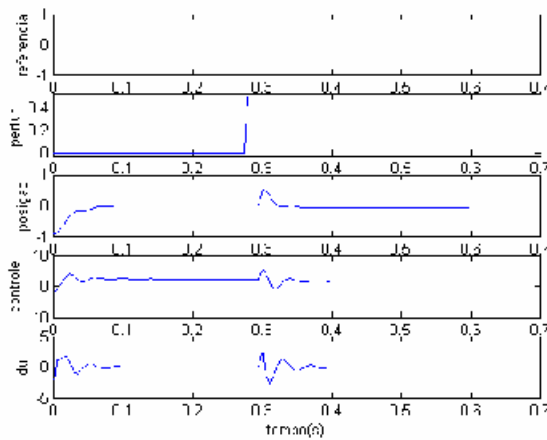


Figura 4.21: Sistema sem técnica de rejeição a perturbação na saída.

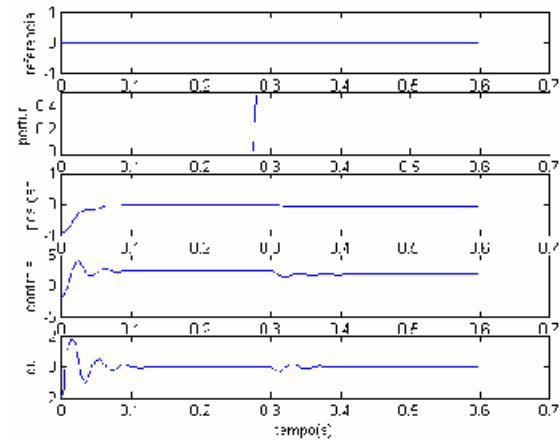


Figura 4.22: Sistema sem técnica de rejeição a perturbação na entrada.

Mais uma vez, na figura 4.23, a perturbação inserida é do tipo degrau e de valor 0.5V, que ocasionou um erro de $-0.021V$, resultando em uma rejeição de 96% do valor da mesma. Na figura 4.24 o mesmo sinal de perturbação atuando na entrada do SLM, gerou o mesmo erro de $-0.021V$, uma rejeição de 96% do valor da perturbação. Portanto a inserção da técnica de rejeição a perturbações externas melhora em torno de 10% o desempenho do sistema quando refere-se a rejeição de perturbações.

Na implementação da rejeição a perturbação foi observada a necessidade de se ponderar o erro entre saída medida e valor predito através de um fator multiplicativo.

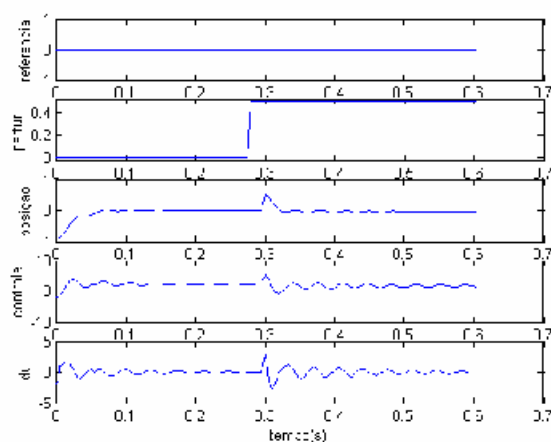


Figura 4.23: Sistema sujeito a perturbação na saída com técnica de rejeição a perturbação.

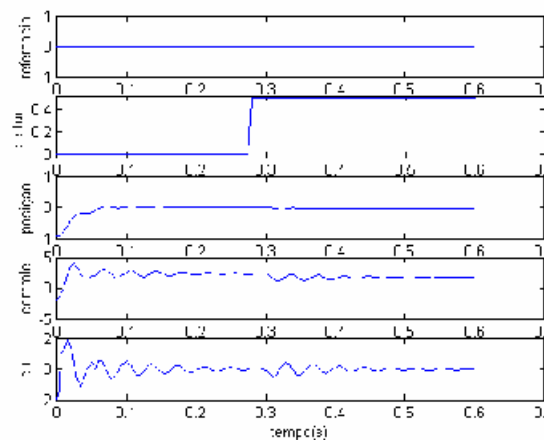


Figura 4.24: Sistema sujeito a perturbação na entrada com técnica de rejeição a perturbação.

Antes de encerrar a seção Simulação, será apresentado um conjunto de informações as quais ilustram o tempo médio de execução das instruções ao se alterar o tamanho dos horizontes de predição e de controle para os casos onde a lei de controle é obtida a partir de condições irrestrita e restrita.

→ Efeito dos horizontes de controle e predição no tempo médio de execução de tarefas.

Os valores apresentados a seguir foram obtidos mediante simulação em ambiente Matlab, e utilizando linguagem Matlab, portanto não se referem ao tempo gasto em cada iteração para o caso experimental, onde o programa está escrito em C/C++, fato que torna o tempo médio de execução bem menor.

Tabela 4.4: Tempo médio de execução para caso irrestrito.

Horizonte de Predição (N)	Horizonte de Controle (Nu)	Tempo Médio de Execução em segundos
1	1	0.0018
10	1	0.0064
20	1	0.01
30	1	0.0156
10	10	0.015
20	10	0.04
30	10	0.067
20	20	0.053
30	20	0.1
30	30	0.1145

Ao observar a tabela 4.5 conclui-se que o tamanho do horizonte de predição exerce um maior peso relacionado ao custo computacional, pois para a ocasião $N = 30$ e $Nu = 10$, o tempo médio de execução foi superior ao tempo médio obtido com $N = 20$ e $Nu = 20$, porém ambos geram uma determinada matriz **Re** com o mesmo número de linhas de restrição, que é igual a $2*N + 2* Nu$.

Tabela 4.5: Tempo médio de execução utilizando programação quadrática

Horizonte de Predição (N)	Horizonte de Controle (Nu)	Tempo Médio de Execução em segundos
1	1	0.35
10	1	0.36
20	1	0.36
30	1	0.37
10	10	0.38
20	10	0.39
30	10	0.425
20	20	0.41
30	20	0.465
30	30	0.485

Ao utilizar programação quadrática para obtenção dos valores de controle observa-se um tempo médio de execução mais homogêneo quando comparado ao caso irrestrito, porém o custo computacional aumenta consideravelmente, chegando a ser 194 vezes maior para $N = 1$ e $Nu = 1$, e 4 vezes maior para $N = 30$ e $Nu = 30$. Contudo, foi observado através de simulações, que bons valores para o horizonte de predição e de controle são aqueles entre 15 e 30, o que é suficiente para estabilizar o sistema sem sobrecarregar o processamento.

4.3 Avaliação experimental utilizando simulação analógica

Antes da experimentação junto ao sistema de levitação magnética da Feedback Instruments, foram realizados testes em sistemas analógicos construídos com a utilização de amplificadores operacionais [39].

A seguir serão determinados os valores dos componentes a serem utilizados, e dar-se-á início a fase de implementação do circuito em *proto-board*, e em seguida a malha será fechada aplicando-se o sinal de controle através da placa PCL-711B [22] que está instalada no computador escravo da topologia mestre-escravo do xPC Target do Matlab. Os computadores mestre (Athlon 900MHz com 256Mb de RAM) e escravo (K6-II 350MHz com 128Mb de RAM) se comunicam por um cabo serial [9] conectado à interface RS-232. Cabe destacar que o algoritmo utilizado nesta fase do trabalho está escrito em linguagem C++, portanto trabalha-se com orientação a objetos. O motivo pelo qual utiliza-se C++ e não C é devido a utilização de uma classe chamada DMatrix, a qual nos auxilia muito nas operações matriciais [3].

Vale lembrar que a partir deste ponto a lei de controle aplicada ao sistema é obtida a partir dos conceitos de controle preditivo, porém sem a utilização do tratamento de restrições. Assim, as variações de controle não são obtidas através da solução de um problema de programação quadrática. O motivo deve-se ao fato de não ter sido possível a utilização em uma mesma S-Function, da classe DMatrix escrita em C++, em conjunto com o algoritmo de programação quadrática escrito em C.

O algoritmo aqui utilizado foi inicialmente escrito e testado no aplicativo Microsoft Visual Studio 6.0, para só então transcrevê-lo para o formato S-Function do Matlab.

Foi observado que a utilização de uma linguagem orientada a objetos, no caso C++, possui a vantagem de simplificar a escrita de algoritmos, porém necessita de uma capacidade computacional bastante maior, quando comparada com a linguagem C.

Considere a função de transferência aproximada:

$$G(s) = \frac{1000}{s^2 - 900} = \frac{Y(s)}{U(s)} \quad (4.15)$$

que pode ser reescrita na forma da EDO:

$$\ddot{y} - 900y = 1000u \quad (4.16)$$

ou ainda, escolhendo-se:

$$\begin{aligned} x_1 &= y \\ x_2 &= \dot{y} \end{aligned} \quad (4.17)$$

tem-se, já na forma matricial:

$$\begin{bmatrix} \dot{x}_1 \\ \dot{x}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 900 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1000 \end{bmatrix} u \quad (4.18)$$

Uma das maneiras de se representar este conjunto de equações esquematicamente está ilustrado na figura 4.25 a seguir:

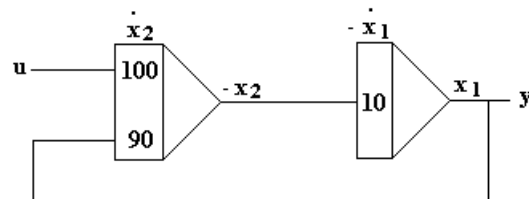


Figura 4.25: Representação do modelo sob a forma de blocos integrativos.

cujo circuito elétrico equivalente pode ser visto na figura 4.26:

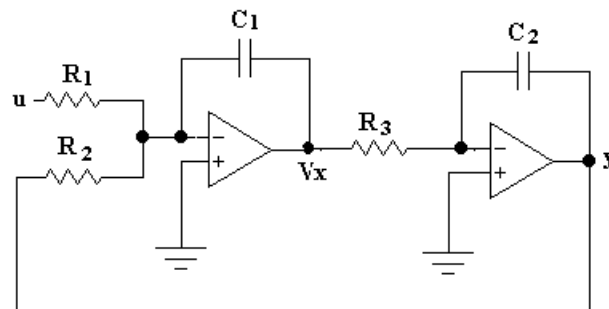


Figura 4.26: Diagrama elétrico equivalente ao modelo linearizado do SLM.

onde:

$$\frac{1}{R_1 C_1} = 100; \quad \frac{1}{R_2 C_1} = 90; \quad \frac{1}{R_3 C_2} = 10$$

Assim escolheu-se:

$$R_1 = 10\text{k}\Omega \pm 5\%; \quad (4.19)$$

$$R_2 = 11\text{k}\Omega \pm 5\%; \quad (4.20)$$

$$R_3 = 1\text{k}\Omega \pm 5\%; \quad (4.21)$$

$$C_1 = 1\mu\text{F}; \quad (4.22)$$

$$C_2 = 100\mu\text{F}. \quad (4.23)$$

Cabe observar a necessidade de se inverter o sinal de saída deste circuito, pois a função de transferência original possui o sinal trocado.

O diagrama de blocos utilizado nesta fase está ilustrado a seguir na figura 4.27.

