

CEFET-MG E UFSJ  
DIRETORIA DE PESQUISA E PÓS-GRADUAÇÃO  
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA  
MODELAGEM E CONTROLE DE SISTEMAS - SISTEMAS DE CONTROLE

Fernando Pereira Silva

ESTUDO DE TÉCNICAS PARA SINTONIA DE COMPENSADOR PARA MINIMIZAÇÃO DOS  
EFEITOS DE INCERTEZAS UTILIZANDO INVERSÃO DINÂMICA ROBUSTA.



Belo Horizonte  
2014

Fernando Pereira Silva

ESTUDO DE TÉCNICAS PARA SINTONIA DE COMPENSADOR PARA MINIMIZAÇÃO DOS  
EFEITOS DE INCERTEZAS UTILIZANDO INVERSÃO DINÂMICA ROBUSTA.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre CEFET-MG e UFSJ como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Modelagem e Controle de Sistemas - MCS. Linha de pesquisa: Sistemas de Controle - SC.

Orientador: Prof. Dr. Valter J. S. Leite

Co-orientador: Prof. Dr. Erivelton Geraldo Nepomuceno



Belo Horizonte  
2014

Fernando Pereira Silva

Engenheiro Industrial Eletricista – CEFET-MG

ESTUDO DE TÉCNICAS PARA SINTONIA DE COMPENSADOR PARA MINIMIZAÇÃO DOS  
EFEITOS DE INCERTEZAS UTILIZANDO INVERSÃO DINÂMICA ROBUSTA.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, associação ampla entre CEFET-MG e UFSJ como parte dos requisitos exigidos para a obtenção do título de Mestre em Engenharia Elétrica. Área de concentração: Modelagem e Controle de Sistemas - MCS. Linha de pesquisa: Sistemas de Controle - SC.

Banca Examinadora:

Prof. Dr. Valter Júnior de Souza Leite  
PPGEL/CEFET-MG

Prof. Dr. Eduardo Nunes Gonçalves  
PPGEL/CEFET-MG

Prof. Dr. Erivelton Geraldo Nepomuceno  
PPGEL/UFSJ

Prof. Dr. Márcio Fantini Miranda  
COLTEC/UFMG

Belo Horizonte  
2014

DEDICO ESTE TRABALHO A DEUS E AOS  
MEUS AVÔS, ANTERO CAMILO E GERALDO  
JOSÉ (IN MEMORIAN).

# Agradecimentos

Agradeço,

primeiramente a Jesus Cristo por estar sempre presente em minha vida nessa caminhada, pois toda sabedoria vem do alto [Tg 3:17].

Ao professor Valter J. S. Leite por compartilhar seu conhecimento e experiência, extrema paciência, boa vontade, competência e amizade. Já tive excelentes professores, mas ele me mostrou o real significado da palavra Mestre quando atribuída a um Professor com primazia em ensinar, na essência da palavra. Divido os méritos da realização do trabalho à excelente qualidade da orientação que recebi.

Ao co-orientador Erivelton Geraldo Nepomuceno que com um olhar peculiar contribuiu em pontos importantes dessa versão do trabalho.

a Rosimeire do PPGEL por sua vasta paciência e presteza em resolver os meus problemas.

Aos meus amados pais, Carlos e Fátima por sempre acreditarem em mim, mesmo quando eu achava não ser possível.

A minha amada e querida irmã Luciana, pelo grande apoio, carinho e o ouvido sempre disponível.

A minha amada esposa, Marina, por seu apoio, companherismo, motivação e paciência em me ver passar noites e fins de semanas estudando.

Aos meus queridos amigos que sempre me apoiaram, torceram e oraram por mim nessa caminhada.

Aos meus tios e tias que me viram ausentes algumas vezes nas reuniões de família, mas compreensivos sempre me enviavam sua torcida.

Aos membros da banca examinadora pelos comentários, sugestões e contribuições, que ajudaram a melhorar a qualidade e a redação final do manuscrito.

Ao CEFET-MG pela ótima estrutura que oferece aos estudantes e a grande oportunidade que me propiciou de realizar esse sonho.

À todas as fontes de pesquisa utilizadas nesse trabalho, periódicos, revistas, jornais, Google e a maravilhosa Wikipedia que me permitiram encontrar o conhecimento.

A EMBRAER BH, pelas horas de liberação do trabalho para as reuniões do mestrado.

A todos que de alguma forma contribuíram para que alcançasse esse objetivo pessoal, que DEUS abençoe cada um de vocês imensamente.

“Eu sou o caminho, e a verdade e a vida;  
ninguém vem ao Pai, senão por mim.”  
Jesus Cristo [Jo 14:6]

# Resumo

Neste trabalho é estudada uma proposta de compensação do sinal de controle de sistemas não-lineares linearizados por realimentação. Um problema importante nesse contexto surge quando o sistema não-linear possui diferenças (ou incertezas) em relação ao modelo nominal utilizado para a síntese da lei de linearização por realimentação. Nesses casos, o sistema real em malha fechada pode ficar até mesmo instável. A abordagem conhecida da literatura como Robust Multi Inversion (em inglês), neste trabalho chamada de Inversão Dinâmica Robusta (IDR), é utilizada para se determinar o sinal de compensação de incertezas. As incertezas são assumidas em um domínio politópico. Entretanto, a solução IDR proposta na literatura para a compensação dessas diferenças não possui indicações de como obter o ganho da malha de compensação. Dessa forma as principais contribuições deste trabalho consistem na proposta de dois procedimentos Heurísticos e Meta-Heurísticos para determinação desse ganho. A estratégia Heurística consiste em determinar um valor de  $K_{IDR}$  que reduza a norma do erro entre o comportamento do sistema não-linear controlado com incertezas em relação ao sistema nominal. A estratégia Meta-Heurística consiste também em determinar um valor de  $K_{IDR}$  que reduza a norma do erro, contudo utilizando um algoritmo genético para tal. Outra contribuição é a investigação da aplicação dessa nova topologia de controle em um modelo não-linear de um robô omnidirecional. Esse estudo de caso ilustra que nossa proposta permite a obtenção de um ganho de compensação que leva o sistema em malha fechada a um melhor desempenho mesmo na presença de incertezas.

Palavras-chave: Inversão Dinâmica Robusta. Domínio politópico. Heurística. Meta-Heurística. Controle Não Linear. Linearização Por Realimentação.

# Abstract

In this work, it is studied a proposal of control signal compensation of nonlinear systems under feedback linearization. An important problem in this context appears when the nonlinear system has differences (or uncertainties) with respect to the nonlinear nominal model used to design the feedback linearization control law. In this case, the practical system in closed-loop can become even unstable. The approach known in the literature as Robust Multi Inversion, is applied to determine the uncertainty signal compensation. The uncertainties are assumed in a polytopic domain. However, the IDR solution proposed in the literature to compensate these differences does not have indications of how to get the gain of the compensation loop. Thus the main contributions of this work consists in the proposal of two Heuristic and Meta-Heuristic procedures for determining this gain. The Heuristic strategy consists in determining a value of  $K_{IDR}$  that reduces the error norm between the controlled nonlinear system behavior with uncertainties and the nominal system. The Meta-Heuristic strategy consists also in determining a value of  $K_{IDR}$  that reduces the error norm, however using a genetic algorithm to do so. Another contribution is the investigation of this new control topology in a nonlinear model of an omnidirectional robot. This case study illustrates that our proposal allows to obtain a gain of compensation that leads the closed loop system to better performance even in the presence of uncertainty.

Key-words: Robust Multi Inversion. Polytopic domain. Heuristic. Meta-Heuristics. Nonlinear Control. Feedback linearization.



# Sumário

<b>Lista de Figuras</b>	<b>xiv</b>
<b>Lista de Tabelas</b>	<b>xv</b>
<b>Lista de Acrônimos e Notação</b>	<b>xvii</b>
<b>1 Introdução Geral</b>	<b>1</b>
1.1 Controle de Processos . . . . .	1
1.1.1 Sistemas de controle . . . . .	2
1.1.2 Sistemas com Incertezas . . . . .	7
1.2 Controle Não Linear . . . . .	8
1.2.1 Problemas de Controle Não Linear . . . . .	10
1.3 Futebol de Robôs . . . . .	12
1.4 Motivação e continuidade de pesquisa . . . . .	13
1.5 Proposta e escopo da pesquisa . . . . .	14
1.6 Estrutura da dissertação . . . . .	14
<b>2 Conceitos Fundamentais e Definições</b>	<b>15</b>
2.1 Linearização por Realimentação . . . . .	15
2.1.1 Conceitos Iniciais . . . . .	15
2.1.2 Abordagens de Realimentação Linearizante . . . . .	18
2.1.3 Abordagens de Linearização Clássica . . . . .	24
2.2 Representação de Incertezas . . . . .	30
2.2.1 Incertezas Paramétricas . . . . .	30
2.2.2 Incertezas Negligenciadas ou de Dinâmica Não-modelada . . . . .	30
2.2.3 Incertezas Concentradas . . . . .	30
2.3 Inversão Dinâmica Robusta - IDR . . . . .	31
2.3.1 Introdução . . . . .	31
2.3.2 Principais elementos sobre IDR . . . . .	32
2.3.3 Comentários sobre a abordagem IDR . . . . .	37
2.4 Comentários Gerais . . . . .	38

<b>3</b>	<b>Estratégias para Determinação do Ganho <math>K_{IDR}</math></b>	<b>39</b>
3.1	Heurística . . . . .	39
3.1.1	Características . . . . .	39
3.1.2	O Algoritmo Heurístico . . . . .	40
3.1.3	Análise do Algoritmo Heurístico . . . . .	42
3.2	Meta-Heurística . . . . .	43
3.2.1	Características . . . . .	44
3.2.2	Algoritmo Genético para Determinação do Ganho $K_{IDR}$ . . . . .	44
3.2.3	O Algoritmo Genético . . . . .	45
3.2.4	Análise do Algoritmo Genético . . . . .	45
3.3	Comentários Gerais . . . . .	45
<b>4</b>	<b>Abordagem IDR aplicada ao modelo do robô móvel F180</b>	<b>47</b>
4.1	Introdução . . . . .	47
4.2	Linearização do modelo . . . . .	48
4.3	Topologia de Controle . . . . .	50
4.4	Experimentos Numéricos . . . . .	52
4.4.1	Sistema com Incertezas . . . . .	53
4.4.2	Síntese dos Controladores . . . . .	55
4.4.3	Crterios de desempenho . . . . .	57
4.4.4	Estudos de Caso . . . . .	58
4.4.5	Simulação de Jogadas . . . . .	67
4.4.6	Análise dos Resultados . . . . .	72
<b>5</b>	<b>Conclusões e Perspectivas</b>	<b>75</b>
5.1	Contribuições da Dissertação . . . . .	76
5.2	Trabalhos Futuros . . . . .	76
<b>A</b>	<b>Modelo do Robô Omnidirecional - F-180</b>	<b>77</b>
A.1	Descrição do modelo do robô F180 Omnidirecional . . . . .	77
A.1.1	Modelo Geométrico do Robô . . . . .	77
A.1.2	Matriz de acoplamento das forças . . . . .	78
A.1.3	Modelagem Cinemática . . . . .	80
A.1.4	Modelo Dinâmico de base Móvel . . . . .	81
A.1.5	Modelo dos atuadores . . . . .	82
A.1.6	Modelo Dinâmico da Base Móvel com Inclusão dos Atuadores . . . . .	83
<b>B</b>	<b>Ferramentas</b>	<b>85</b>
B.1	Álgebra de Lie . . . . .	85
B.1.1	Colchete de Lie . . . . .	85
B.1.2	Derivada de Lie . . . . .	85