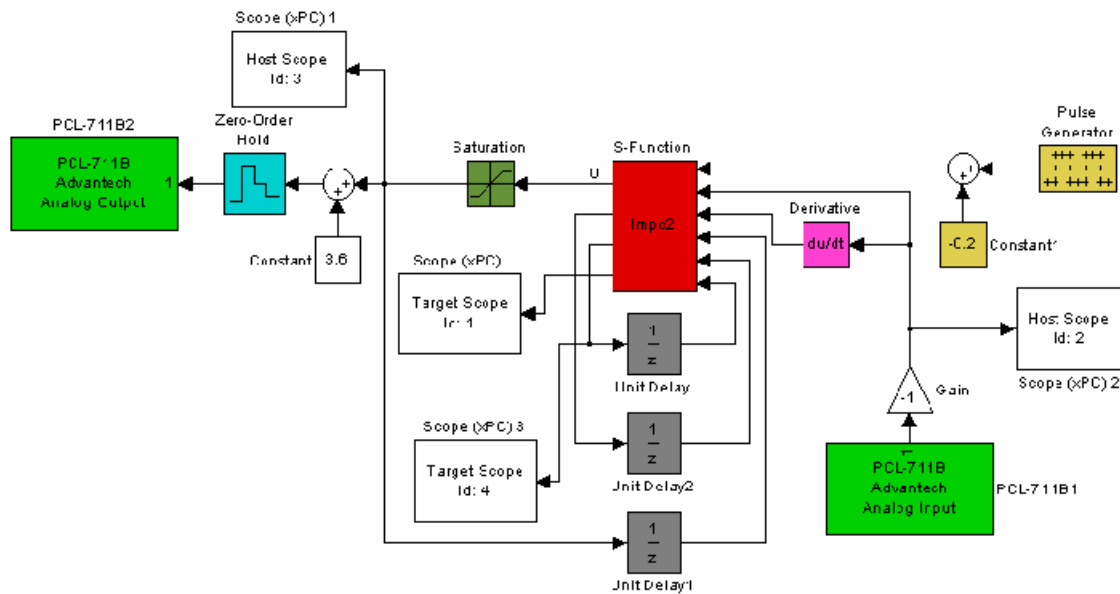


Figura 4.27: Diagrama de blocos utilizado na experimentação.

Nesta fase de experimentação foi observado que a utilização de técnicas para correção do erro em regime não era necessária. Tal constatação se explica pelo fato da resposta dinâmica do modelo utilizado pelo controlador MPC e da planta construída a partir de amplificadores operacionais serem bastante próximas.

Os valores dos parâmetros utilizados foram $\delta=100$, $\lambda=95$, $\alpha=0.9$, $\Delta t=0.001s$, e os resultados obtidos podem ser visualizados a seguir na figura 4.28. Vale destacar que o período de amostragem utilizado nesta seção foi diminuído devido a impossibilidade em se estabilizar o circuito composto com AO, que simulam a dinâmica do SLM. Aparentemente tal necessidade deve-se ao fato do método de discretização adotado ter mostrado ser bastante sensível ao período de amostragem.

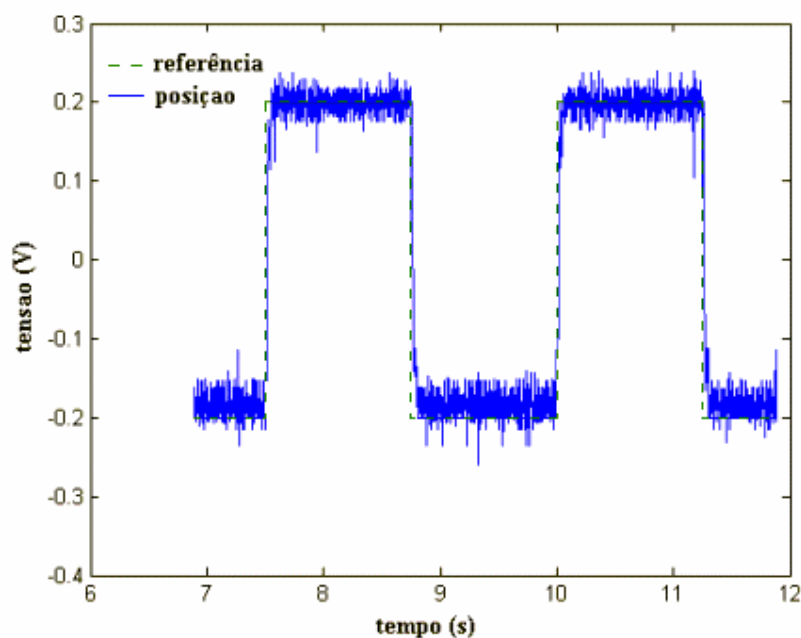


Figura 4.28: Formas de onda obtida a partir da simulação.

A resposta observada na figura 4.28 não apresenta sobresinal, porém apesar de apresentar um pequeno erro de regime, segue a trajetória comandada de forma fiel, podendo-

se afirmar, a partir deste ponto, que a tecnologia xPC Target do Matlab e o circuito condicionador de sinais estão operando de forma adequada.

4.4 Experimentos utilizando o SLM

Novamente surgiu a necessidade em se diminuir o valor do período de amostragem, pois o período de 0.001s, utilizado na seção 4.3, não era suficientemente pequeno para estabilizar a esfera no SLM. Optou-se, então, por um valor de Δt igual a 0.0005s. Os blocos devem ser executados sobre o modo “single-tasking”, que pode ser selecionado através do menu Simulation->Simulation Parameters->Solver, na janela do modelo do Simulink. Outra observação importante a ser destacada é que horizontes de predição maiores que 30 geravam sobrecarga computacional, isto é o tempo médio de execução de tarefas era maior do que o período de amostragem (Δt).

4.4.1 Correção do erro em regime

Nesta seção serão apresentados resultados referente à correção do erro em regime. Como já observado em capítulos anteriores, esta correção pode ser feita aumentando-se em um a ordem das equações de estados, sendo que este estado adicional modela uma perturbação constante sobre a saída do sistema.

É interessante lembrar que na seção onde se realizaram experimentos utilizando AO, técnicas de correção de erro em regime não foram necessárias, pois os comportamentos do modelo e da planta mostraram-se bastante similares.

Os valores dos parâmetros utilizados neste item foram $\delta=100$, $\lambda=70$, $\alpha=0.9$, $N=28$, e $Nu=21$.

As figuras 4.29 e 4.30 apresentam a saída do sistema sem e com correção do erro em regime, respectivamente. Como se pode ver, um erro de regime relativamente grande, apresentado na figura 4.29, foi sensivelmente reduzido, tal como ilustrado na figura 4.30.

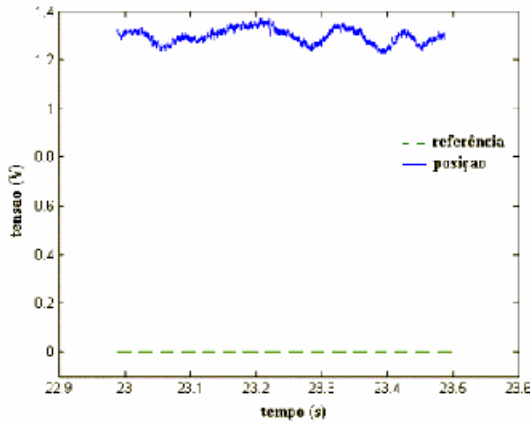


Figura 4.29: Forma de onda sem correção de erro em regime

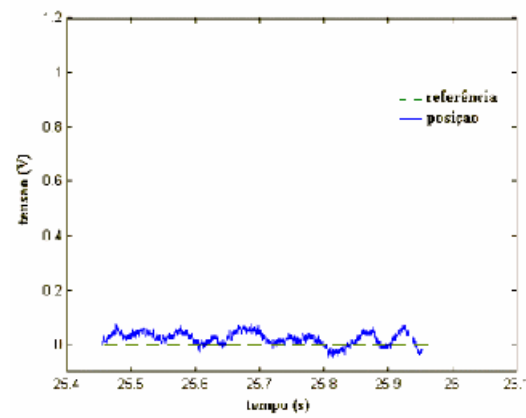


Figura 4.30: Forma de onda com correção do erro em regime

4.4.2 Rastreamento de uma trajetória comandada utilizando MPC

Nesta seção serão observadas quais as influências exercidas sobre o comportamento da planta quando são alterados os parâmetros α , N , Nu , λ e δ do controlador preditivo. Em seguida o LMPC será ajustado, com o que se considerou sendo a melhor sintonia, e será determinada a maior excursão realizada pela esfera metálica. Por fim realizar-se-á uma breve comparação com o comportamento da planta obtido através de um compensador em avanço de fase.

É interessante lembrar que os procedimentos para ajuste dos parâmetros em ambiente MPC ainda não estão bem definidos na literatura. Existem guias que auxiliam no ajuste destes parâmetros, porém são guias criados para serem utilizados junto aos programas comerciais, tal como DMC por exemplo. Em [32], pôde-se observar que para sistemas com período de

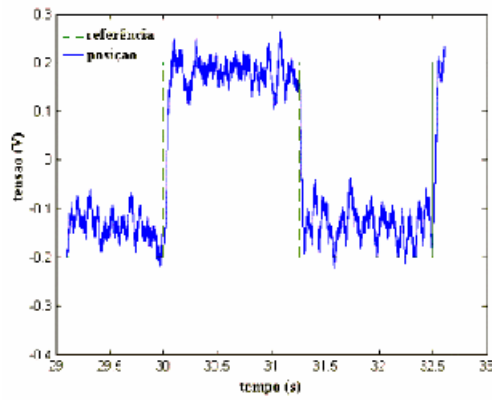
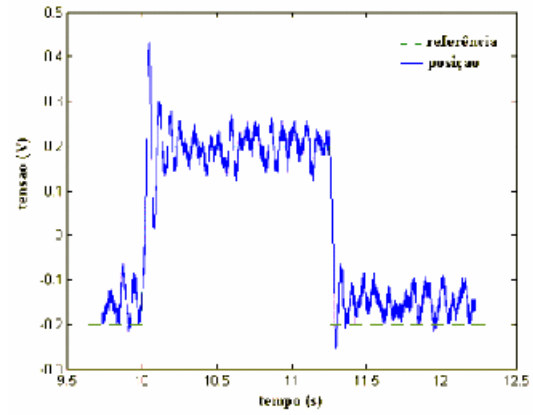
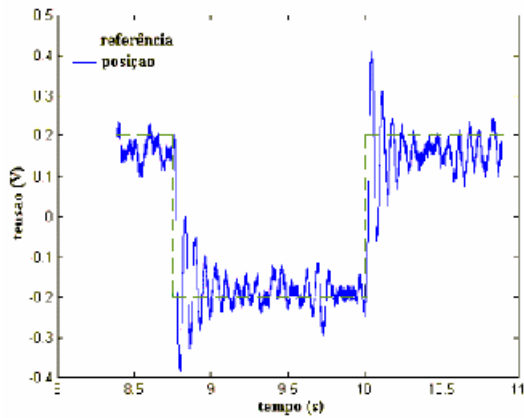
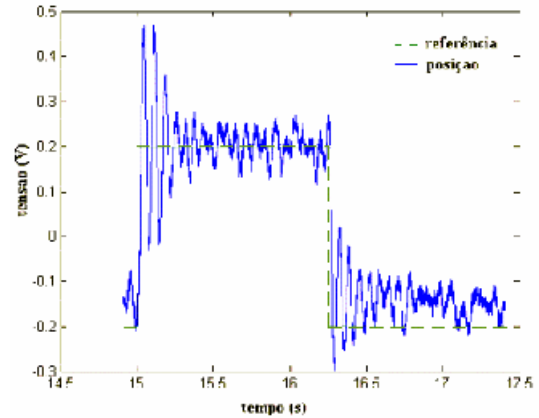
amostragem diferentes, o tamanho do horizonte de predição adequado altera-se consideravelmente. Assim os valores aqui utilizados podem não corresponder a bons valores para outros sistemas.

Parâmetro α

Iniciou-se a análise do efeito do parâmetro α com um valor igual a 0.9, considerando assim uma trajetória de referência bastante suave, para a partir de então subtrair 0.2 a cada novo teste.

Os parâmetros fixos utilizados foram $\delta = 100$, $\lambda = 70$, $N = 28$, $Nu = 21$.

É possível observar diferenças consideráveis na resposta do SLM quando se altera o parâmetro α do controlador MPC. Por exemplo, para $\alpha = 0.9$ a posição da esfera segue a trajetória comandada de forma suave, sem apresentar grande sobresinal porém de forma um pouco mais lenta do que para os outros casos, como ilustra a figura 4.31. Já ao diminuir o valor deste parâmetro, observa-se que a resposta do SLM torna-se mais rápida, porém mais oscilatória, como apresentado nas figuras 4.32, 4.33 e 4.34, chegando ao ponto de se tornar instável. Assim, é possível afirmar que valores de α pequenos resultam em sistemas com resposta rápida, porém pouco amortecida. É possível ainda encarar o parâmetro α como sendo um parâmetro de amortecimento, pois através de seu ajuste é possível gerar uma resposta superamortecida, caso de valores de α grande, e oscilatória, caso de valores de α pequenos.

Figura 4.31: $\alpha = 0.9$ Figura 4.32: $\alpha = 0.7$ Figura 4.33: $\alpha = 0.5$ Figura 4.34: $\alpha = 0.3$

Para $\alpha = 0.2$ o sistema torna-se instável, porém a partir de valores de α menores que 0.5, o sistema já se torna bastante oscilatório.

Parâmetros δ e λ

Na análise dos parâmetros δ e λ , deve-se observar que só existe mudança no comportamento da planta quando a relação entre eles é alterada, ou seja a ordem de grandeza de ambos não influencia no desempenho do sistema desde que a relação seja mantida. Assim valores de $\delta = 100$ e $\lambda = 70$ geram a mesma matriz ganho que valores de $\delta = 10$ e $\lambda = 7$, por

exemplo, porém uma vez alterada a relação entre estes parâmetros altera-se o comportamento em malha fechada do sistema. As figuras 4.35, 4.36, 4.37 e 4.38 mostram o comportamento o sistema para diferentes relações entre δ e λ .

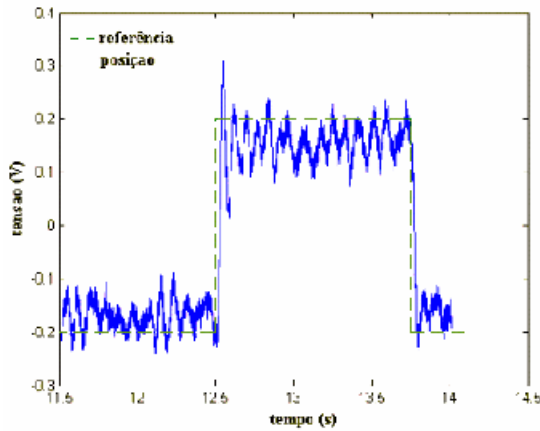


Figura 4.35: $\delta = 10$ e $\lambda = 10$

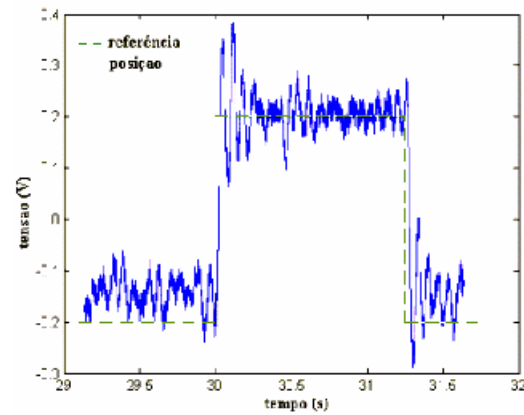


Figura 4.36: $\delta = 20$ e $\lambda = 10$

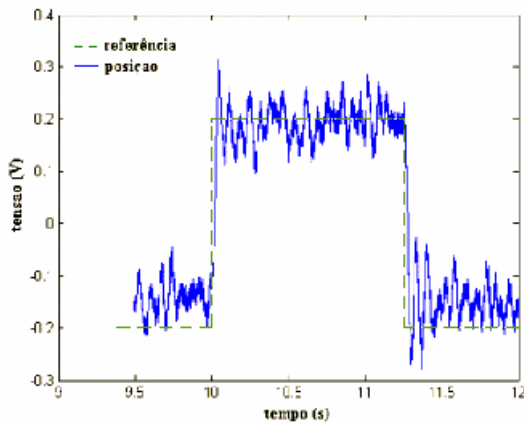


Figura 4.37: $\delta = 30$ e $\lambda = 10$

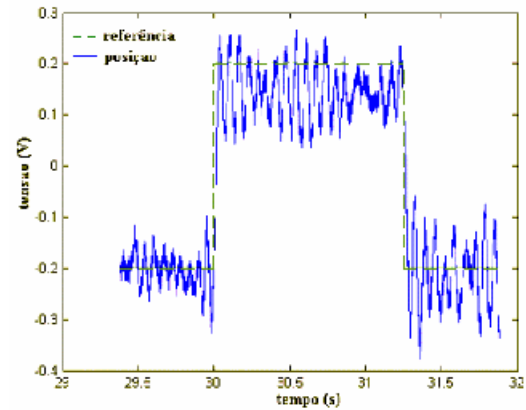


Figura 4.38: $\delta = 40$ e $\lambda = 10$

Analisando as figuras acima pode-se observar que razões de δ/λ maiores geram respostas mais oscilatórias até chegar a um limite de estabilidade que ocorre para $\delta = 50$ e $\lambda =$

10, ou seja valores de δ cinco vezes maiores do que λ tornam o conjunto SLM + MPC instável.

Parâmetro N

Todas as simulações desta seção foram realizadas utilizando-se os seguintes parâmetros: $\lambda = 70$, $\delta = 100$, $\alpha = 0.8$ e $Nu = 7$.

Analisando as figuras 4.39, 4.40, 4.41 e 4.42 é possível observar que valores menores de horizonte de predição geram respostas mais bruscas, que apresentam maior sobresinal, e tornam o sistema mais oscilatório.

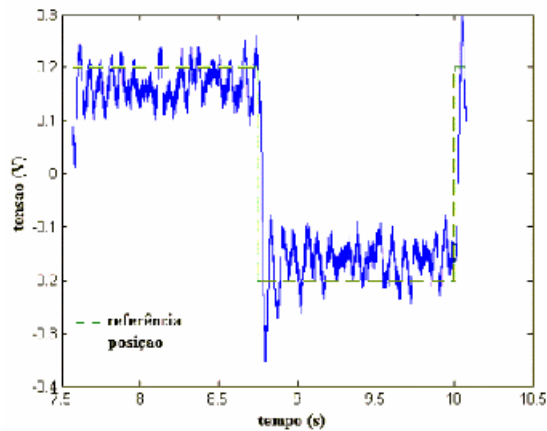


Figura 4.39: N = 28

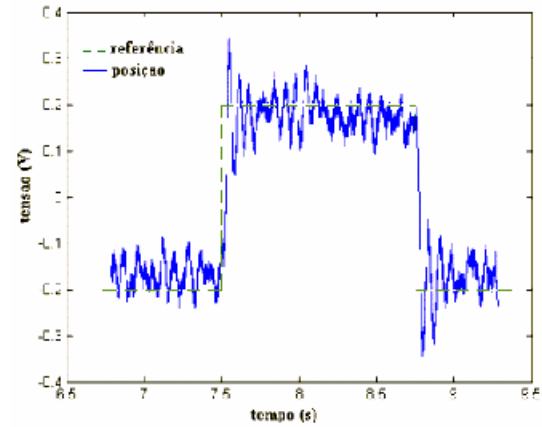


Figura 4.40: N = 21

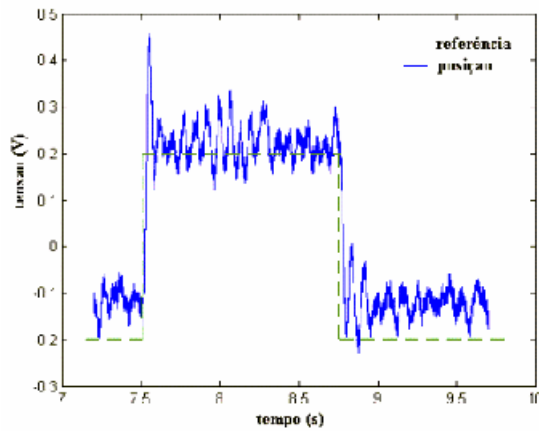


Figura 4.41: N = 14

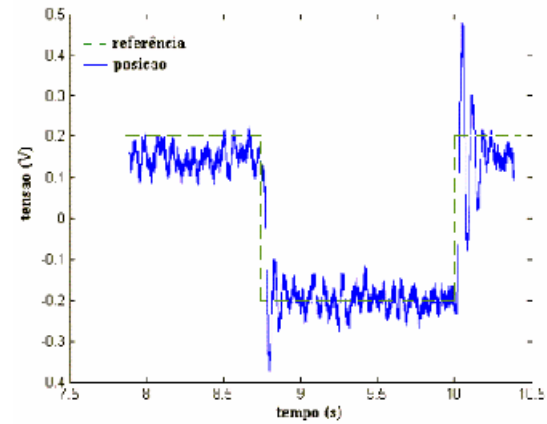


Figura 4.42: N = 7

Parâmetro Nu

Todas as simulações desta seção foram realizadas utilizando-se os seguintes parâmetros: $\lambda = 70$, $\delta = 100$, $\alpha = 0.8$ e $N = 28$.

Analisando as figuras 4.43 e 4.44 observa-se que o comportamento de ambas parece não se alterar consideravelmente, como já observado em resultados de simulação.