<b>C</b>	<b>Código Fonte</b>	<b>87</b>
C.1	Algoritmo de busca de máximos e mínimos dos vértices . . . . .	87
C.2	InitMIMO.m . . . . .	90
C.3	Algoritmo Heurístico de sintonia do ganho $K_{IDR}$ . . . . .	92
C.3.1	CalculatePolytopeNLMaxMin . . . . .	92
C.3.2	BuscaGanhoKIDRFinal.m . . . . .	93
C.3.3	runGanhoOtimoIDR . . . . .	97
C.4	Algoritmo Genético de sintonia do ganho $K_{IDR}$ . . . . .	110
C.4.1	BuscaIDRGenetica . . . . .	110
C.4.2	runBuscaIDRGenetica . . . . .	111
C.4.3	CalculatePolytopeNLMaxMin . . . . .	111
<b>D</b>	<b>Resultados Das Simulações</b>	<b>113</b>
D.1	Tabelas de resultados dos casos de estudos . . . . .	113
D.1.1	Resposta para Degrau Unitário . . . . .	113
D.1.2	Trajectoria Reta Sem Rotação . . . . .	114
D.1.3	Trajectoria Reta Com Rotação . . . . .	116
D.1.4	Trajectoria Quadrada Sem Rotação . . . . .	117
D.1.5	Trajectoria Quadrada Com Rotação . . . . .	118
D.1.6	Trajectoria Circular Sem Rotação . . . . .	120
D.1.7	Trajectoria Circular Com Rotação . . . . .	121
D.1.8	Trajectoria em Oito Sem Rotação . . . . .	122
D.1.9	Trajectoria em Oito com Rotação . . . . .	123
	<b>Bibliografia</b>	<b>125</b>



# Lista de Figuras

1.1	Representação do ambiente da liga F180 - Fonte: [Rob11]	13
2.1	Sistema Linearizado.	18
2.2	Planta com múltiplas incertezas.	31
2.3	Linearização por Realimentação.	33
2.4	Linearização por Realimentação com ganho integral.	34
2.5	Abordagem IDR completa aplicada.	35
3.1	Fluxograma do algoritmo heurístico	43
4.1	Projeto de sistemas de controle	51
4.2	Região circular para alocação de polos	51
4.3	Topologia de controle de seguimento com compensação de incertezas	52
4.4	Resposta temporais de $y$ ao degrau unitário.	57
4.5	Resposta trajetória reta aplicada em $r$ para todos os vértices de $\mathcal{A}$	60
4.6	Resposta trajetória quadrada aplicada em $r$ para todos os vértices de $\mathcal{A}$	61
4.7	Resposta trajetória circular aplicada em $r$ para todos os vértices de $\mathcal{A}$	63
4.8	Resposta trajetória em Oito aplicada em $r$ para todos os vértices de $\mathcal{A}$	65
4.9	Resposta $\dot{X}_m$	66
4.10	Resposta $\dot{Y}_m$	66
4.11	Resposta $\dot{\theta}$	67
4.12	Resposta jogada Corner-Centro	68
4.13	Erros da jogada Corner-Centro	68
4.14	Resposta jogada Lateral-Área	69
4.15	Erros da jogada Lateral-Área	69
4.16	Respostas da jogada Centro-Ponta	70
4.17	Erros da jogada Centro-Ponta	70
4.18	Respostas da jogada Condução-Bola	71
4.19	Erros da jogada Condução-Bola	71
A.1	Modelo Geométrico do Robô. Fonte: [Soa11]	78
D.1	Resposta $\dot{\theta}$ para trajetória reta sem rotação	114

D.2	Resposta $\dot{\theta}$ para trajetória reta com rotação . . . . .	116
D.3	Resposta $\dot{\theta}$ para trajetória quadrada com rotação . . . . .	119
D.4	Resposta $\dot{\theta}$ para trajetória circular sem rotação . . . . .	120

# Lista de Tabelas

4.1	Parâmetros nominais do modelo dinâmico . . . . .	53
4.2	Máximos e mínimos dos parâmetros . . . . .	54
4.3	Erro para trajetória reta sem rotação. . . . .	59
4.4	Erro para trajetória reta com rotação. . . . .	59
4.5	Erro para trajetória quadrada sem rotação. . . . .	61
4.6	Erro para trajetória quadrada com rotação. . . . .	62
4.7	Erro para trajetória circular sem rotação. . . . .	63
4.8	Erro para trajetória circular com rotação. . . . .	64
4.9	Erro para trajetória em oito sem rotação. . . . .	66
4.10	Erro para trajetória em oito com rotação. . . . .	66
D.1	Erros dos vértices para entrada em Degrau . . . . .	113
D.2	Erro por Vértice: Trajetória Reta Sem Rotação . . . . .	115
D.3	Erro por Vértice: Trajetória Reta Com Rotação . . . . .	117
D.4	Erro por Vértice: Trajetória Quadrada Sem Rotação . . . . .	118
D.5	Erro por Vértice: Trajetória Quadrada Com Rotação . . . . .	119
D.6	Erro por Vértice: Trajetória Circular Sem Rotação . . . . .	121
D.7	Erro por Vértice: Trajetória Circular Com Rotação . . . . .	122
D.8	Erro por Vértice: Trajetória em Oito Sem Rotação . . . . .	123
D.9	Erro por Vértice: Trajetória em Oito com Rotação . . . . .	124





# Lista de Acrônimos e Notação

## Acrônimos

F180	Categoria da Robocup
GPS	Global Positioning System (Sistema de posicionamento global)
IA	Inteligência Artificial
MIMO	Multiple Input Multiple Output (múltiplas entradas e múltiplas saídas)
PD	Proporcional Derivativo
PI	Proporcional Integral
PID	Proporcional Integral Derivativo
SISO	Single Input Single Output (uma entrada e uma saída)
IDR	Inversão Dinâmica Robusta
RMI	Robust Multi-Inversion
LQR	Linear-quadratic regulator (Regulador linear quadrático)
LTR	Loop Transfer Recover (Recuperação da malha de transferência)
$H_\infty$	H-Infinity
Small Size League	Liga F-180 de futebol de robôs da Robocup

## Notação

$\exists$	existe
$\in$	pertence a
$\notin$	não pertence a
$\subset$	está contido em
$\mathbb{D}$	Domínio de operação
$\mathcal{P}$	Denominação para politopo
$I_p$	matriz identidade de dimensão $p$
$\delta$	especialmente utilizado para representar as incertezas de um sistema
$\Delta$	usado como um conjunto de incertezas
$x_0$	utilizado para denotar o ponto de equilíbrio do sistema
$S_r$	representa o sistema de referência global

$X_r$	coordenada $x$ do robô no sistema de referência global
$Y_r$	coordenada $y$ do robô no sistema de referência global
$S_m$	representa o sistema de referência móvel
$X_m$	coordenada $x$ do robô no sistema de referência móvel
$Y_m$	coordenada $y$ do robô no sistema de referência móvel
$\theta$	coordenada de posição angular do robô
$\phi_t$	ângulo entre o eixo da roda $t$ e o eixo $X_m$
$\mathcal{L}$	utilizado para denotar a derivada de Lie

# Introdução Geral

Sistemas de controle estão presentes em toda estrutura da sociedade. Eles aparecem em nossas casas, carros, indústria, sistemas de comunicação e transporte, são alguns dos exemplos [Ast02]. O controle de sistemas representa um papel vital no avanço da engenharia e ciência, por isso, é cada vez mais utilizado em atividades de caráter crítico. Garantindo que processos não irão parar de funcionar ou falhar de forma inesperada, melhorando taxas de produção, melhoria da qualidade, diminuição dos custos de produção e redução de operações manuais [Ast02][Oga85]. A pesquisa e projeto de equipamentos experimentais e instrumentação tem se valido largamente da utilização de controle de sistemas, otimizando o tempo de desenvolvimento de tecnologias que irão ser utilizadas no nosso futuro, isso em diversos campos de pesquisa: economia, biologia, medicina e outros. Essas vantagens fazem com que haja grande interesse da comunidade científica e do mercado em realizar pesquisas na área, visando otimizar sistemas em relação ao consumo de energia, impacto ambiental, segurança das pessoas, custo etc. Aumentando os benefícios para a sociedade moderna [Ast02] [Nis01].

Controle, como vários outros ramos da ciência de engenharia, tem sido desenvolvido no mesmo padrão que a ciência natural. Contudo a ciência natural visa entender um fenômeno da natureza, ou seja, definir uma lei que descreve a natureza. A ciência em engenharia visa entender um fenômeno, inventar e projetar um sistema artificial para interagir com esse fenômeno. A interação é a chave da praticidade de todos os sistemas projetados por homens. Atualmente um dos objetivos da ciência em engenharia é encontrar abordagens para compreender e projetar sistemas artificiais capazes de interagir com outros sistemas extremamente complexos naturais ou artificiais, com eficácia e eficiência [Ast02]. Dentro desse contexto surgem diversas abordagens para controle de processos.

## 1.1 Controle de Processos

As indústrias usam continuamente processos de fabricação, assim, o controle desses processos é essencial para essas indústrias em que os processos de produção não pode ser executados sem os sistemas de controle. Os pontos chaves em controle de processos são [Ast02]:

- Manter variáveis essenciais de qualidade a valores especificados.

- Minimizar o uso de energia e de matéria-prima.
- Realizar rapidamente mudanças na produção, ou torná-las mais ágeis.