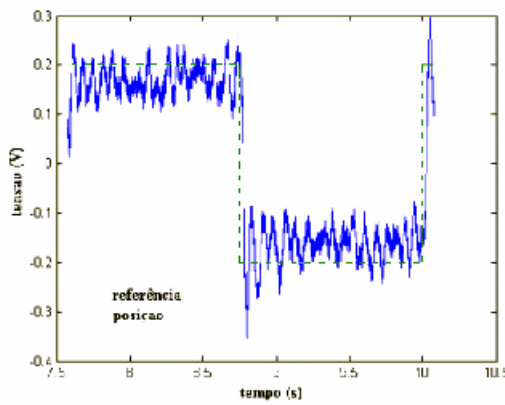


Figura 4.43: Nu = 28

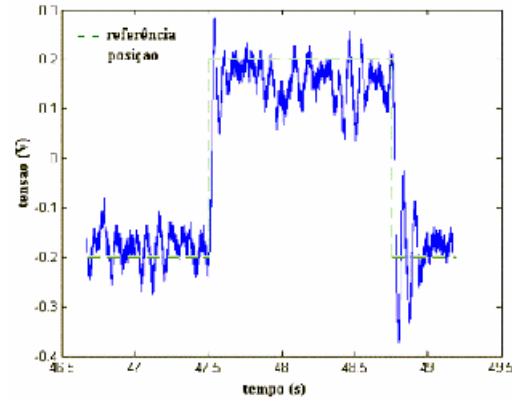


Figura 4.44: Nu = 14

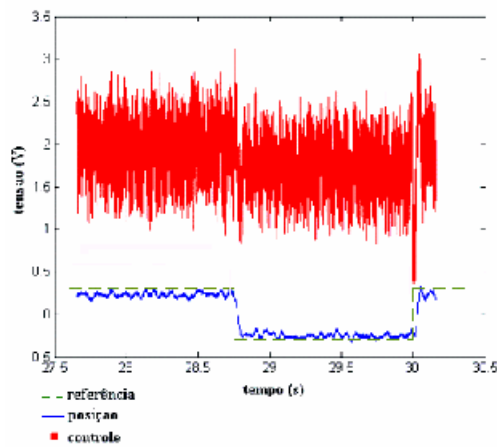
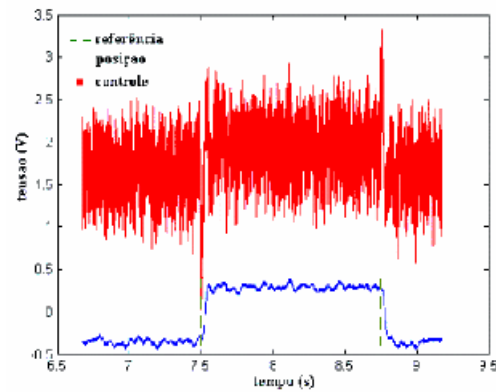
Determinação da maior excursão possível da esfera metálica

Esta seção é dedicada a determinar qual a maior excursão que pode ser realizada pela esfera metálica, sem perda da estabilidade. Foram realizadas diversas medições para diferentes valores de excursão, tal como mostra a tabela 4.6.

Tabela 4.6: Relação entre os respectivos sinal de referência, excursão e figura.

Valor de pico a pico do sinal de referência (V)	Excursão equivalente (cm)	Valor de pico a pico do sinal de saída (V)	Figura ilustrativa
0.6	0.2	0.6	4.40
0.8	0.25	0.8	4.41
1.0	0.3	1.0	4.42
1.2	0.35	1.15	4.43

Os valores dos parâmetros utilizados para determinação da maior excursão possível da esfera são $N = 28$, $Nu = 21$, $\lambda = 10$, $\delta = 20$ e $\alpha = 0.9$.

**Figura 4.45: excursão 0.6V****Figura 4.46: excursão 0.8V**

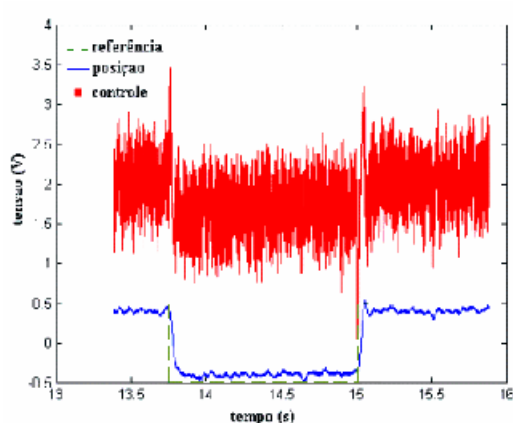


Figura 4.47: excursão 1V

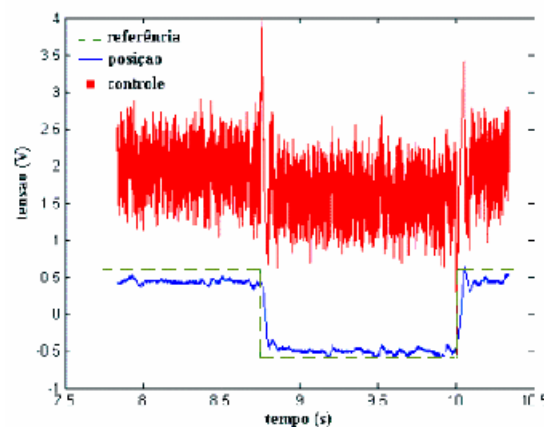


Figura 4.48: excursão 1.15V

Para excursões acima de 1.2V o sistema torna-se instável.

Compensador Lead

Para efeito de comparação observou-se a forma de onda referente à posição da esfera do sistema de levitação magnética, quando este estava sendo controlado por um compensador lead (em avanço). As formas de onda podem ser analisadas a partir da figura 4.49.

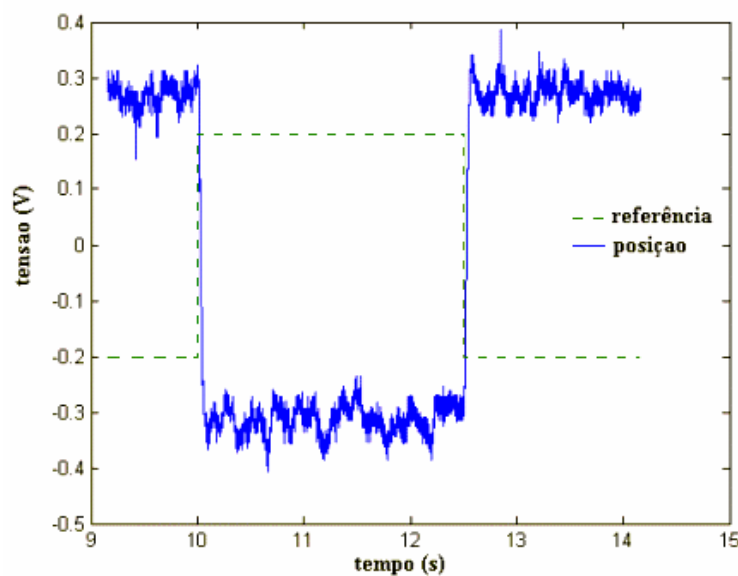


Figura 4.49: Simulação utilizando compensador lead.

Comparando as respostas obtidas pelo compensador lead e pelo controlador MPC, observa-se que para o primeiro caso existe um erro em regime, que pode ser reduzido aumentando-se o ganho, porém esta ação gera respostas com sobresinal. Já para o caso MPC foi possível obter uma resposta sem sobresinal e com erro em regime bastante menor quando comparado ao compensador em avanço, o que caracteriza este último como possuindo um melhor ganho em baixas frequências, devido a inserção da variação do controle na formulação MPC, e a introdução do estado associado à perturbação de saída.

Capítulo 5

CONCLUSÃO

5.1 Conclusões Finais

O principal objetivo deste trabalho consistiu na implementação em tempo real de uma estratégia de controle preditivo aplicada a um sistema de levitação magnética, que é uma planta com dinâmica não-linear, instável em malha aberta e relativamente rápida. O algoritmo MPC foi escrito inicialmente em linguagem Matlab, e em seguida codificado para linguagem “C/C++”, de forma a possibilitar sua utilização junto a estrutura S-Function e xPC Target.

Nesta seção serão realizadas conclusões referentes ao desenvolvimento, simulação, implementação e desempenho de tal estrutura e estratégia de controle.

Para uma melhor compreensão as principais conclusões deste trabalho foram divididas em tópicos.

a) Modelo do processo

Foi observado que um bom modelo contribui para um melhor desempenho do controlador MPC. Tal fato foi verificado ao se utilizar o algoritmo LMPC junto com o circuito construído a partir de amplificadores operacionais, o qual representa com maior precisão a dinâmica do modelo utilizado. Percebeu-se que a resposta do sistema (planta + MPC) não apresentava erro em regime, conseqüentemente a ordem do modelo pode ser reduzida o que implica em um menor custo computacional.

b) Tratamento diferenciado para plantas estáveis e instáveis

Sabe-se que a metodologia de controle preditivo gera sua lei de controle através da minimização em malha aberta de um certo índice de desempenho sujeito à dinâmica da planta e a eventuais restrições impostas pelo sistema físico. E que plantas e modelos instáveis em malha aberta devem possuir um tratamento diferenciado quando comparado a plantas e modelos estáveis em malha aberta. Primeiro, modelos instáveis devem ser realinhados à planta a cada iteração; em segundo, para se realizar a correção do erro em regime em modelos instáveis, o simples fato de se inserir uma perturbação na predição não é o suficiente para sanar este problema, sendo necessário aumentar a ordem do modelo tal como descrito no capítulo 3.

c) Plataforma xPC Target

A utilização de S-Functions escritas em C/C++ em conjunto com a ferramenta xPC Target mostrou ser uma boa tecnologia para implementação de controle em tempo real, desde que a velocidade do sistema não seja muito grande. No caso do SLM o conjunto mostrou-se no limiar de sua capacidade de operação para o equipamento computacional empregado. Um aspecto a ser destacado é o fato de algoritmos escritos em linguagem C++ necessitarem de um poder computacional muito maior quando comparado a algoritmos escrito em linguagem C. Ao solicitar que algoritmos, de mesma função, escritos em C e C++ realizassem uma determinada tarefa, observou-se que a diferença de tempo médio de execução de tarefas gira em torno de 2 a 5 vezes, dependendo dos horizontes de controle e predição utilizados.

d) Ajuste de parâmetros em ambiente MPC

O ajuste de parâmetros em ambiente MPC mostrou ser uma tarefa que requer tempo, paciência e bons conhecimentos da planta a ser controlada. Foi observado que períodos de amostragem diferentes necessitam de parâmetros diferentes e o grau de descasamento entre o modelo e a planta também influencia na adequação dos valores dos parâmetros. Este último fato foi observado ao comparar valores de parâmetros utilizados em simulação e experimentação do sistema de levitação magnética. Por isso é difícil criar regras gerais para o ajuste de parâmetros em ambiente MPC.

e) Desempenho do sistema

Foi verificado que a resposta do controlador torna-se mais amortecida quando a relação entre os pesos δ e λ é pequena, o mesmo acontece para valores grandes de horizontes de controle e predição. Porém apesar de um pior desempenho o sistema torna-se menos oscilatório.

Os resultados finais foram bastante animadores, pois foi possível implementar uma metodologia de controle moderno, que é a preditiva, em um sistema “mal-comportado”, obtendo um controle de posição da esfera metálica sem erro em regime e com uma rápida resposta a mudanças de trajetória.

5.2 Recomendações para trabalhos futuros

Por fim são propostos alguns tópicos, os quais podem ser estudados com o objetivo de dar continuidade a este trabalho.

- Reescrever as rotinas que tratam matrizes, como a classe DMatrix, em linguagem C, de forma a otimizar o custo computacional.

- Gerar novas leis de controle preditivo alterando-se a função objetivo, como por exemplo mudar o tipo de norma utilizada, de forma a visualizar as alterações em desempenho.
- Utilizar outra tecnologia de controle em tempo real que não seja a xPC Target.
- Aplicar técnicas de controle preditivo em sistemas multicamadas, ou seja implementar uma lei de controle preditivo em outros níveis hierárquicos.
- Inserir técnicas de controle adaptativo de forma a lidar com a magnetização da esfera, e até mesmo pensando em uma aplicação onde a massa do elemento a ser levantado é variável, como acontece nos trens que utilizam levitação magnética.
- Utilizar técnicas de controle preditivo baseado em modelo não-linear.
- Levar em consideração as incertezas referente a erros de modelamento utilizando técnicas de controle preditivo robusto.
- Implementar o MPC com restrições em tempo real.
- Analisar analiticamente a estabilidade em malha fechada através de qualquer uma das duas abordagens, seja através de uma restrição de contração, ou através da utilização da função objetivo como função de Lyapunov.

Referências Bibliográficas

- [1] AGUIRRE, L. A.. **Introdução à identificação de sistemas: técnicas lineares e não-lineares aplicadas a sistemas reais**. Belo Horizonte: UFMG, 2000. 554p.
- [2] AL-GHAZZAWI A. et al. On-line tuning strategy for model predictive controllers. **Journal of Process Control**, v.11, p.265-284. 2001.
- [3] BECERRA, V. M.. **DMATRIX: A C++ Matrix Class**. Reference Manual. Reino Unido. 79p.
- [4] BEMPORAD, A.; CHISCI, L.; MOSCA, E. On the stabilizing property of SIORHC. **Automatica**, v.30, p.2013-2015. 1994.
- [5] BLET, N. ; MEGÍAS, D. ; SERRANO, J. ; DE PRADA, . Nonlinear MPC versus MPC using on-line linearisation – A Comparative Study. In: 15th Triennial World Congress, **Proceedings...** Barcelona: IFAC. 2002.
- [6] CAMACHO, E. F.; BORDONS, C. **Model predictive control**. Londres: Springer. 1999. 280p.
- [7] CLARKE, D. W.; MOHTADI, C.; TUFFS, P. S. Generalized predictive control. **Automatica**, v.23, p.137-160, 1987.
- [8] COELHO, L. S. ; COELHO, A. A. R.. **Controle preditivo neural aplicado a processo não-linear multivariável: Projeto e Implementação**. Florianópolis: CBA, 2000. (trabalho CBA311).

- [9] COUTINHO, J. Conexão entre PCs utilizando cabos seriais. Disponível em <http://geocities.yahoo.com.br/conexaopcpc/cabo_serial_e_paralelo.htm>. Acesso em : 22 set. 2003.
- [10] CUTLER, C. R.; RAMAKER, B. L. Dynamic matrix control – a computer control algorithm. In: JOINT AUTOMATIC CONTROL CONFERENCE, 1980, San Francisco. **Proceedings...** São Francisco: [S.n.t.], 1980. p. artigo WP5-B.
- [11] CHEN, M. ; WU, K. ; FU, L.. Design, implementation and self-tuning adaptive control of maglev guiding system. **Mechatronics**, v.10, p.215-237. 2000.
- [12] CHO, D.; KATO, Y.; SPLIMAN, D. Sliding mode and classical control of magnetic levitation systems. **IEEE Control Systems**, v.13, n.1, p.42-48, 1993.
- [13] DAVISON, E. J.; SMITH, H. W. Pole assignment in linear time-invariant multivariable systems with constant disturbance. **Automatica**, v.7, p.489-498. 1971.
- [14] FRANKLIN, G. F.; POWELL, J. D.; EMAMI-NAEINI, A. **Feedback Control of Dynamics Systems**. Stanford: Addison-Wesley. 1986. 778p.
- [15] GARCÍA, C. E.; PRETT, D. M.; MORARI, M.. Model predictive control: theory and practice – a survey. **Automatica**, v.25, n.3, p.335-348. 1989.
- [16] GREEN, S. A. ; CRAIG, K. C.. Robust, Digital, Nonlinear Control of Magnetic-Levitation Systems. **Transaction of the ASME**, v.120, p.488-495. 1998.
- [17] GRIMM, C. **Um controlador digitalmente assistido para um sistema de levitação magnética**. 68f. 2002. Dissertação (Mestrado em Sistemas e Controle) - Instituto Tecnológico de Aeronáutica, São José dos Campos.
- [18] HENSON, M. A.. Nonlinear Model Predictive Control: Current Status and Future Directions. **Computers and Chemical Engineering**, v.23, p.187-202. 1998.
- [19] KAPLAN, B. Z.; SARAFIAN, G. Control strategy for stabilising magnetic levitation. **Electronics Letters Online**, v.33, n.23, p.1960-1961, 1997.

- [20] MACIEJOWSKI, J. M. **Predictive Control with Constraints**. Reino Unido: Prentice-Hall. 2002. 331p.
- [21] Magnetic Levitation System. Feedback Instruments Ltd. 1997.
- [22] Manual do usuário PCL-711B.
- [23] MAYNE, D. Q. et al. Constrained model predictive control: stability and optimality. **Automatica**, v.36, p.789-914. 2000.
- [24] MIURA, R. O. ; GALVÃO, R. K. H.. Controle preditivo de um sistema de levitação magnética. In: Congresso Temático em Dinâmica, Controle e Aplicações. 2003. **Anais...** São José dos Campos: ITA. 2003. 1 CD.
- [25] MOHTADI, C. **Advanced self-tuning algorithms**. 1986. Thesis. (PhD in Department of Engineering Science) - Oxford University, Oxford.
- [26] MORARI, M. ; LEE, J. H.. Model Predictive Control: Past, Present and Future. **Computers and Chemical Engineering**, v.23, p.667-682. 1999.
- [27] MURRAY, W. ; GILL, P. E. ; WRIGHT, M.H.. **Practical Optimazation**. London: Academic Press. 1981.
- [28] MUSKE, K. R. ; BADGWELL, T. A.. Disturbance modeling for offset-free linear model predictive control. **Journal of Process Control**, v.12, p.617-632. 2002.
- [29] OZKAN, L. ; KOTHARE, M.V. ; GEORGAKIS, C.. Model predictive control of nonlinear systems using piecewise linear models. **Computers and Chemical Engineering**, v.24, p.793-799. 2000.
- [30] PANNOCCHIA, G ; RAWLINGS, J. B. Disturbance Models for Offset-Free Model-Predictive Control. **AIChE Journal**, v.49, n.2, p.426-437. 2003.
- [31] QIN, S. J. ; BADGWELL, T. A.. A survey of industrial model predictive control technology. **Control Engineering Practice**, v.11, p.733-764. 2003.

- [32] RANI, K. Y. ; UNBEHAUEN, H.. Study of predictive controller tuning methods. **Automatica**, v.33, n.12, p.2243-2248. 1997.
- [33] RAO, C. V.; RAWLINGS, J. B. Linear programming and model predictive control. **Journal of Process Control**, v.10, p.283-289, 2000.
- [34] Revista CBPF. **Fenômenos e materiais magnéticos**. Disponível em <<http://www.cbpf.br/RevistaCBPF/pdf/Magnetismo.pdf>> Acesso em: 15 ag. 2003.
- [35] RICHALET, J. et al. Model predictive heuristic control: applications to industrial processes. **Automatica**, v.14, p.413-428. 1978.
- [36] RODRIGUES, M. A.. **Estabilidade Robusta de Controladores Preditivos**. 147f. 2001. Tese (Doutorado em Engenharia Química) Escola Politécnica da Universidade de São Paulo. São Paulo.
- [37] SINHA, P. K. ; PECHEV, A. N.. Model reference adaptive control of a maglev system with stable maximum descent criterion. **Automatica**, v.35, p.1457-1465. 1999.
- [38] SINHA, P. K.. **Eletromagnetic Suspension: Dynamics & Control**. IEE Control Engineering Series v.30. Londres.1987. 290p.
- [39] TOBEY ; GRAEME ; HUELSMAN. **Operational Amplifier: Design and Applications**. International Student Edition. Burr-Brown McGraw-Hill. 1971. 473p.
- [40] TRUMPER, D. L.; OLSON, S. M.; SUBRAHMANYAN, P. K. Linearizing control of magnetic suspension systems. **Actions on Control Systems Technology**, v.5, n.4, p.427-438. 1997.
- [41] ZANIN, A. C. **Implementação industrial de um otimizador em tempo real**. 161f. 2001. Tese. (Doutorado em Engenharia Química) - Escola Politécnica da Universidade de São Paulo. São Paulo.
- [42] ZHENG, A. Reducing on-line computational demands in model predictive control by approximating QP constraints. **Journal of Process Control**, v.9, p.279-290, 1999.

- [43] ZHURAVLYOV, Y.N.. On LQ-Control of Magnetic Bearing. **IEEE Transaction on Control Systems Technology**, v.8, n.2, p.344-350. 2000.

Anexos

Anexo A – Ferramentas Computacionais

A.1 S-Function

A.1.1 Introdução

S-Function (System Functions) é uma ferramenta utilizada junto ao Simulink, que provê um mecanismo poderoso para extensão das capacidades do último. Através das S-Functions é possível criar blocos para modelos do Simulink, que por sua vez é um programa que roda em companhia do Matlab. O Simulink fornece uma interface gráfica ao usuário (GUI, Graphical User Interface) que é usada na construção de diagrama de blocos, execução de simulações, além de permitir a análise dos resultados.

Para utilizar uma S-Function em um modelo do Simulink, é necessário arrastar o bloco referente a uma S-Function da biblioteca de Funções do Simulink para dentro do modelo. E então especificar o nome do arquivo a ser utilizado pelo bloco S-Function junto ao campo “S-Function name”, que pode ser acessado dando-se dois cliques junto ao bloco referente a S-Function.

É possível também passar parâmetros a uma S-Function, para isso é necessário conhecer quais parâmetros são requisitados pela S-Function e a ordem de solicitação dos mesmos. Os parâmetros devem ser inseridos no campo “S-Function parameters”, e devem estar separados por vírgula.

No Simulink, os modelos obedecem à uma hierarquia, permitindo a visualização de um sistema complexo em um único bloco, e dando dois cliques neste bloco através do mouse, expande-se este sistema em sub-sistemas, podendo assim caminhar-se através dos níveis da

hierarquia. Cada bloco no Simulink é dirigido por uma S-Function. Uma S-Function é uma descrição em linguagem de computador de um bloco do Simulink. S-Functions podem ser escritas em Matlab, C, C++, Ada ou Fortran.

As S-Functions escritas em outras linguagens que não seja a Matlab devem ser compiladas com o utilitário MEX (Matlab Executable). Este utilitário permite o Matlab rodar arquivos compilados no ambiente Simulink. E podem ainda ser utilizada com a caixa de ferramentas “Real-Time”.

Uma S-Function é mais comumente utilizada para resolver tarefas, tais como:

- Adicionar novos blocos de utilização geral ao Simulink;
- Adicionar blocos que representem “drivers” de dispositivos de hardware;
- Incorporar o código C em uma determinada simulação;
- Descrever sistemas como um conjunto de equações.

A.1.2 Funcionamento de uma S-Function

Um bloco do Simulink consiste de um conjunto de entradas, um conjunto de estados, e um conjunto de saídas, onde as saídas são uma função do período de amostragem, das entradas, e dos estados do bloco.

A execução de um modelo no Simulink procede em estágios. Primeiro vem a fase de inicialização. Nesta fase, o Simulink incorpora a biblioteca de blocos existentes dentro do modelo, propaga as dimensões e os tipos dos dados, o período de amostragem, avalia os parâmetros dos blocos, determina a ordem de execução dos blocos, e aloca a memória. Então o Simulink entra no loop de simulação, onde cada passo do loop é referenciado como sendo um passo da simulação. Durante a cada passo da simulação, o Simulink executa cada bloco do modelo na ordem pré-determinada na inicialização. Para cada bloco, o Simulink chama

funções que calculam os estados, suas derivadas, e as saídas do bloco para cada período de amostragem. E isto continua até a simulação ser completada.