Controlar um processo para se atingir esses fatores-chave, nada mais é do que manipular o comportamento do sistema presente no processo, de forma a obter-se as saídas desejadas. Para isso é importante que se conheça bem as características do sistema. Sistema é um conjunto de componentes que atuam juntos com intuito de realizar uma determinada função. O estado de um sistema é o conjunto de variáveis necessárias e suficientes que determinam a cada instante a condição atual do sistema como um todo, ou seja, determinam a dinâmica do sistema. Assim sistemas dinâmicos são sistemas cujas variáveis de estado variam no tempo, segundo leis físicas que podem ser modeladas. Tal modelagem determina o modelo do sistema, que são representações que permitem estabelecer relações entre causa e efeito do sistema dinâmico. Os modelos podem ser físicos, em que são a representação simplificada do modelo real, mantendo as características mais importantes, ou podem ser matemáticos, que procuram representar o comportamento dinâmico por meio de equações matemáticas (equações de derivadas, equações diferenciais). O modelo serve como base para se projetar o controlador do sistema, ou seja, quanto mais preciso for o modelo do sistema real, melhor será o desempenho do controlador projetado [Car12]. Contudo a modelagem só ocorre em sistemas determinísticos, não sendo possível em sistemas estocásticos. Os sistemas determinísticos são divididos em sistemas lineares e não-lineares de acordo com a descrição matemática da sua dinâmica. Sendo os lineares descritos por equações lineares e geralmente variam lentamente no tempo, possuem coeficientes de dinâmica lenta. Ao passo que sistemas não-lineares possuem dinâmica rápida e são descritos por equações não-lineares. Sistemas lineares são amplamente conhecidos na teoria de controle linear, e atualmente modelar e controlar processos de sistemas lineares é algo realizado com determinada facilidade. Entretanto a grande maioria dos sistemas físicos são de natureza não-linear. Isto não significa que a teoria de controle de sistemas lineares não possa ser aplicada. Nesses casos, para tal aplicação, deve-se realizar uma linearização (se possível) do sistema não-linear a fim de tornar o controle menos suscetível às não linearidades. Assim, para se controlar o processo de um sistema linear ou não, é necessário que seja projetado um sistema de controle [Oga85] [Car12].

### 1.1.1 Sistemas de controle

Sistemas de controle são considerados na teoria de controle que trata do comportamento de sistemas dinâmicos e é baseada no princípio da realimentação (ou retroação) e objetiva o controle de determinadas variáveis de um sistema [Oga85]. Controlar um sistema nada mais é do que conduzir a saída real do sistema a ser a saída desejada, referência, ou seja, quando uma ou mais variáveis de saída necessitam seguir uma referência ao longo do tempo, um controlador manipula as entradas do sistema para obter o efeito desejado nas saídas deste sistema [Ast02] [Car12]. As técnicas de controle para se projetar um controlador de sistemas, podem ser divididas em dois grandes grupos [Wik13][Car12]. Em que o primeiro grupo é denominado de controle clássico e utiliza os seguintes controladores:

- Controle Liga/Desliga (On/Off em inglês).

- Controle auto-operados.
- Controle Proporcional.
- Controle Derivativo.
- Controle Integral.
- Controle Proporcional Integral Derivativo (PI, PD ou PID).
- Avanço e/ou atraso de fase.

O segundo grupo é chamado de controle moderno e possui os seguintes controladores:

- Controle Multivariável.
- Controle Adaptativo.
- Controle Ótimo.
- Controle Não-Linear.
- Controle linear à parâmetros variantes.
- Controle Robusto.
- Controle Inteligente.

A descrição de algumas dessas técnicas é apresentado a seguir bem como suas limitações, as demais serão abordadas nas demais seções.

### **Controle Liga/Desliga**

O controle Liga/Desliga é caracterizado por possuir somente dois estados de atuação, ligado ou desligado. Um exemplo deste tipo de controle é de nível de uma caixa d'água, o controle fica ligado/ativo enquanto está enchendo a caixa, quando atinge determinado nível a boia desarma a chave e o controle fica desligado. Quando o nível de água baixa, a boia arma a chave e o controle é ligado. Observe que nesse formato próximo ao ponto de operação do sistema de controle qualquer pequena variação no nível da caixa, positiva ou negativa faz que com o sistema de controle fique sendo ativado e desativado em um intervalo curtos de tempo. Isso gera desgaste precoce no atuador, e conquentemente o danifica. Uma solução para esse problema é a inclusão de uma zona morta no ponto de operação, fazendo com que o controle fique desligado sempre que o estado estiver próximo (e não apenas igual) do ponto de operação. Contudo nesse tipo de técnica a resposta fica oscilando entre os valores mínimo e máximo da zona morta e entre os extremos o sistema segue a sua própria dinâmica não-linear, uma vez que não há atuação dentro da zona morta [Car12].

## Controle Auto-Operados

Em controladores auto-operados a energia necessária para a atuação está presente no próprio sistema controlado. Nem sempre é possível projetar um controlador deste tipo para a grande parte dos sistemas físicos existentes, um exemplo deste tipo de controlador é um regulador de pressão. Nesse caso a válvula acionada por um diafragma ajusta a pressão de saída com base na pressão desejada/refrência e no ajuste de operação [Car12].

## Controle Proporcional Integral Derivativo - PID

O controlador é composto por três tipos de ação de controle: Controle Proporcional (P), Controle Integral (I) e o Controle Derivativo (D), que podem ser utilizados de forma independente ou de forma combinada, PI, PD ou PID. A ação combinada, PID, é geralmente a mais eficaz na solução dos problemas de controle devido a características distintas de cada termo do controlador. O primeiro termo P, a ação de controle é proporcional ao sinal do erro (diferença entre a saída desejada e a saída real), ou seja, quanto maior o erro maior será a atuação. O termo proporcional é uma resposta rápida do sistema de intensidade dada por um ganho  $K_p$  ao erro do sistema. O uso de um controlador exclusivamente proporcional só é indicado em sistemas com a constante de amortecimento não nula, do contrário irão oscilar indefinidamente.

O segundo termo D é relativo à ação de controle derivativo e é proporcional à variação do erro. Dessa forma quanto maior for a taxa de variação do erro, maior será a ação derivativa. O termo derivativo quando utilizado deve ser associado ao termo proporcional, assim o tempo derivativo é dado pela relação  $T_d = \frac{K_p}{K_d}$ , em que  $K_d$  é o ganho derivativo do controlador. Observe que um controlador PD, devido as características dos termos, pode não ser capaz de eliminar o erro residual quando o sistema encontra-se em regime permanente. Para tal, é necessário que seja incluso ao controlador o termo de integração, I.

A ação de um controlador integral muda de forma proporcional a integral do sinal de erro. O termo I pode ser utilizado em associação ao termo P, formando o controlador PI. Contudo o controlador PID é o mais abrangente e uma das vantagens desse tipo de técnica é ele controla até sistemas naturalmente instáveis, como o exemplo clássico do pêndulo invertido. Controladores PID são utilizados em sistemas sujeitos a perturbações, e não naturalmente amortecidos ou com amortecimento insuficiente [Car12].

## Avanço e/ou Atraso de Fase

O controlador em avanço tem como finalidade suprir o atraso de fase estabelecido naturalmente pelas características do sistema. Esse tipo de compensação remodela o lugar das raízes de maneira a se obter pólos dominantes em malha fechada, aumentando o amortecimento com menor tempo de subida e de acomodação. Além disso as margens de ganho e fase são melhoradas, contudo o erro do estado estacionário não é afetado. O controlador em atraso melhora o erro em regime permanente, no entanto, reduz a largura da faixa, o que implica em uma resposta mais lenta, com maior tempo de subida e acomodação. O controlador Avanço-Atraso une esses dois controladores, e é utilizado em casos em que deseja-se uma resposta rápida, porém com diminuição do erro em regime permanente [Ara07].

## Controle Multivariável

Grande parte das técnicas de controle são projetadas para tratar sistemas simplificados de única entrada e única saída, contudo em aplicações práticas de controle os sistemas possuem mais de uma entrada e mais de uma saída. Caracterizando-se por terem mais de uma variável a serem controladas simultaneamente, que respondem aos estímulos de distintas entradas. Esses sistemas são denominados por MIMO (sigla em inglês para Multiple Inputs Multiple Outputs), e o controle dessas variáveis é chamado de controle multivariável.

Nesse tipo de controle pode ser utilizado a Matriz de Ganhos Relativos (RGA) desenvolvida por Bristol em 1966. Os elementos da matriz RGA são definidos como a razão entre o ganho de malha aberta e de malha fechada considerando o sistema em estado estacionário. Ela fornece dois tipos de informações úteis: medida das interações de processos e a recomendação sobre o melhor emparelhamento das variáveis controladas e manipuladas [Sko01].

## Controle Adaptativo

Controle adaptativo é a ação de controle de sistemas capazes de modificar seus próprios parâmetros em resposta a alterações verificadas em algum módulo que esteja sendo observado. O termo adaptativo significa alterar o comportamento em função a novas circunstâncias de operação, visando manter um nível esperado de desempenho. O controle adaptativo é uma extensão natural da teoria de controle clássica, ou seja, de sistemas realimentados clássicos. Buscando projetar controladores dotados de maior grau de autonomia, a teoria de controle adaptativo foi muito atuante nos anos 50 no desenvolvimento de controladores de voo em aeronaves supersônicas, já que ganhos constantes não eram suficientes para desempenho de operação na região supersônica. Em controladores adaptativos são necessárias duas malhas de realimentação: a malha de controle convencional e a malha de adaptação, que monitora o desempenho e ajusta os parâmetros do controlador em função das condições de operação. Existem dois tipos de controle adaptativo [OR95]:

- Controle Adaptativo por Modelo de Referência: Trata-se de um método direto, pois as regras de ajuste indicam como os parâmetros do controlador devem ser ajustados.
- Reguladores Auto-Ajustáveis: Método indireto em os parâmetros do modelo do sistema são ajustados e os parâmetros do controlador são obtidos por procedimentos de projeto.

Ainda nos dias atuais o controle adaptativo é largamente utilizado em controladores de voo. Um grande avanço na utilização dessa técnica foi o crescimento no desempenho de recursos computacionais.

## Controle Ótimo

Controle Ótimo lida com o problema de encontrar uma lei de controle para um dado sistema tal que um certo critério de otimização seja atingido. Um problema de controle inclui uma função custo que é uma função das variáveis de estado e variáveis de controle. O controle ótimo é um conjunto de equações diferenciais descrevendo os caminhos das variáveis de controle que minimizam a função custo. O controle ótimo pode ser obtido utilizando o princípio de

Pontryagin, ou resolvendo a equação de Hamilton Jacobi Bellman (condição suficiente) [ZW98]. Um exemplo de um problema de controle ótimo é encontrar o percurso para a trajetória de um carro que minimize o consumo de combustível, e que o percurso seja completado em um tempo máximo. O problema consiste em minimizar o custo financeiro total para completar a viagem, assumindo valores monetários para o tempo e combustível.

O controle ótimo atual é baseado no trabalho de Wiener de 1940, contudo só adquiriu uma forma madura de teoria em 1960 e passou a ser chamado de *Linear Quadratic Gaussian* ou controle LQG. Seu desenvolvimento ocorreu simultaneamente a grandes programas de pesquisa e recebeu financiamento considerável dos governos norte americano e da antiga União Soviética em questões aeroespaciais. Esses problemas eram referentes a manobra de foguetes com redução do consumo de combustível, nesse tipo de aplicação os engenheiros aeroespaciais foram bem sucedidos. Contudo o emprego da técnica em problemas cotidianos por engenheiros de controle em outras áreas, não obtiveram o mesmo êxito. Pois modelos precisos de plantas nem sempre eram possíveis e a suposição de perturbações com ruído branco nem sempre é relevante [Sko01].