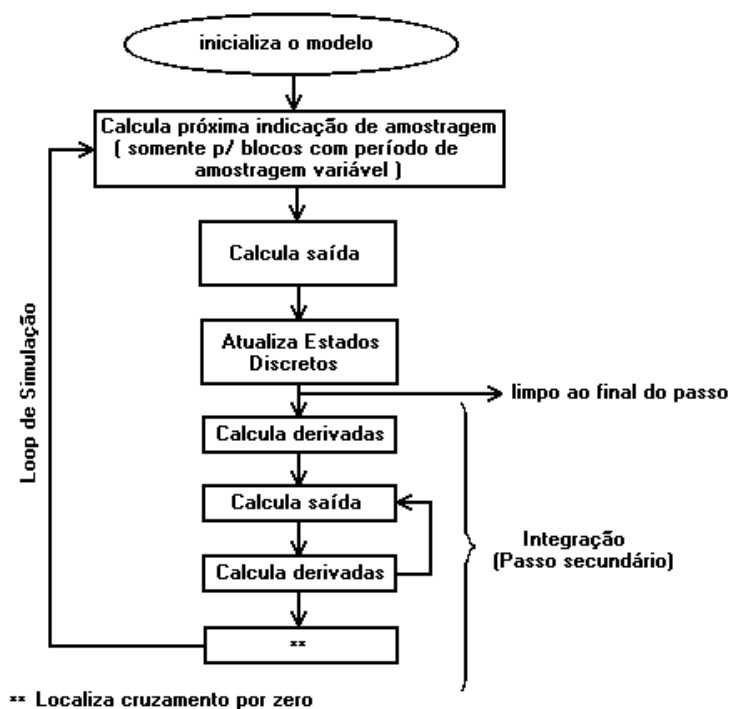


Figura A.2: Sequência das tarefas realizadas.

Uma S-Function engloba um conjunto de métodos de chamada que executam tarefas necessárias a cada estágio da simulação. Durante a simulação de um modelo, a cada estágio de simulação, o Simulink chama métodos apropriados para cada bloco de S-Function no modelo. As tarefas executadas por estes métodos incluem:

① Inicialização – É a primeira tarefa a ser executada em uma simulação, pois é aqui que o Simulink inicializa a S-Function. Durante este estágio, o Simulink:

- Inicializa o SimStruct, uma estrutura da simulação que contém informações sobre a S-Function
- Ajusta o número e dimensão das portas de entrada e saída
- Ajusta o período de amostragem do bloco

- Aloca áreas de armazenamento e tamanho de vetores.

② Cálculo do próximo período de amostragem – Caso tenha sido criada uma S-Function com período de amostragem variável é neste passo onde se calcula o tamanho do próximo passo.

③ Cálculo das saídas para o passo atual – Após esta chamada ter sido completada, todas as portas de saída do bloco estão válidas para o passo atual.

④ Atualiza os estados discretos para o passo atual – Nesta chamada, todos os blocos devem executar atividades realizadas uma vez por passo, como por exemplo atualizar os estados discretos para o próximo passo da simulação.

⑤ Integração – Esta chamada é aplicada a modelo com estados contínuos.

A.1.3 Formatos de uma S-Function

Tal como já dito uma S-Function pode ser escrita em diversas linguagens de computador. Caso seja escrita em linguagem Matlab, será um arquivo .M, e caso seja escrita em qualquer outra linguagem permitida, será um arquivo .MEX.

→ S-Function como arquivo .M

Este tipo de S-Function consiste de uma função Matlab da seguinte forma:

`[sys,x0,str,ts]=f(t,x,u,flag,p1,p2,...)`

onde *f* é o nome da S-Function, *t* é o tempo atual, *x* é o vetor de estados da S-Function, *u* é o vetor de entradas da S-Function, *flag* indica a tarefa a ser executada, e *p1,p2,...* são os parâmetros do bloco. Durante a simulação de um modelo, o Simulink invoca *f* repetidamente,

usando o flag para indicar qual tarefa deve ser executada. Toda vez que a S-Function executar uma tarefa, ela retornará o resultado em uma estrutura já pré-determinada.

A tabela a seguir lista os estágios de simulação, a rotina, e seu respectivo flag.

Tabela 2. Relação entre estágio de simulação, rotina e flag.

Estágio de Simulação	Rotina	Flag
Inicialização	mdlInitializaSizes	0
Cálculo do próximo período de amostragem	mdlGetTimeOfNextVarHit	4
Cálculo da Saída	mdlOutputs	3
Atualização dos estados discretos	mdlUpdates	2
Cálculo das derivadas	mdlDerivatives	1
Fim das tarefas de simulação	mdlTerminate	9

Para o Simulink reconhecer uma S-Function escrita em arquivo .m, é necessário especificar algumas informações sobre a S-Function, tais como, número de entradas, saídas, estados entre outras. E para isso utiliza-se a função “simsizes”, veja tabela a seguir:

Tabela 3. Descrição dos campos.

Nome do Campo	Descrição
<code>Sizes.NumContStates</code>	Número de Estados Contínuos
<code>Sizes.NumDiscStates</code>	Número de Estados Discretos
<code>sizes.NumOutputs</code>	Número de Saídas
<code>sizes.NumInputs</code>	Número de Entradas
<code>Sizes.DirFeedthrough</code>	Flag que indica ligação direta entre entrada e saída
<code>sizes.NumSampleTimes</code>	Número de Amostras

Após a inicialização da estrutura de tamanhos é necessário chamar-se a função “simsizes” novamente, e encaminhá-la ao vetor sys, através da linha de comando:

```
sys = simsizes(sizes);
```

para que o Simulink tenha conhecimento dos valores contidos nesta estrutura.

→ S-Function como arquivo .MEX

Da mesma forma que uma S-Function escrita em Matlab (arquivo M), uma S-Function escrita como arquivo MEX (linguagem C, C++, Fortran , Ada) consiste de um conjunto de chamada de rotinas que o Simulink invoca para realizar diversas tarefas durante a simulação. Uma diferença está na forma de chamada das rotinas, ou seja utilizando arquivos M o Simulink invoca tarefas a S-Function através dos flags, porém em arquivos MEX esta solicitação é feita através do nome das rotinas, nomes estes que são considerados padrão pelo Simulink, portanto qualquer alteração destes nomes pode resultar no não funcionamento da S-Function. Uma outra diferença é que o conjunto de funções de chamadas utilizadas em

arquivos MEX é muito maior daqueles que podem ser utilizados em arquivos M, além das funções MEX terem acesso direto a estrutura interna de dados, denominada SimStruct.

→ Comparação entre S-Functions MEX e M

S-Functions M e MEX possuem suas vantagens e desvantagens. A vantagem de S-Functions M está relacionada a velocidade de desenvolvimento, além de permitir fácil acesso a funções e ferramentas do Matlab.

Já a principal vantagem de S-Functions escritas como arquivos MEX é a versatilidade. O grande número de rotina de chamada e o acesso ao SimStruct permitem as funções MEX implementar funcionalidades não acessíveis a funções M. Tal funcionalidade inclui a habilidade de tratar com outros dados que não sejam double, complexos, matrizes, e etc. Além da utilização junto a caixa de ferramentas de tempo-real.

A.1.4 Escrevendo uma S-Function em C

Um arquivo C – MEX que define um bloco S-Function deve prover informação sobre o modelo para o Simulink durante a simulação. Assim que a simulação é inicializada, o Simulink, o solucionador de EDO, e o arquivo MEX interagem pra realizar tarefas específicas. Estas tarefas incluem definição de condições iniciais e características do bloco, cálculo das derivadas, estados discretos, e saídas.

Tal como em uma M S-Function, o Simulink interage com a C MEX S-Function invocando-as através dos métodos de chamada que a S-Function implementa. Cada método executa uma tarefa pré-definida, tal como calcular as saídas do bloco, necessárias para simular o bloco cuja funcionalidade é definida pela S-Function. O Simulink define de maneira geral a tarefa de cada chamada. A S-Function é livre para executar a tarefa de acordo com sua

funcionalidade. Por exemplo, o Simulink especifica que o método mdlOutput calcula a saída do bloco para o instante atual, porém não especifica o que a saída precisa ser.

Obs.: Ao contrário das M S-Functions, as C MEX S-Functions não possuem o parâmetro flag. Isto deve-se ao fato do Simulink chamar cada método da S-Function diretamente em seu instante apropriado durante a simulação.

A.1.5 O SimStruct

O arquivo simstruct.h é um arquivo de cabeçalho escrito em C que define a estrutura de dados do Simulink e a macro de acesso ao SimStruct. Ele envolve todos os dados relacionados ao modelo ou S-Function, incluindo os parâmetros e saídas do bloco.

Existe uma estrutura de dados SimStruct alocada para o modelo Simulink. Cada S-Function no modelo tem sua própria SimStruct associada. A organização desta SimStruct é parecida com uma árvore de diretórios. A SimStruct associada com o modelo é a SimStruct raiz. A SimStruct associada com uma S-Function é a SimStruct filho.

A.1.6 Compilando C S-Functions

S-Functions podem ser compiladas em um dos três modos identificados pela presença de uma das seguintes definições:

- `MATLAB_MEX_FILE` - indica que a S-Function está sendo construída como um arquivo MEX para uso com o Simulink.
- `RT` – indica que a S-Function está sendo construída com a ferramenta de tempo real dando origem a uma aplicação que possa ser executada em tempo real usando um passo fixo para o solucionador.

- NRT – indica que a S-Function está sendo construída com a ferramenta de tempo real, porém a aplicação resultante não será utilizada em tempo real podendo utilizar um passo variável para o solucionador.

A linha de comando que deve ser utilizada para compilação é:

No prompt da janela de comandos do Matlab, digite:

> mex nomedoarquivo.c (para caso de C SFunction)

ou

> mex nomedoarquivo.cpp (para caso de C++ SFunction)

Para se construir S-Functions em C++ é necessário utilizar o comando mex, da mesma forma como era utilizado em C S-Functions, com a diferença que o arquivo c++ deve ser nomeado como .cxx ou .cpp.

→ Visão dos Dados

Os blocos S-Function possuem sinais de entrada e saída, parâmetros, estados internos, mais outras áreas de trabalho gerais. De modo geral, as entradas e saídas dos blocos são escritas em, ou lidas a partir de, vetores de entrada/saída. As entradas podem ser também provenientes de:

- Entradas externas “interfaceadas” por placas de conversão analógica-digital.
- Terra se o sinal de entrada estiver aterrado ou desconectado.

As saídas dos blocos podem também ser enviadas ao mundo externo, também por interface de dispositivos D/A. Além de sinais de entradas e saídas, S-Functions também podem ter:

- Estados Contínuos;
- Estados Discretos;

- Outras ares de trabalho tais como vetores de reais, de inteiros, ou de ponteiros.

A rotina `mdlInitializeSizes` da S-Function ajusta o tamanho de vários sinais e vetores.

Os métodos das S-Functions que são chamados durante o loop de simulação pode determinar o tamanho e valor destes sinais.

Um método de uma S-Function pode acessar sinais de entradas de duas formas:

- Via ponteiros
- Utilizando entradas contínuas

A.1.7 Acessando sinais utilizando ponteiros

Durante o loop de simulação, o acesso aos sinais de entrada é feito utilizando-se a seguinte linha de comando:

```
InputRealPtrsType uPtrs = ssGetInputPortRealSignalPtrs(S,portIndex);
```

Este é um vetor de ponteiros, onde `portIndex` começa em 0. E deve haver uma linha de comando desta para cada entrada. Para acessar um elemento deste sinal deve-se utilizar:

```
*uPtrs[element]
```

É interessante destacar que um vetor de ponteiros de entrada pode apontar para pedaços de memória que não sejam seqüenciais.