Um problema do controle tradicional LQG é quando assume-se que a dinâmica da planta é linear e conhecida, e que a medição do ruído e sinais de distúrbio são estocásticos com propriedades estatísticas. Para o modelo da planta,

$$\dot{x} = Ax + Bu + \omega_d \quad (1.1)$$

$$y = Cx + Du + \omega_n \quad (1.2)$$

em que por simplicidade adota-se  $D = 0$ .  $\omega_d$  e  $\omega_n$  são os distúrbios e ruídos de medição respectivamente, que são usualmente assumidos como processos estocásticos Gaussianos de média zero, não correlacionados com matrizes com densidade espectral de potência constante. Para a solução desse problema existe a proposta do Teorema de separação [Sko01]. Uma outra limitação para um sistema com controlador LQG em que há uma combinação de filtro de Kalman e uma lei de controle LQR, é que não há margens de estabilidade garantida. Essa afirmação foi duramente apresentada por Doyle em 1978 à comunidade de controle. Ele mostrou que existe combinações arbitrárias LQG com pequenas margens de ganhos [Sko01].

## Controle linear a parâmetros variantes (Síntese LPV)

Um sistema linear pode ser considerado variante no tempo ou não-estacionário quando um ou mais parâmetros do seu modelo variam amplamente com o tempo e não podem ser considerados como parâmetros incertos. Muitas vezes, os sistemas não-lineares podem ser tratados como sistemas lineares não-estacionários. Um sistema LPV pode ser descrito através das equações a seguir:

$$\dot{x} = A(\theta(t))x(t) + B(\theta(t))u(t) \quad (1.3)$$

$$y(t) = C(\theta(t))x(t) + D(\theta(t))u(t) \quad (1.4)$$

Em que as matrizes A, B, C e D são dependentes do parâmetro continuamente variante no tempo  $\theta(t)$ . Quando  $\theta(t)$  possui elementos cuja variação depende, além do tempo, de um ou mais estados do sistema, configura-se uma não-linearidade no sistema, fazendo com que o termo *quasi*-LPV seja mais adequado, ou seja, apesar do sistema ser não-linear, considera-se que o parâmetro  $\theta(t)$  evolui num domínio específico sem obedecer à dinâmica de  $x(t)$  [Sko01].



## Comentários

Diversas técnicas da teoria de controle clássica e moderna foram citadas. Note que técnicas de aplicação a sistemas lineares quando aplicadas a sistemas não lineares exigem uma linearização do sistema não linear. Isso nem sempre é possível, inviabilizando o controle do sistema. Em outros casos quando a linearização é possível, ela limita o controlador a um ponto de operação, ou seja, o ponto onde o sistema foi linearizado e sua vizinhança. Sendo que fora deste ponto não há garantias que o controlador atenda os critérios de desempenho ou que o sistema seja estável. Isso é problema sério, pois limita a dinâmica de operação do sistema a somente um dado ponto, algumas das técnicas apresentadas como o controle adaptativo tenta amenizar esse problema com a mudança de comportamento de atuação de acordo com o ponto de operação do sistema, ou seja, pode-se linearizar vários pontos do sistema não linear, criando uma gama de pontos de operação em que o comportamento da dinâmica são conhecidos e garantido o desempenho. Contudo em sistemas variantes no tempo, não há garantia que na transição de um ponto de operação para outro seja mantida a estabilidade, ou que haja garantia de estabilidade nas regiões entre os pontos.

Dessa forma nota-se que as técnicas de controle de sistemas lineares são altamente dependentes do modelo para sintonia do controlador. Diversos trabalhos foram desenvolvidos com essas técnicas, podemos citar [Ass73] em que uma condição necessária e suficiente para desacoplar um sistema não linear com realimentação de estados é obtida. Mostrando que quando a condição é satisfeita existe uma lei de controle capaz de controlar cada variável de saída associada a uma entrada. Ou ainda o trabalho apresentado em [HSM92] que apresenta uma linearização entrada-saída para um sistema não linear utilizando realimentação de estado. Outros exemplos são demonstrados em [MPT95], [MC89] e [Pop61]. Nesses trabalhos a síntese dos controladores é feita baseada no modelo que representa o sistema real. Contudo, diferenças entre o modelo e o sistema real a ser controlado, o projeto implementado pode frustrar a expectativa de desempenho e até de estabilidade. Dentro desse contexto surge a necessidade de técnicas de controle capazes de lidarem com a presença de incertezas e controlar os sistemas não lineares.

### 1.1.2 Sistemas com Incertezas

Para garantir critérios de desempenho e estabilidade é importante que o sistema real tenha comportamento extremamente próximo do modelo para o qual o controlador foi projetado. Isso reflete um problema já que sistemas reais possuem incertezas associadas que diferem naturalmente o sistema real do sistema modelado. Fazendo com que o desempenho para a síntese de projeto não seja alcançado, ou pior, que as divergências entre modelo e modelo real sejam suficientes para que o modelo linearizado divirja consideravelmente do ponto linearizado e de sua vizinha, levando o controlador linear a operar em uma região não linear. A relação entre os modelos e a realidade que eles representam é sutil e complexa. Um modelo matemático provém um mapa das entradas para as saídas, e a qualidade do modelo depende de quão preciso ele é em relação à planta real. Contudo é sabido que um modelo nunca será idêntico ao sistema real, já que uma representação física fiel da planta pode nunca ser alcançada ou construída, assim um conjunto de modelos matemáticos aproximados tendem a facilitar o trabalho do engenheiro.

As técnicas de modelagem devem levar em conta as inevitáveis inadequações do modelo frente à planta. Um bom modelo deve ser simples o suficiente para facilitar o projeto, e ainda sim complexo o suficiente para dar ao engenheiro confiabilidade que o projeto baseado no modelo irá funcionar na planta real [ZW98] [Sko01].

O termo incerteza é referente às diferenças ou erros entre o modelo e a realidade, e os mecanismos utilizados para expressar esses erros são chamados de representação de incertezas. Representação de incertezas variam primariamente em termos da quantidade de estrutura nelas contidas. As incertezas podem ser devidas à variação dos parâmetros da planta e/ou de dinâmicas não-modeladas. No sistema real isso representa parâmetros que variam no tempo, ou dinâmica do sistema omitida na simplificação do modelo. Essas incertezas podem afetar o desempenho nominal do sistema, bem como a estabilidade nominal, quando o controlador é levado a operar em regiões para as quais não foi projetado [Sko01]. No entanto imagine um controlador capaz de garantir os critérios de desempenho para um sistema em uma região  $\mathbb{L}$ , em que essa região é formada pelas incertezas associadas, e que o mesmo garanta estabilidade do sistema nessa região. Poderíamos dizer que trata-se de um controlador robusto [ZW98].

O controle robusto é um braço da teoria de controle que explicitamente lida com incertezas na abordagem de síntese de controladores. Os métodos de controle robusto são projetados para funcionar corretamente na presença de incertezas paramétricas e distúrbios em um mesmo conjunto (normalmente compacto). Métodos robustos visam alcançar desempenho e ou estabilidade robusta na presença de erros de modelagem limitados. O critério de estabilidade robusta preceitua que projete-se um controlador para a planta nominal,  $G_0$ , que resulte em um sistema estável para qualquer descrição de fator de incerteza. Entretanto para aplicação dos critérios de robustez é necessário que haja uma representação matemática das incertezas associadas [ZW98]. Existem diversas formas de representação de incertezas, dentre algumas são: incertezas na forma afim, incertezas limitadas em norma, incertezas intervalares, incertezas na forma linear fracionária e incertezas politópicas. Neste trabalho é seguida a representação politópica por ser mais geral e possuir um grande potencial para o desenvolvimento de métodos ou procedimentos para a síntese de compensadores. Um politopo é o fecho convexo de um conjunto finito de pontos  $\mathcal{P}$  = feixe convexo  $\{p_1, p_2, p_3, \dots, p_N\}$ . Um vértice é um ponto de  $\mathcal{P}$  que não pode ser gerado como a combinação convexa de dois outros pontos distintos de  $\mathcal{P}$ . Entretanto qualquer ponto interno de  $\mathcal{P}$ , pode ser gerado por meio da combinação convexa dos vértices de  $\mathcal{P}$ .

$$\mathcal{P} = \sum_{i=1}^N \lambda_i p_i, \quad \sum_{i=1}^N \lambda_i = 1, \quad \lambda_i \geq 0. \quad (1.5)$$

O conjunto de restrições para  $\lambda_i$  é chamado de unidade simples.

Construídos os conceitos de incertezas e de sua representação via politópica, é necessário levantar conceitos de controle não linear.

## 1.2 Controle Não Linear

As técnicas de controle linear já estão bem evoluídas e possuem uma enorme variedade de poderosos métodos desenvolvidos ao longo dos anos com diversos casos de sucesso em aplicações industriais, como apresentado na seção 1.1.1. Contudo, várias áreas da engenharia como

aeroespacial, robótica, controle de processos e engenharia biomédica, realiza diversas pesquisas no desenvolvimento e aplicações de técnicas de controle não linear. Algumas das razões para interesse na área de controle linear são [Isi85]:

- métodos de controle linear operam em uma região muito pequena para que o modelo linearizado seja válido. Quando a região de operação deve ser maior, os controladores lineares apresentam desempenho muito ruim, ou são instáveis, já que as não linearidades do sistema não podem ser compensadas. O mesmo não ocorre com controladores não lineares, já que operam em não linearidades em uma larga região. Isso é facilmente demonstrado em problemas de controle de movimento de robôs. Quando um controlador linear é usado para controlar o movimento de um robô, ele negligencia as forças não lineares associadas com o movimento. A precisão do controlador degrada rapidamente pontos do movimento como velocidade etc. utilizando um controlador não linear, esse pode compensar integralmente as forças não lineares no movimento do robô e lidar com a alta precisão necessária para atender a larga região de operação e velocidade do robô [Isi85].
- Uma outra premissa do controle linear é que o modelo do sistema seja linearizado. Todavia, como já explicitado, em sistemas de controle possuem diversas não linearidades por vezes descontínuas que nem sempre permitem uma aproximação linear. Essas não linearidades são chamadas de "hard nonlinearities", como saturação, zonas morta e histerese. Esses efeitos não podem ser tratados com métodos lineares, e requerem o uso de técnicas não lineares[Isi85].
- No projeto de controladores lineares é usualmente necessário assumir que os parâmetros do sistema modelado são razoavelmente bem conhecidos. Todavia, muitos dos problemas de controle envolvem incertezas nos parâmetros do modelo. Além do fator que os parâmetros de um sistema podem variar lentamente com o tempo (ar, pressão, desgaste, temperatura), ou variar abruptamente (combustível de um foguete, liberação de carga em um avião). Controladores lineares quando expostos a essas variações apresentam desempenho prejudicado ou instabilidade. Duas classes de controladores já apresentadas, são técnicas para esse tipo de situação: Controladores robustos e controladores adaptativos [Isi85].
- Na concepção de um modelo foi citada que ela deve ser simples o suficiente para facilitar o projeto, e complexo o suficiente para ter confiabilidade. Contudo seguir essa filosofia nem sempre é simples, por vezes os modelos são simplificados a tal ponto de negligenciar informações importantes do sistema, como a dinâmica de suas não linearidades. Assim o modelo resultante possui dinâmicas não modeladas, que serão incertezas ao controlador projetado frente ao sistema real. Isso causa impacto extremamente negativos em controladores lineares, mas em controladores não lineares esse fator pode ser contornado [Isi85].

Existem diversas razões para o uso de técnicas de controle não linear, tal como custo e otimização de desempenho. Nas plantas industriais existem diversas máquinas com consideráveis não linearidades, utilizando controladores lineares. Esses controladores lineares requerem atuadores extremamente precisos e sensores de alta qualidade para garantir o comportamento esperado na operação. Enquanto que controladores não lineares permitem o uso de elementos

mais simples e de menor custo. Dessa forma, o controle não linear é uma área importante do controle automático. O uso de técnicas básicas de controle não linear em análise e projeto de controladores podem simplificar consideravelmente o controle de processos, trazendo melhoras significativas em desempenho e estabilidade, além de redução de custos e outros fatores interligados ao processo. No passado, a aplicação de métodos de controle não linear eram limitados por dificuldades computacionais associados na fase de projeto e análise do controle não linear. Nos últimos anos, os avanços computacionais tem resolvido esse problema. O tópico de projeto de controle não linear para uma ampla região de atuação tem atraído muita atenção, porque o advento de poderosos microprocessadores tem possibilitado a implementação desses controladores atendendo à crescente necessidade de se projetar sistemas de controles aos novos sistemas robóticos, aeronaves de alto desempenho, etc. Contudo ainda existem poucos trabalhos significativos nessa área, que começa a ser explorada de forma efetiva no presente atual [Isi85].

### 1.2.1 Problemas de Controle Não Linear

Se as tarefas do sistema de controle envolve uma ampla região de operação e/ou movimentos de alta velocidade, efeitos da não linearidade serão significativos na dinâmica e o controle não linear deve ser suficiente para se atingir o desempenho desejado. De uma maneira geral, as tarefas de um sistema de controle possuem dois objetivos: estabilização e seguir a referência. Nos problemas de estabilização, o controlador ou estabilizador é projetado de forma que o estado do sistema em malha-fechada seja estabilizado em torno de um ponto de equilíbrio. Nos problemas de seguir a referência ou rastreamento (*tracking* em inglês), o objetivo do projeto é construir um controlador que a saída do sistema siga uma trajetória variante no tempo [Isi85]. Problemas como esses são abordados em trabalhos como em [PG09], em que um controlador híbrido é proposto para garantir estabilidade à um pêndulo mola invertido com carga, e [CLPV03] em que é proposto um controlador não linear de realimentação para um conjunto de sistemas lineares com atuadores não lineares.

Normalmente, problemas de *tracking* são mais difíceis de serem resolvidos do que os de estabilização porque no primeiro o controlador não deve somente manter o estado estabilizado mas também guiar a saída do sistema para a referência. Todavia, problemas de estabilização podem ser tratados como um caso especial de problemas de “tracking”, com a trajetória desejada constante. Um dos alicerces da teoria de controle é especificar o comportamento desejado do sistema. Em controle linear, o comportamento desejado de um sistema de controle pode ser especificado sistematicamente, no domínio do tempo ou no domínio da frequência. No projeto de controle linear, o primeiro passo é quantificar as especificações do sistema de controle em malha-fechada, e então sintetizar o controlador para atender essas especificações. Todavia, especificações sistemáticas para sistemas não lineares são muito menos óbvias, porque a resposta de sistema não linear a um comando não reflete sua resposta a um outro comando. Dessa forma as especificações de comportamento de um sistema de controle não linear deve atender aos seguintes interesses [Isi85]:

- Estabilidade, deve ser garantida para o modelo nominal localmente e globalmente. A região de estabilidade e convergência também são de interesse.

- Precisão e velocidade de resposta, deve ser considerada para alguma trajetória de movimento típica na região de operação.
- Robustez, é a sensibilidade para efeitos que não são considerados no projeto, tais como distúrbios, medição ruidosa, dinâmica não modelada etc.
- Custo, é determinado em um sistema de controle principalmente pelo número e tipo dos atuadores, sensores, e computadores necessários para implementá-lo.

Note que estabilidade não implica em uma habilidade do sistema em resistir a distúrbios persistentes, mesmo de baixa amplitude. A estabilidade de um sistema não linear é definida em relação às condições iniciais e somente distúrbios temporários devem ser traduzidos como condição inicial. A habilidade em “resistir” a distúrbios persistentes em sistemas de controle não linear é relativo ao conceito de robustez. Outro fator fundamental é que um bom controlador só pode ser obtido baseando-se nas seguintes relações: estabilidade/robustez, estabilidade/desempenho, custo/desempenho e assim por diante. Não existem muitas abordagens para tratar a questão de robustez em controle de sistemas não lineares e nenhum método geral para se projetar controladores não lineares. O que existe é uma coleção de técnicas alternativas e complementares, cada qual com uma melhor aplicação em uma classe particular de problemas de controle não linear [Isi85]. Algumas dessas técnicas são apresentadas a seguir, além do controle adaptativo já mencionado.

- Tentativa e erro, a ideia consiste em usar ferramentas de análise para guiar a busca por um controlador que possa então ser justificada por análise e simulação. O método de plano de fase, o método de função descritiva, e análise de Lyapunov podem ser utilizados para este propósito. O processo é muito dependente de experiência e intuição, muitas vezes devido a isso em sistemas complexos o método falha.
- Ganho escalonável é uma abordagem de como aplicar a bem desenvolvida metodologia de controle linear para o controle de sistemas não lineares. A ideia do ganho escalonável é em selecionar um número de pontos de operação que cobrem a gama de operação do sistema. Então, em cada um desses pontos é feita uma aproximação linear variante no tempo para a dinâmica da planta e projeta-se um controlador linear para cada planta linearizada. Entre os pontos de operação do sistema, os parâmetros dos compensadores são interpolados, ou escalonados, resultando em um compensador global. Embora seja conceitualmente simples, e com inúmeros casos de sucesso, o problema do ganho escalonável é sua limitada garantia de estabilidade na operação não linear.
- Controle Robusto é uma abordagem em que o controlador é projetado baseado em dois pontos, no modelo nominal e em alguma caracterização do modelo de incertezas. As técnicas de controle não linear robusto tem provido soluções a uma efetiva variedade de problemas de controle não linear.
- Linearização por Realimentação, essa abordagem lida com técnicas de transformação do modelo do sistema original em modelos equivalentes simplificados. A linearização por

realimentação pode ser utilizada como uma metodologia de projeto não linear. A ideia básica consiste em transformar um sistema não linear (completamente ou parcialmente) em um sistema linear, e então usar uma técnica conhecida e poderosa de projeto linear para completar o projeto de controle. A técnica é aplicável a uma vasta gama de problemas de controle não linear, contudo a robustez não é garantida na presença de incertezas paramétricas ou distúrbios.

A abordagem de Linearização por Realimentação, técnica utilizada neste trabalho a um modelo não linear proposto, é discutida com mais detalhes nos capítulos subsequentes. Um exemplo de modelo não linear são os robôs utilizados em futebol de robôs, são modelos não lineares de dinâmica rápida em que incertezas paramétricas são um fator comum. Fator esse que dificulta consideravelmente o projeto de controle.

### 1.3 Futebol de Robôs

Dentro do contexto há uma competição denominada *Robocup*. Trata-se de uma competição de nível mundial que ocorre anualmente, visando o estudo e desenvolvimento da Inteligência Artificial (IA) e da Robótica. Fornecendo desafios e problemas em que várias tecnologias e metodologias podem ser combinadas com intuito de se obter melhores resultados. Cada evento da *Robocup* é constituído de duas partes: as competições e o simpósio. As competições são divididas em três grupos: *RobocupSoccer*, *RobocupRescue* e *RobocupJunior*, sendo cada uma composta por várias ligas. No simpósio realizado após as competições, são apresentados e discutidos trabalhos científicos da área. As equipes são condicionadas a apresentar as soluções desenvolvidas e disponibilizar o seu trabalho, para que possam participar das competições. Assim garantindo o incentivo e contínua evolução das equipes, bem como o surgimento de novas equipes com alto padrão de desempenho.

A primeira edição ocorreu em 1997 em Nagoya no Japão, logo após a organização de uma pré-Robocup em Osaka (Japão) para identificar possíveis problemas e dificuldades na organização deste tipo de eventos a uma escala global. A iniciativa foi lançada pelo Dr. Hiroaki Kitano (Tóquio), um pesquisador da área de Inteligência Artificial que tornou-se presidente e fundador da *Robocup Federation*, com o objetivo de dinamizar a evolução da IA, em particular dos Sistemas Autônomos Multi-Agente. Dentro da *RobocupSoccer* existem 5 ligas esportivas: *Humanoid*, *Middle Size*, *Simulation*, *Small Size* e *Standart Plataform*, cada uma com suas características e peculiaridades [Rob11]. Para o contexto deste trabalho a liga alvo de desenvolvimento é a *Small Size*, que é definida da seguinte forma:

“A liga *Small Size* ou F180 como é conhecida, é uma das ligas mais antigas da *Robocup*. Ela é focada no problema da robótica de inteligência multi-agente cooperativa e controle envolvendo alta dinâmica com um sistema híbrido centralizado/distribuído”. [Rob11]

Nesta liga os robôs são bem menores e o ambiente é muito mais controlado. Embora sejam independentes uns dos outros, todo o processamento é realizado em um computador central, baseado em informações (posição dos jogadores, da bola, etc.) oriundas de uma câmera de

vídeo colocada por cima do campo de jogo, os comandos são enviados para os robôs via *wireless*. A intervenção humana resume-se à introdução e remoção dos robôs no campo. A figura 1.3 apresenta a representação do ambiente de competição da liga F180.