E o sinal de saída pode ser posto acessível através do seguinte comando:

```
real_T *y = ssGetOutputPortSinal(S,outputPortIndex);
```

A.1.8 Acessando sinais de entrada continuamente.

O método `mdlInitializeSizes` de uma S-Function pode especificar que os elementos de um sinal de entrada precisam ocupar uma área seqüencial de memória, utilizando

`ssSetInputPortRequiredContiguous`. Se as entradas são contínuas, outros métodos podem utilizar `ssGetInputPortSignal` para acessar as entradas.

→ Portas de entrada e saída

O Simulink permite que uma S-Function crie e utilize qualquer número de portas de entrada e saída em um bloco. Será explicado como criar e inicializar estas portas de entrada e saída.

→ Criando Portas de Entrada

Para criar e configurar portas de entrada, o método `mdlInitializeSizes` deve inicialmente especificar o número de portas de entrada que a S-Function possui, utilizando `ssSetNumInputPorts`. E então, para cada porta de entrada, o método deve também especificar:

- A dimensão da porta de entrada;
- Se a porta de entrada permite a expansão a escalar das entradas;
- Se o sinal da porta de entrada é utilizado na saída (feedthrough);
- O tipo de dado da porta de entrada;
- O tipo numérico da porta de entrada, se real ou complexo.

→ Criando Portas de Saída

Para criar e configurar portas de saída, o método `mdlInitializeSizes` deve inicialmente especificar o número de portas de saída que a S-Function tem, utilizando `ssSetNumOutputPorts`. E então, para cada porta de saída, o método deve especificar:

- A dimensão da porta de saída;
- O tipo do dado da porta de saída;
- O tipo numérico da porta de entrada, se real ou complexo.

A.1.9 Período de Amostragem

O Simulink suporta blocos que sejam executados a diferentes períodos. Um bloco S-Function pode especificar suas taxas de amostragem como:

- Blocos baseados em período de amostragem;
- Portas baseadas em período de amostragem;
- Sistemas híbridos com bloco e porta baseados em períodos de amostragem.

Com o bloco baseado em período de amostragem, a S-Function especifica um conjunto de taxas de operação para o bloco como um todo durante a fase de inicialização da simulação. Com as portas baseadas em períodos de amostragem, a S-Function especifica um período de amostragem para cada porta de entrada e saída individualmente durante a inicialização. Durante a fase de execução, com bloco baseado em período de amostragem, a S-Function processa todas as entradas e saídas a cada vez que uma indicação de amostragem ocorre para o bloco. Da mesma forma, com a porta baseada em período de amostragem, o bloco processa uma determinada porta somente quando uma indicação de amostragem ocorre para aquela porta.

A.2 QuadProg

É uma ferramenta Matlab utilizada na resolução de problemas de programação quadrática. Assim seu objetivo é encontrar o valor de \mathbf{x} que minimize uma função do tipo:

$$\min_x \left\{ \frac{1}{2} \cdot x^T \cdot H \cdot x + f^T \cdot x \right\} \quad (\text{A.1})$$

sujeita à restrições:

$$\begin{aligned} A \cdot x &\leq b \\ Aeq \cdot x &= beq \\ lb &\leq x \leq ub \end{aligned} \quad (\text{A.2})$$

A sintaxe adequada para utilização correta desta ferramenta está apresentada a seguir:

$x = \text{quadprog}(H, f, A, b, Aeq, beq, lb, ub, x0, options);$

Esta linha de comando é interpretada pelo Matlab, como “minimize a função

$\min_x \left\{ \frac{1}{2} \cdot x^T \cdot H \cdot x + f^T \cdot x \right\}$, sujeita às restrições de desigualdade ($A \cdot x \leq b$), às restrições de

igualdade ($Aeq \cdot x = beq$) e aos limites inferior e superior de \mathbf{x} ($lb \leq x \leq ub$), e em seguida

retorne o vetor \mathbf{x} que minimiza a referida função”.

A.3 xPC Target

A.3.1 Introdução

O xPC Target é uma solução para projetar, desenvolver e testar sistemas de tempo-real que utilizem equipamentos padrão PC. É um ambiente que utiliza um PC escravo, separado de um PC mestre, para executar as aplicações em tempo-real.

Neste ambiente é possível utilizar um computador como mestre com Matlab e Simulink para se criar modelos, que podem então ser simulados, por enquanto ainda não em tempo real.

O xPC Target permite que sejam adicionados blocos de entrada e saída no modelo, e então utilizando em conjunto, o PC mestre, a oficina de tempo-real, e um compilador C/C++, é possível criar-se códigos executáveis. O código executável é então baixado do PC mestre para o PC escravo que deve estar executando o kernel tempo real xPC Target. Após baixar o código, é possível rodar e testar a aplicação em tempo real.

- Equipamentos especiais necessários – O aplicativo xPC Target necessita de um PC mestre, um PC escravo, e para comunicação de entrada e saída, o PC escravo deve possuir placas de conversão A/D, por exemplo.

- Aplicativos especiais necessários – O aplicativo xPC Target precisa também de um dos compiladores, Microsoft Visual C/C++ ou Watcom C/C++, além do Matlab, Simulink, e oficina de tempo real.

O xPC Target é um produto compatível com a plataforma PC, que pode ser instalado em um computador que rode Windows 95, Windows 98, Windows 2000, ou Windows NT.

O xPC Target necessita dos seguintes produtos da MathWorks:

- Matlab – Interface de linha de comando para o xPC Target
- Simulink – Ambiente para modelar sistemas físicos e controladores utilizando diagrama de blocos
- Oficina de Tempo Real – Converte blocos do Simulink em um código escrito em linguagem C.
- Compilador C – Necessário para se compilar e criar a aplicação alvo.

O computador escravo não precisa de nenhum sistema operacional, tal como Windows, UNIX ou até mesmo o DOS, e por isso deve ser inicializado com um disquete próprio. Este disquete contém o kernel do xPC Target. O kernel faz com que o computador escravo fique pronto para baixar as aplicações vindas do computador mestre, aplicações estas que devem estar todas em linguagem C. Após baixada as aplicações o computador escravo está pronto para uso em tempo real.

A.3.2 Aplicação xPC Target

Antes de criar uma aplicação alvo, é necessário que se crie um modelo no Simulink. Após ter-se criado o modelo, este poderá ser executado, porém não ainda em tempo real.

O xPC Target irá então utilizar este modelo, a oficina de tempo-real, e um compilador C/C++ para criar uma aplicação alvo.

Inicialmente é interessante gerar o disco de inicialização que irá carregar o kernel do xPC Target no computador escravo. Para isso basta seguir os seguintes passos:

1. insira um disquete formatado no drive A;
2. digite “xpcsetup” no prompt da janela de comandos do Matlab;
3. clique em “BootDisk”.

Ou ainda, com um disquete formatado em A, digite xpcbootdisk no prompt da janela de comandos do Matlab.

Este disquete deverá ser inserido na unidade A: do computador escravo ainda desligado. Certifique-se de que a opção “floppy disk” está no início da sequência de boot, e só então inicialize o computador escravo. A partir de então o PC escravo já está apto a se comunicar com o PC mestre para troca de informações.

Retornando-se ao computador mestre, deve-se na janela onde está contido o modelo com o qual deseja-se trabalhar em tempo real, dirigir-se ao menu “Simulation”, e em seguida escolher a opção “Simulation parameters”, ou ainda através da tecla de atalho “Ctrl+E”. Será então aberta uma nova janela, na qual é possível ajustar-se os parâmetros do modelo do Simulink.

Selecione a aba “Real-Time Workshop”, e no campo “System target files” selecione “Browse” e escolha “xpctarget.tlc”. Por fim clique em “Build” e aguarde que seu código para execução em tempo real está sendo gerado.

Antes de criar a aplicação tempo-real, é necessário ainda, na aba “Solver”, na região “Solver options”, campo “Type” escolher a opção “Fixed-step”.

Após criada e baixada a aplicação para o PC escravo, basta através dos comandos +tg inicializar a simulação, ou -tg encerrar a simulação.

Uma observação interessante, é que a comunicação entre o computador mestre e o escravo pode ser feita utilizando-se uma conexão do tipo Ethernet com velocidade de até

100Mbit/s, ou ainda utilizando-se a interface serial de ambos os computadores com velocidade de até 115kBaud/s.

A.3.3 Ambiente xPC Target

O ambiente xPC Target define o ambiente de software e hardware de ambos os computadores mestre e escravo. Por isso um bom entendimento do ambiente é essencial à correta configuração do xPC Target.

Neste ambiente existem algumas propriedades que definem a comunicação entre os computadores escravo e mestre, o tipo do compilador e qual sua localização na árvore de diretórios do computador mestre, e o tipo do disco de boot criado durante o processo de ajustes. E algumas funções que permitem a alteração dos valores destas propriedades.

Algumas funções de grande utilidade estão descritas abaixo:

xpcsetup → abre janela de configuração.

xpctest → realiza testes na plataforma xPC Target.

xpctargetping → realiza teste de comunicação entre os computadores escravo e mestre.

A.4 Dmatrix

A.4.1 Introdução

Uma das característica mais importante da linguagem de programação C++, senão a mais importante, é sua capacidade de lidar com a natureza de orientação a objetos. Este estilo de programação apresenta um grande contraste com o procedimento tradicional de programação. Um dos conceitos chaves por trás da programação orientada a objetos é a abstração de dados. A abstração de dados consiste em encapsular em um único bloco (o objeto) ambos os dados e as funções para manipulação dos dados. Um conceito importante em C++ é o de classe. Uma classe é um tipo definido pelo usuário. A classe pode prover

encobrimento de dados, gerenciamento de memória controlado pelo usuário, inicialização de dados garantida, mecanismos para sobrecarregar operadores e outras características bastante úteis.

A idéia de utilizar a característica de orientação a objetos para álgebra de matrizes deve ser provavelmente tão antiga quanto à própria linguagem C++. O uso da abstração de dados e da sobrecarga de operadores permite programar equações de matrizes de forma natural e fácil. Entretanto, uma classe de matriz não está incluída em nenhum compilador comercial de C++. De forma que os programadores devem desenvolver sua própria classe que trabalhe com álgebra de matrizes, ou então obter algum pacote seja ele comercial ou de domínio público.

Apresenta-se então a classe DMatrix desenvolvida para lidar com álgebra de matrizes em ambiente C++. Esta classe é adequada para utilização em controle em tempo real de sistemas que envolvam algoritmos baseados matrizes complicadas, tais como o caso de controle ótimo linear-quadrático, filtro de Kalman estendido, estimação de parâmetros recursivamente e controle preditivo baseado em modelo

A.4.2 Construtores

Um construtor é uma função definida para inicializar (declara) um objeto de uma classe. Vários construtores foram implementados na classe Dmatrix, tal como pode-se ver a seguir.

Tabela 4. Descrição das sintaxes de chamada.

Syntax de chamada	Descrição
DMatrix A	Construtor padrão, cria uma matriz vazia.
DMatrix A(n,m)	Cria uma matriz n x m, aloca memória dinamicamente para o armazenamento dos elementos.
DMatrix A(n,m,a11,a12,...)	Cria uma matriz n x m, aloca memória dinamicamente para o armazenamento dos elementos e determina os valores dos elementos.
DMatrix A(v,dimv,n,m)	Cria uma matriz n x m e usa o vetor v pré-alocado para o armazenamento dos elementos.
DMatrix A(n)	Cria uma matriz n x 1 (vetor coluna), e aloca memória dinamicamente para armazenamento dos elementos.
DMatrix B(A) DMatrix B=A	Construtor de cópia. Cria um novo objeto através da cópia de um objeto DMatrix pré-definido.