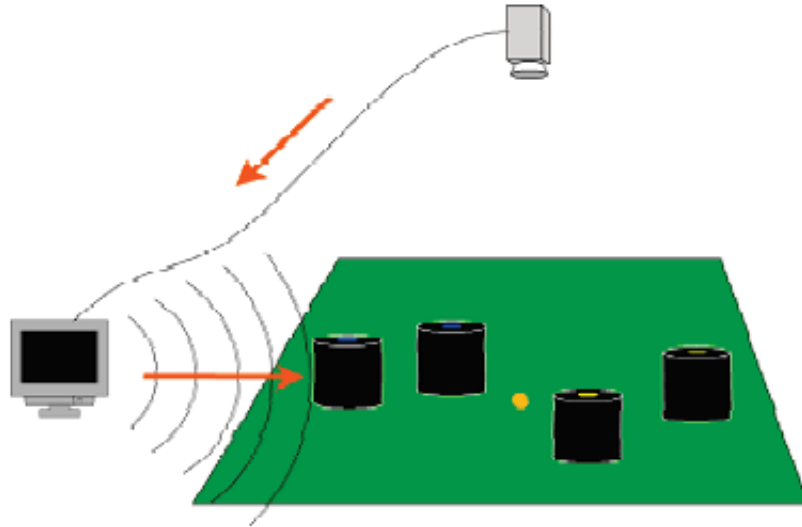


Figura 1.1: Representação do ambiente da liga F180 - Fonte: [Rob11]

## 1.4 Motivação e continuidade de pesquisa

Diversas questões estão envolvidas nos problemas de controle não linear, uma delas é o controle da trajetória para robôs móveis com rodas. Esse problema deve ser analisado cuidadosamente, e parte importante deste processo é gerar um modelo mais próximo possível do sistema real. Contudo, como já discutido o modelo não considera as incertezas dos parâmetros e dos atuadores, essas discrepâncias no modelo podem causar instabilidade ou baixo desempenho do controlador do robô. Isso é um problema em uma competição de futebol de robôs, e meios de se evitar esses pontos é uma linha interessante de problemas de controle não linear.

O trabalho desenvolvido por [Soa11] apresenta o problema de controle não linear base, para a pesquisa proposta nesse trabalho. Em [Soa11] a proposta de pesquisa foi o desenvolvimento de um modelo matemático que descreva com fidelidade o comportamento de um robô móvel omnidirecional e a concepção de um módulo de controle que promova a convergência para uma determinada trajetória, mesmo na presença de incertezas em seus parâmetros lineares.

O presente trabalho propõe uma continuidade de pesquisa ao trabalho apresentado em [Soa11]. Ao modelo não linear do robô omnidirecional da classe F-180 apresentado em [Soa11], é aplicada uma linearização por realimentação ao modelo nominal e adiciona-se incertezas paramétricas, que devem ser atenuadas por um compensador de inversão dinâmica robusta. A essa planta são propostas técnicas de síntese de ganho com intuito de garantir um desempenho previamente definido. O estudo de estabilidade se dará tanto do ponto de vista de estabilidade quanto de desempenho.

## 1.5 Proposta e escopo da pesquisa

O objetivo deste trabalho é o estudo da linearização por realimentação clássica e principalmente apresentar a utilização da técnica de compensação de Inversão Dinâmica Robusta (em inglês Robust Multi Inversion, RMI), que trata-se de um compensador de incertezas no estudo de caso do Robô F-180 de [Soa11]. Aplicando diferentes técnicas de sintonia de compensador para minimização dos efeitos de incertezas ao modelo não-linear proposto em [Soa11], analisando o desempenho dos controladores projetados frente as incertezas. Nesta proposta, podem ser elencados os seguintes objetivos específicos:

1. Estudar representações de incertezas mais adequadas para os principais parâmetros do modelo não-linear proposto em [Soa11].
2. Aplicar a técnica de compensação de incertezas por meio da inversão dinâmica robusta do sistema (IDR).
3. Investigar a aplicação de técnicas de sintonia de compensador para minimização dos efeitos de incertezas.
4. Comparar quantitativamente o desempenho dos controladores projetados por cada uma das técnicas apresentadas.

## 1.6 Estrutura da dissertação

O **Capítulo 2** apresenta conceitos fundamentais necessários para desenvolvimento e compreensão do trabalho, dentre eles: representação de incertezas, linearização por realimentação e a inversão dinâmica robusta. No **Capítulo 3**, são propostos dois algoritmos para sintonia do ganho de compensação apresentado no capítulo 2. Primeiramente é apresentado um algoritmo heurístico para determinação do ganho IDR, incluindo suas características e limitações. Em seguida é apresentado um algoritmo que utiliza Meta-Heurística para determinação do ganho IDR, sendo utilizado um algoritmo genético para tal. Os pseudo-códigos dos algoritmos são apresentados, bem como suas vantagens e limitações de utilização. No **Capítulo 4**, a técnica de linearização clássica é aplicada ao modelo do robô F-180. O modelo linearizado é utilizado em uma topologia de controle do tipo seguimento de referência em conjunto com a abordagem IDR para compensação das incertezas paramétricas, em que a síntese do controlador é feita por alocação de polos. A síntese do controlador IDR é realizada utilizando heurística e o algoritmo genético de minimização. Simulações realizadas no MATLAB são apresentadas de forma a validar a qualidade dos controladores propostos. Conclusões e perspectivas para trabalhos futuros são apresentadas no **Capítulo 5**.



## Conceitos Fundamentais e Definições

Este capítulo apresenta uma breve revisão de literatura em que itens relevantes para a compreensão do trabalho são reproduzidos. Primeiramente, é apresentado a técnica de linearização por realimentação e suas abordagens, bem como conceitos fundamentais relativos a representação de incertezas. Posteriormente é apresentada a técnica de compensação de incertezas, originalmente desenvolvida em [Lav05] que é denominada de RMI (Robust Multi-Inversion) e neste trabalho entende-se que a melhor tradução seja IDR (Inversão Dinâmica Robusta), com intuito de familiarizar o leitor com o desenvolvimento do trabalho nas próximas etapas.

### 2.1 Linearização por Realimentação

Linearização por realimentação (em inglês: *Feedback Linearization*) é uma abordagem para projeto de controle não-linear que tem atraído pesquisadores nos últimos anos [SL91]. A ideia central desta abordagem é que por meio de uma transformação algébrica, um sistema de dinâmica não-linear seja transformado (completamente ou em parte) em um sistema linear, no qual técnicas de controle linear possam ser utilizadas. Essa abordagem difere completamente da linearização convencional de Jacob como será apresentado [SL91].

A ideia de simplificar a forma do sistema dinâmico através da escolha de uma representação diferente de estados não é inteiramente nova. Na mecânica por exemplo, é bem conhecido que a forma e complexidade de um sistema modelado depende consideravelmente da escolha do sistema de coordenadas. A linearização por realimentação pode ser vista como um meio de transformar modelos de sistemas originais em modelos equivalentes em uma forma simplificada [SL91].

Essa seção apresenta uma descrição da linearização por realimentação, seus pontos principais, utilização no projeto de controle e principais técnicas. A abordagem do emprego da técnica se dá por exemplificação para casos SISO e teoricamente expandidas para o caso MIMO.

#### 2.1.1 Conceitos Iniciais

De uma forma simplista, a linearização por realimentação consiste no cancelamento das não-linearidades do sistema não-linear tal que a dinâmica em malha fechada fique na forma linear.

A linearização por realimentação se apoia nos seguintes pontos:

- O principal obstáculo na síntese de controladores para sistemas não-lineares é a presença de não-linearidades (isoladas ou não) dentro do modelo.
- Existem muitas técnicas eficazes de síntese para os casos de modelos lineares e não para não-lineares.

É possível encontrar transformações capazes de tornar um modelo não-linear em um modelo linear, o que permitiria empregar as já conhecidas técnicas de síntese de sistemas lineares para a síntese do modelo não-linear, agora linearizado. Essa é a ideia da linearização por realimentação: prover uma transformação tal que torne um modelo não-linear em um modelo linear, em que é possível utilizar-se de conhecidas técnicas de controle linear.

### Representação Não-linear

Existem diversos tipos de representações de modelos não-lineares. O formato abordado neste trabalho são sistemas contínuos e representados em espaço estados com o seguinte formato:

$$\dot{x} = f(x) + g(x)u \quad (2.1)$$

$$y = h(x) \quad (2.2)$$

Em que  $x$  é um vetor de variáveis de estado  $n$ -dimensional;  $u$  é um vetor de variáveis de entrada manipuladas  $m$ -dimensional;  $y$  é um vetor  $m$ -dimensional de variáveis de saída controladas;  $f(x)$  é um vetor  $n$ -dimensional de funções não-lineares;  $g(x)$  é uma matrix  $n \times m$ -dimensional de funções não-lineares e  $h(x)$  é um vetor  $m$ -dimensional de funções não-lineares. Com  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$  e  $y \in \mathbb{R}^p$ , tal que existe uma lei de controle do tipo

$$u = \alpha(x) + \beta(x)v \quad (2.3)$$

e uma mudança de variáveis  $z = T(x)$  tal que o sistema (2.1) é transformado em um sistema linear equivalente. O caso SISO é explorado para facilitar a compreensão da técnica da forma:

$$\dot{z} = Az + Bv \quad (2.4)$$

### Linearização por Realimentação $\times$ Linearização Jacobiana

Linearização por realimentação é completamente diferente da técnica de linearização convencional Jacobiana, pois a linearização por realimentação realiza com exatidão a transformação de estados e a realimentação, isso se for possível uma exata transformação  $z = T(x)$  em vez de aproximações lineares da dinâmica. Considerando a linearização Jacobiana do modelo não-linear (2.1) sobre o ponto de equilíbrio  $(u_0, x_0, y_0)$ :

$$\dot{x} = \left[ \frac{\partial f(x_0)}{\partial x} + \frac{\partial g(x_0)}{\partial x} u_0 \right] (x - x_0) + g(x_0)(u - u_0) \quad (2.5)$$

$$y - y_0 = \frac{\partial h(x_0)}{\partial x} (x - x_0) \quad (2.6)$$

Usando derivadas parciais, o modelo Jacobiano pode ser escrito como um sistema linear em espaço estado,

$$\dot{x} = Ax + Bu \quad (2.7)$$

$$y = Cx \quad (2.8)$$

com as definições óbvias para as matrizes  $A$ ,  $B$  e  $C$ . É importante notar que o modelo Jacobiano é uma representação do modelo linear somente nos pontos  $x_0, u_0$  e em sua vizinha, ou seja, pontos próximos. Como um resultado, a estratégia de controle no modelo linearizado pode ser insatisfatória em performance e robustez para outros pontos de operação.

A linearização por realimentação propõe uma classe de técnicas de controle não-linear que podem produzir um modelo linear que seja exatamente uma representação do modelo não-linear original sobre uma ampla região de condições de operação, ou seja, ao contrário da linearização Jacobiana que só garante o controle com as condições de robustez e performance desejadas em torno de um único ponto, no caso da linearização por realimentação as mesmas condições são garantidas para uma ampla região com mais pontos de operação.

## Formulação Geral

Há necessidade que exista uma formulação geral da técnica para o uso em qualquer modelo de sistema não-linear. Tal notação deve ser capaz de prover uma propriedade estrutural do sistema ao anular os termos não lineares. Removendo o termo  $\alpha(x)$  por subtração,  $u$  e  $\alpha(x)$  aparecem como uma soma  $u + \alpha(x)$ . Uma classe de sistemas não-lineares para o qual se pode cancelar os termos não-lineares por realimentação, é dada pela forma:

$$\dot{x} = Ax + B\gamma(x)(u - \alpha(x)) \quad (2.9)$$

em que,  $A \in \mathbb{R}^{n \times n}$ ,  $B \in \mathbb{R}^{n \times m}$ , com o par  $A, B$  é controlável. As funções  $\alpha : \mathbb{R}^n \rightarrow \mathbb{R}^m$  e  $\gamma : \mathbb{R}^n \rightarrow \mathbb{R}^{m \times m}$ , são duas funções definidas dentro de um domínio  $D \subseteq \mathbb{R}^n$  que contém  $x = 0$ . Se o termo  $\gamma(x)$  for removido por divisão,  $u$  e  $\gamma(x)$  devem aparecer como um produto  $u\gamma(x)$ . Dessa forma se  $\gamma(x)$  é invertível para todo  $x$  pertencente ao domínio de interesse  $D$  então ele pode ser cancelado, considerando  $u = \beta(x)$  e  $v = \gamma(x)^{-1}u$ . Para tal,  $\gamma(x)^{-1}$  existe para todo  $x \in D$ . Então a realimentação linearizante é  $u = \alpha(x) + \gamma(x)^{-1}v$  e o sistema linearizado é  $\dot{x} = Ax + Bv$ . Analisando a forma do sistema (2.7), pode-se imaginar que se o sistema não estiver nessa forma, não é possível aplicar linearização por realimentação. Contudo isso não é verdade, pois o modelo de um sistema não é único, mas depende da escolha da variável de estado. Pode existir uma transformação (mudança de variáveis) que torne o sistema a forma (2.9). O sistema linearizado terá o formato da figura 2.1.

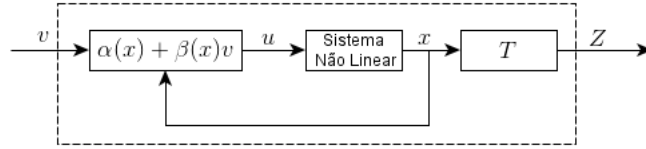


Figura 2.1: Sistema Linearizado.

A transformação  $T(x)$  deve ser invertível para encontrar  $x = T^{-1}(z)$  para todo  $z \in T(D)$ . Como as derivadas de  $x$  e de  $z$  são contínuas, a transformação  $T$  e a sua inversa também devem ser contínuas. Essa característica de  $T$  é chamada de difeomorfismo:

**Definition 2.1.1.** Uma transformação continuamente diferenciável com um inverso continuamente diferenciável é chamado de difeomorfismo. Se a matriz Jacobiana  $\frac{\partial T}{\partial x}$  é não-singular em um ponto  $x_0 \in D$ , então ele possui uma vizinhança  $N$  de  $x_0$  tal que a restrição de  $T$  sobre  $N$  é um difeomorfismo.

Uma transformação  $T$  é chamada difeomorfismo global, se e somente se é um difeomorfismo sobre  $\mathbb{R}^n$  e  $T(\mathbb{R}^n) = \mathbb{R}^n$ .  $T$  é um difeomorfismo global se e somente se  $\frac{\partial T}{\partial x}$  é não-singular  $\forall x \in \mathbb{R}^n$  e  $T$  seja próprio,  $\min ||T(x)|| = \infty$ . Um sistema não-linear,

$$\dot{x} = f(x) + g(x)u \quad (2.10)$$

em que,  $f : D \rightarrow \mathbb{R}^n$  e  $G : D \rightarrow \mathbb{R}^{n \times m}$  suficientemente suave (ou seja, todas as suas derivadas parciais são contínuas e definidas) sobre um domínio  $D$ , é chamado *Linearizado por realimentação ou linearização entrada-estado*, caso exista um difeomorfismo  $T : D \rightarrow \mathbb{R}^n$  tal que,

- $D_z = T(D)$  contém o original e a mudança de variáveis  $z = T(x)$  transformando o sistema em:

$$\dot{z} = Az + B\gamma(x)(u - \alpha(x)) \quad (2.11)$$

- o par  $(A, B)$  seja controlável e  $\gamma(x)$  seja invertível  $\forall x \in D$ .

## 2.1.2 Abordagens de Realimentação Linearizante

A literatura apresenta duas abordagens de realimentação linearizante que possibilitam tornar o sistema do formato (2.10) para o formato (2.11). A primeira é denominada linearização por realimentação clássica, iniciada nos anos 80 em trabalhos como [HSM83]. Trata-se de um método bem conhecido e difundido em diversos trabalhos acadêmicos de controle não-linear [Isi85] [SL91]. Nessa abordagem o sistema não-linear é transformado em um sistema linear na forma canônica de Brunovsky, que representa a cadeia de integradores. Consequentemente, o formato linear resultante permite o emprego de controlador com técnicas lineares. Contudo há um inconveniente no emprego desta abordagem relativo à sua alta sensibilidade a variação de parâmetros, além do sistema resultante não ter nenhuma semelhança com o sistema original [OL03].

A linearização por realimentação robusta apresentada em [GB00] tem por objetivo obter maior robustez para o sistema controlado. Após a transformação o sistema resultante representa

a aproximação linear do sistema original em torno do ponto de operação, mantendo informações do sistema original. Dessa forma, o comportamento do sistema linearizado é semelhante ao comportamento do sistema original, resultando em maior robustez do sistema em malha fechada. Consequentemente para variações de parâmetros e/ou incertezas de modelo não-estruturadas os dois sistemas apresentam respostas semelhantes. Portanto, as propriedades atribuídas ao controlador linear são conservadas quando ele é associado à linearização por realimentação robusta e aplicado ao sistema não-linear [Fra06].

Obviamente ambas as técnicas possuem limitações tangíveis à sua aplicação. É necessário uma modelagem que represente precisamente o comportamento dinâmico do sistema não-linear, uma vez que a partir da modelagem será sintetizada uma lei de controle que cancelará as não-linearidades, porém podem existir perturbações paramétricas. A existência de um difeomorfismo é restringida pela necessidade do sistema não-linear atender as condições suficientes e necessárias que garantam que ele seja linearizável pela realimentação, pelo menos em uma região do espaço de estados. Uma outra restrição é a necessidade de acesso a informação de todos os estados, que representam grandezas utilizadas na modelagem dinâmica do sistema, para implementação da lei de controle.

### Condições para uso da Linearização por Realimentação

Para que as condições necessárias para o uso da linearização por realimentação sejam apresentadas, é necessário que o conceito de grau relativo de sistemas não-lineares seja conhecido. O grau relativo é definido como o número de vezes que é necessário derivar a saída do sistema para que a entrada de controle apareça na equação [SL91].

#### Definition 2.1.2. Definição de grau relativo

Dado o sistema não-linear:

$$\dot{x} = f(x) + g(x)u \quad y = h(x) \quad (2.12)$$

em que  $x \in \mathbb{R}^n$  representa os estados,  $u \in \mathbb{R}^m$  é a entrada de controle,  $y \in \mathbb{R}^p$  é a saída e  $f(x), g_1(x), \dots, g_m(x), h(x)$  são campos vetoriais suaves em um subconjunto aberto de  $\mathbb{R}^n$ .

Cada saída  $h_j(x)$  tem grau relativo  $r_j$ , se:

$$\mathcal{L}_{g_i} h_f(x) = \mathcal{L}_{g_1} \mathcal{L}_f h_f(x) = \dots = \mathcal{L}_{g_1} \mathcal{L}_f^{r_j-2} h_j(x) = 0 \quad \forall i \in [l, m] \quad (2.13)$$

e se existe pelo menos um  $i$  tal que:

$$\mathcal{L}_f^{r_j-1} h_j(x) \neq 0 \quad (2.14)$$

Em que o operador  $\mathcal{L}$  representa a derivada de Lie [Wik14b], que é uma derivação na álgebra de funções diferenciais sobre uma variação diferenciável  $\mathcal{M}$ , veja B.1. O sistema não linear 2.12 tem grau relativo  $r = r_1 + \dots + r_p$ .

Considerando a definição 2.1.2, o sistema 2.12 tem grau relativo  $r$  em um ponto de equilíbrio,  $x_0$ , se for necessário derivar  $r$  vezes a saída  $y$  para que a entrada  $u$  apareça no equacionamento. Em [Fra06] são definidas distribuição e involutividade da seguinte forma:

**Definition 2.1.3.** Definição (Distribuição)

Uma distribuição  $D$  em uma variedade  $M$  é uma função que atribui a cada  $p \in M$  um subespaço linear  $D(p)$  do espaço tangente  $T_p M$ .

**Definition 2.1.4.** Definição (Involutividade)

Uma distribuição  $D$  é chamada involutiva de  $[X, Y] \in D$  sempre que  $X$  e  $Y$  são campos vetoriais de  $D$ .

Sejam as distribuições  $G_0 \subset G_1 \subset \dots \subset G_{n-1}$  associadas ao sistema não-linear 2.12 e construídas de acordo com

$$G_1 = \text{span} \{ ad_f^k g_1 : 0 \leq k \leq i, 1 \leq j \leq m, 0 \leq i \leq n-1 \}, \quad (2.15)$$

em que  $ad_f g = [f, g]$  representa o colchete de Lie, conforme a definição do apêndice B.1.

Assume-se que as distribuições  $G_1$  satisfazem as hipóteses:

- **(H1)** Cada uma das distribuições  $G_i$ , para  $0 \leq i \leq (n-1)$ , tem dimensão constante ao redor de  $x_0$ ,
- **(H2)** cada uma das distribuições  $G_i$ , para  $0 \leq i \leq (n-2)$  é involutiva e
- **(H3)** a distribuição  $G_{n-1}$  tem dimensão  $n$ .

De acordo com [Isi85], se as distribuições  $G_0, G_1, \dots, G_{n-1}$  satisfazem as hipóteses **H1**, **H2** e **H3** então existe funções escalares  $\lambda_1(x), \dots, \lambda_m(x)$  definidas em uma vizinhança  $U$  de  $x_0$  de graus relativos  $r_1, \dots, r_m$  com  $r_1 + \dots + r_m = n$ , tais que:

- Para todo  $i \in [1, m]$ , todo  $j \in [1, m]$  e todo  $x \in U$   
 $\mathcal{L}_{g_i} \lambda_j(x) = \mathcal{L}_{g_i} \mathcal{L}_f \lambda_j(x) = \dots = \mathcal{L}_{g_i} \mathcal{L}_f^{r_j-2} \lambda_j(x) = 0$ .
- A matriz  $E(x)$  dada por

$$E(x) = \begin{bmatrix} \mathcal{L}_{g_1} \mathcal{L}_f^{r_1-1} \lambda_1 & \dots & \mathcal{L}_{g_m} \mathcal{L}_f^{r_1-1} \lambda_1 \\ \vdots & \ddots & \vdots \\ \mathcal{L}_{g_i} \mathcal{L}_f^{r_m-1} \lambda_m & \dots & \mathcal{L}_{g_m} \mathcal{L}_f^{r_m-1} \lambda_m \end{bmatrix};$$

é não singular para todo  $x \in U$ .