A.4.3 Operadores

Os operadores permitem a implementação de diferentes operações entre os objetos, tais como adição e multiplicação.

Tabela 5. Relação entre operador, notação matemática e notação utilizada pelo Dmatrix.

Operador	Notação matemática	Notação DMatrix
= : determinação	$A=B$	$A=B$
* : produto	$C=AB$	$C=A*B$
* : produto escalar-matriz	$C=Fb$	$C=f*B$
* : produto matriz-escalar	$C=Bf$	$C=B*f$
= : produto com substituição	$A=AB$	$A=B$
+ : adição	$C=A+B$	$C=A+B$
+= : adição com substituição	$A=A+B$	$A+=B$
- : subtração	$C=A-B$	$C=A-B$
-= : subtração com substituição	$A=A-B$	$A-=B$
/ : divisão a direita	$C=AB^{-1}$	$C=A/B$
/ : divisão por escalar	$C=A/f$	$C=A/f$
/= : divisão por escalar com substituição	$A=A/f$	$A/=f$
% : divisão a esquerda	$C=A^{-1}B$	$C=A\%B$
() : elemento da matriz	$f = A_{ij}$	$f=A(i,j)$
(),[] : elemento do vetor	$f = x_j$	$f=x(j)=x[j]$
: concatenação horizontal	$C = A \ B $	$C=A B$
&& : concatenação vertical	$C = A$ B	$C=A\&\&B$
> : maior que	$A>f$	$A>f$

\geq : maior ou igual que	$A \geq f$	$A \geq f$
$<$: menor que	$A < f$	$A < f$
\leq : menor ou igual que	$A \leq f$	$A \leq f$
$==$: comparação de igualdade	$A = f ?$	$A = f$

A.4.4 Funções “Public Member”

As funções “membro” podem ser chamadas por um objeto utilizando-se o operador ponto “.” ou através de um ponteiro para objeto usando o operador “->”. Estas funções possuem acesso aos elementos privados da classe. As funções membro implementadas estão descrita na tabela a seguir.

Tabela 6. Descrição das funções “public member”.

Nome	Descrição
GetNoRows()	Adquire número de linhas
GetNoCols()	Adquire número de colunas
GetPr()	Adquire ponteiro do vetor de elementos
element()	Adquire elemento
Print()	Imprime matriz na saída padrão
Fprint()	Imprime matriz no arquivo
Read()	Lê matriz a partir de um arquivo
Save()	Cria arquivo e salva matriz
Load()	Abre arquivo e lê matriz
FillWithZeros()	Completa os elementos da matriz com zeros

assign()	Determina valor ao elemento
Row()	Extrai uma linha da matriz
Column()	Extrai uma coluna da matriz
SetColumn()	Ajusta valores de uma coluna a partir do argumento
SetRow()	Ajusta valores de uma linha a partir do argumento
colMult()	Multiplica coluna por um escalar
rowMult()	Multiplica linha por um escalar
Sub_matrix()	Extrai uma sub-matriz
SetSubMatrix()	Copia uma sub-matriz em uma matriz
SwapColumns()	Troca duas colunas de uma matriz
SwapRows()	Troca duas linhas de uma matriz
Transpose()	Substitui a matriz por sua transposta
Diag()	Especifica a diagonal de uma matriz a partir do argumento
Resize()	Redimensiona um objeto DMatrix
MemCpyArray()	Copia os elementos de um vetor para uma matriz.

A.4.5 Funções “Public Friend”

As funções “amigas” são aquelas que tem acesso aos elementos privados da classe. Em contraste as funções “membro” elas não são chamadas utilizando-se ponto ou operadores de deferência. Uma lista das funções “amigas” implementadas pode ser observada na tabela abaixo.

Tabela 7. Descrição das funções “public friend”.

Função	Descrição
enorm()	Norma euclidiana de uma matriz
InfNorm()	Norma infinita de uma matriz
norm()	Norma de uma matriz
Fnorm()	Norma Frobenious de uma matriz
Max()	Maior valor de uma matriz
MaxAbs()	Maior valor absoluto de uma matriz
Min()	Menor valor de uma matriz
MinAbs()	Menor valor absoluto de uma matriz
abs()	Valor absoluto de uma matriz
sum()	Soma dos elementos de uma coluna
prod()	Produto dos elementos de uma coluna
mean()	Valor médio das colunas de uma matriz
std()	Desvio padrão das colunas de uma matriz
identity(), eye()	Cria matriz identidade de uma dada ordem
zeros(), ones()	Cria uma matriz cheia de zeros ou uns
diag()	Extraí a diagonal de uma matriz
inv()	Inversão de matriz
tra()	Transposta de uma matriz
expm()	Exponencial de uma matriz
eig()	Autovalores da matriz
mpow()	Potência inteira de uma matriz
rcond()	Número de condição recíproca em norma-1 de

	uma matriz
cond()	Número de condição da matriz
rank()	Rank de uma matriz
det()	Determinante da matriz
trace()	Traço da matriz
isSymmetric()	Teste de simetria da matriz.
TProduct()	Produto de matriz com o primeiro operando transposto
ProductT()	Produto de matriz com o segundo operando transposto
TProductT()	Produto de matriz com ambos os operandos transposto
elemProduct()	Produto de uma matriz elemento a elemento
dotProduct()	Produto entre vetores
kronProduct()	Produto Kronecker de duas matrizes
LUSolve()	Solução de equações lineares pela decomposição LU
LU()	Decomposição LU
LUFSolve()	Solução de equações lineares de matriz LU
CholSolve()	Solução de equações lineares pela decomposição Cholesky
Chol()	Decomposição de Cholesky
CholFSolve()	Solução de equações lineares de matriz Cholesky
QRSolve()	Solução de equações lineares pela

	decomposição QR
QR()	Decomposição QR
QRFSolve()	Solução de equações lineares de matriz QR
SVD()	Decomposição do valor singular
pinv()	Pseudoinversa de uma matriz retangular de rank cheio
Orth(), null()	Calcula base ortogonal e nula associada a uma matriz
schur()	Calcula a decomposição schur de uma matriz
SVDSolve()	Solução de um sistema de norma mínima por decomposição do valor singular.
LSMNSolve()	Solução de sistemas retangulares mínimos quadrados ou norma mínima

A.4.6 DMatirx_3d

Uma outra classe também foi implementada, cujo propósito é o de representar matrizes tri-dimensionais, ou variantes no tempo. Devido a esta classe ser derivada da classe DMatrix pode-se utilizar as mesmas funções e propriedades desta última.

Tabela 8. Descrição das funções que pertencem a classe DMatrix_3d

Função	Descrição
DMatrix_3d(n,m,k)	Construtor: cria um objeto DMatrix_3d contendo k matrizes de tamanho n x m.
DMatrix_3d()	Construtor padrão: cria um objeto Dmatrix_3d com dimensão zero.
put_matrix(i,M)	Copia o argumento da matriz para dentro do i-ésimo elemento do referido objeto
operator[i]	Retorna uma referência ao i-ésimo elemento da matriz do referido objeto.
GetNoElemMtx()	Retorna o número de elementos da matriz
GetNoElemCols()	Retorna o número de colunas por elemento
GetNoElemRows()	Retorna o número de linhas por elemento
Resize_3d(n,m,k)	Redimensiona o objeto Dmatrix_3d

A.4.7 Interface com o Matlab

Um conjunto de funções foram escritas para se implementar uma interface com o Matlab, veja tabela abaixo. Duas funções permitem a leitura e armazenamento de objetos DMatrix para e dos arquivos .MAT. Outras duas funções permitem copiar objetos DMatrix para e da estrutura de dados Matriz da linguagem C do Matlab. Porém torna-se necessário à utilização de arquivos “include” e bibliotecas fornecidas com o Matlab.

Tabela 9. Descrição das funções que realizam interface com o Matlab.

Função	Descrição
MatRead()	Lê uma matriz a partir de uma arquivo .MAT
MatSave()	Escreve uma matriz em um arquivo .MAT
DMatrix2Matrix()	Copia um objeto DMatrix para uma estrutura de matriz
Matrix2DMatrix()	Copia uma estrutura de matriz para um objeto DMatrix.

FOLHA DE REGISTRO DO DOCUMENTO			
1. CLASSIFICAÇÃO/TIPO TM	2. DATA 12 de dezembro de 2003	3. DOCUMENTO N° CTA/ITA-IEE/TM-016/2003	4. N° DE PÁGINAS 136
5. TÍTULO E SUBTÍTULO: Controle Preditivo de um Sistema de Leitação Magnética			
6. AUTOR(ES): Renzo Orchiucci Miura			
7. INSTITUIÇÃO(ÕES)/ÓRGÃO(S) INTERNO(S)/DIVISÃO(ÕES): Instituto Tecnológico de Aeronáutica. Divisão de Engenharia Eletrônica – ITA/IEE			
8. PALAVRAS-CHAVE SUGERIDAS PELO AUTOR: Controle Preditivo; Levitação Magnética; xPC Target.			
9. PALAVRAS-CHAVE RESULTANTES DE INDEXAÇÃO: Controle automático; Predição; Algoritmos; Sistemas lineares; Estimação de sistemas; Identificação de parâmetros; Programação matemática; Simulação computadorizada; Controladores; Engenharia de sistemas; Controle.			
10. APRESENTAÇÃO:		X Nacional	Internacional
ITA, São José dos Campos, 2003, 136págs.			
11. RESUMO: A utilização de controladores preditivos já está de certa forma consolidada na indústria de processos, porém é aplicada, geralmente, a plantas cuja dinâmica é lenta e estável em malha aberta. A proposta deste trabalho é implementar um algoritmo que realize o controle preditivo de um sistema não-linear, instável em malha aberta e com dinâmica relativamente rápida, e que possua técnicas para correção de erro em regime, rejeição de perturbações, tratamento de restrições de entrada e saída, e seguimento de referência. Como sistema-exemplo, empregou-se um processo de levitação magnética construído pela Feedback Instruments. Para possibilitar a utilização de técnicas bem estabelecidas de controle preditivo baseadas em modelos lineares, a estratégia adotada usa versões linearizadas do modelo do levitador. A linearização é refeita a cada período de amostragem, de forma a manter o mais próximo possível o comportamento da planta e do modelo. O algoritmo de controle foi executado no computador mestre de uma topologia mestre-escravo implementada com a ferramenta xPC Target do aplicativo Matlab. Os computadores se comunicam através da interface serial RS-232, e é no computador escravo, que é inicializado com o kernel xPC Target, que está instalada uma placa de conversão analógica-digital a qual realiza a interface entre o algoritmo e o sistema físico. As simulações computacionais são resolvidas utilizando-se um algoritmo de programação quadrática, acionado pelo comando “quadprog” do matlab. Os teste experimentais também são resolvidos através de uma lei de controle obtida a partir dos conceitos de controle preditivo, porém não levam em consideração as restrições do sistema físico, ou seja as simulações levam em consideração o tratamento de restrições, porém no caso experimental a lei de controle é obtida a partir do caso irrestrito. Ao longo do trabalho são observados, analisados e discutidos resultados obtidos inicialmente por meio de simulação computacional, em seguida através da utilização de um computador analógico e por fim através da implementação do controle junto ao sistema físico. Em todos os casos foi possível não só estabilizar a saída do sistema, como também fazê-la seguir uma determinada trajetória de referência.			
12. GRAU DE SIGILO: (X) OSTENSIVO () RESERVADO () CONFIDENCIAL () SECRETO			