O vetor  $\lambda(x)$  formado pelas funções  $\lambda_1(x), \dots, \lambda_m(x)$  na forma

$$\lambda(x) = \begin{bmatrix} \lambda_1(x) \\ \vdots \\ \lambda_m(x) \end{bmatrix};$$

é escolhido como a saída do sistema 2.12, resultando no novo sistema

$$\dot{x} = f(x) + g(x)u = f(x) + \sum_{i=1}^m g_i(x)u_i, y = \lambda(x) \quad (2.16)$$

Esse novo sistema possui  $m$  entradas e  $m$  saídas, sendo denominado sistema quadrado. Se as distribuições  $G_0, G_1, \dots, G_{n-1}$  satisfazem as hipóteses **H1**, **H2** e **H3**, então o sistema não-linear (2.16) é linearizável por realimentação. Definida as condições para que um sistema seja linearizável, passa-se à apresentação dos métodos de linearização por realimentação clássica e de linearização por realimentação robusta.

### Linearização Clássica

O objetivo da técnica de linearização clássica é obter uma transformação de estados (difeomorfismo) e uma transformação de entrada que linearizem o sistema (2.16), resultando em um sistema na forma de Brunovsky.

$$\dot{x}_c = A_c x_c + B_c v_c \quad (2.17)$$

$$y = C_c x_c \quad (2.18)$$

sendo  $x_c \in \mathbb{R}^n$  o estado linearizado,  $v_c \in \mathbb{R}^n$  a entrada de controle e as matrizes  $A_c, B_c$  e  $C_c$  dadas por

$$A_c = \begin{bmatrix} A_{c1} & 0_{r1 \times r2} & \dots & 0_{r1 \times rm} \\ 0_{r2 \times r1} & A_{c2} & \dots & 0_{r2 \times rm} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{rm \times r1} & 0_{rm \times r2} & \dots & A_{cm} \end{bmatrix};$$

$$B_c = \begin{bmatrix} B_{c1} & 0_{r1 \times 1} & \dots & 0_{r1 \times 1} \\ 0_{r2 \times 1} & B_{c2} & \dots & 0_{r2 \times 1} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{rm \times 1} & 0_{rm \times 1} & \dots & B_{cm} \end{bmatrix};$$

$$C_c = \begin{bmatrix} C_{c1} & 0_{1 \times r2} & \dots & 0_{1 \times rm} \\ 0_{1 \times r1} & C_{c2} & \dots & 0_{1 \times rm} \\ \vdots & \vdots & \ddots & \vdots \\ 0_{1 \times r1} & 0_{1 \times r2} & \dots & C_{cm} \end{bmatrix}$$

Sendo que cada matriz  $A_{ci}$  é uma matriz  $r_i \times r_i$  cuja diagonal é composta por 1 e todos os outros elementos são zeros, cada vetor  $B_{ci}$  é um vetor  $r_i \times 1$  cujo último elemento vale 1 e todos os outros elementos são zeros e cada vetor  $C_{ci}$  é um vetor  $1 \times r_i$  cujo primeiro elemento é um e todos os outros são zeros.

Assume-se que existe um difeomorfismo  $T(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$  válido numa região  $\Omega$  para o sistema (2.16). O resultado dessa transformação de estados sobre o sistema não-linear, é um sistema não-linear com as não linearidades diretamente associadas ao controle, com parte do sistema na forma de cascata de integradores. Para realizar a transformação de (2.16) para (2.17), é preciso determinar um difeomorfismo  $T(x)$  e uma transformação de entrada  $u_c(x)$  que cancele as não linearidades desse sistema e forneça uma nova entrada de controle.

Para se obter a linearização, deve-se derivar a saída  $y$  da equação (2.16) no tempo até que a entrada do sistema apareça na equação resultante.

$$\dot{y} = \frac{\partial \lambda(x)}{\partial x} \dot{x} \quad (2.19)$$

Substituindo  $\dot{x}$  na equação (2.19), tem-se

$$\dot{y} = \frac{\partial \lambda(x)}{\partial x} [f(x) + g(x)u], \quad (2.20)$$

$$\dot{y} = \frac{\partial \lambda(x)}{\partial x} f(x) + \frac{\partial \lambda(x)}{\partial x} g(x)u \quad (2.21)$$

Utilizando a notação da derivada de Lie (veja apêndice B.1), tem-se

$$\dot{y} = \mathcal{L}_f \lambda(x) + \mathcal{L}_g \lambda(x)u \quad (2.22)$$

Estendendo essas derivadas para todas as saídas do sistema, a equação (2.22) pode ser escrita de forma mais geral, conforme mostrado na equação (2.23).

$$\dot{y}_i = \mathcal{L}_f \lambda_j(x) + \sum_{i=1}^m (\mathcal{L}_{g_i} \lambda_j(x))u_i, \quad (2.23)$$

em que  $\mathcal{L}_f \lambda_j(x)$  é a derivada de Lie da saída  $\lambda_j(x)$  correspondente ao termo de  $f(x)$  e  $\mathcal{L}_{g_i} \lambda_j(x)$ , a derivada de Lie de saída  $\lambda_j(x)$ , correspondente aos termos de  $g_i(x)$ , representadas por

$$\mathcal{L}_f \lambda_j(x) = \sum_{i=1}^m \frac{\partial \lambda_j(x)}{\partial x_i} \dot{x}_i f \quad (2.24)$$

$$\mathcal{L}_{g_i} \lambda_j(x) = \sum_{i=1}^m \frac{\partial \lambda_j(x)}{\partial x_i} \dot{x}_i g \quad (2.25)$$

Se  $\mathcal{L}_{g_i} \lambda_j(x) = 0$  para todo  $i$ , as entradas não aparecem na expressão resultante e a saída tem que ser diferenciada novamente. Considerando que  $r_j$  é o menor inteiro tal que no mínimo uma das entradas apareçam em  $y_j^{r_j}$ , então

$$y_j^{r_j} = \mathcal{L}_f^{r_j} \lambda_j(x) + \sum_{i=1}^m (\mathcal{L}_{g_i} \mathcal{L}_f^{(r_j-1)} \lambda_j(x))u_i \quad (2.26)$$

com  $\mathcal{L}_{g_i} \mathcal{L}_f^{(r_j-1)} \lambda_j(x) \neq 0$  para no mínimo um  $i$ , para todo o  $x$  pertencente à região onde a linearização é válida.

Aqui,  $r_j$ , representa o grau relativo referente a saída  $j$ . Utilizando o método de linearização entrada-saída, tem-se que para que um sistema não linear seja completamente linearizável é necessário que esse tenha grau relativo igual ao número de estados, ou seja,

$$\sum_{j=1}^n r_j = n, \quad (2.27)$$

em que  $n$  é o número de estados do sistema. Quando  $\sum_{j=1}^n r_j < n$ , o sistema não pode ser totalmente linearizável. Nessa situação há dois subsistemas, um que é linearizável e outro denominado dinâmica nula. Para o presente trabalho, são considerados sistemas com grau relativo igual ao número de estados, ou seja, totalmente linearizável.



A equação (2.27) pode ser escrita no seguinte formato

$$\begin{bmatrix} y_1^{r_1} \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} \mathcal{L}_f^{r_1} \lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m} \lambda_m(x) \end{bmatrix} + E(x) \begin{bmatrix} u_1 \\ \vdots \\ u_m \end{bmatrix};$$

Como já mencionado, a matriz  $E(x)$  é uma matriz de desacoplamento do sistema MIMO. Se essa matriz for não-singular, a transformação da entrada

$$\begin{bmatrix} u_1^{r_1} \\ \vdots \\ u_m \end{bmatrix} = -E^{-1}(x) \begin{bmatrix} \mathcal{L}_f^{r_1} \lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m} \lambda_m(x) \end{bmatrix} + E(x) \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix}; \quad (2.28)$$

leva a uma relação linear entre a saída  $y_i$  e a nova entrada  $v_j$

$$\begin{bmatrix} y_1^{r_1} \\ \vdots \\ y_m^{r_m} \end{bmatrix} = \begin{bmatrix} v_1 \\ \vdots \\ v_m \end{bmatrix};$$

Observe que a relação entrada-saída acima é desacoplada e linear e uma lei de controle no formato de (2.28) é chamada de lei de controle desacoplada ou lei de controle não-iterativa. Devido ao desacoplamento, pode ser utilizado um sistema de uma entrada e uma saída (SISO) para cada canal  $y_i - v_j$  da dinâmica desacoplada acima, de modo a obter controladores estáveis [SL91]. Utilizando a notação  $u(x, v) = \alpha(x) + \beta(x)v$ , a equação (2.28) pode ser reescrita com o seguinte formato:

$$u_c(x, v_c) = \alpha_c(x) + \beta_c(x)v_c \quad (2.29)$$

Em que  $\alpha_c(x)$  e  $\beta_c(x)$  são dadas por

$$\alpha_c(x) = -E^{-1}(x) \begin{bmatrix} \mathcal{L}_f^{r_1} \lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m} \lambda_m(x) \end{bmatrix} \quad (2.30)$$

$$\beta_c(x) = E^{-1}(x) \quad (2.31)$$

Determinada a realimentação linearizante, o próximo passo é a determinação do difeomorfismo  $T(x)$  que transforme juntamente (2.29), o sistema (2.16) no sistema (2.17). De acordo com [SL91] pode-se assumir a igualdade  $T(x) = x_c(x)$ , e o difeomorfismo é dado por:

$$T(x) = \begin{bmatrix} T_{c_1}(x) \\ \vdots \\ T_{c_m}(x) \end{bmatrix} \quad (2.32)$$

Logo

$$T(x) = \begin{bmatrix} \lambda_j(x) \\ \mathcal{L}_f \lambda_1(x) \\ \vdots \\ \mathcal{L}_f^{r_m-1} \lambda_j(x) \end{bmatrix} \quad (2.33)$$

Derivando o difeomorfismo  $T_c(x)$  em relação a  $x$ , tem-se

$$\dot{x}_c(x) = \frac{\partial x_c(x)}{\partial x}(\dot{x}) \quad (2.34)$$

Substituindo  $\dot{x}$  na equação (2.34), obtêm-se

$$\begin{aligned} \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}[f(x) + g(x)u] \\ \dot{x}_c(x) &= \frac{\partial x_c(x)}{\partial x}f(x) + \frac{\partial x_c(x)}{\partial x}g(x)u \end{aligned} \quad (2.35)$$

Aplicando a notação da derivada de Lie, tem-se

$$\dot{x}_c(x) = \mathcal{L}_f x_c(x) + \mathcal{L}_g x_c(x)\alpha_c(x) + \mathcal{L}_g x_c(x)\beta_c(x)v_c \quad (2.36)$$

A equação (2.36) é reescrita na forma canônica de Brunovsky conforme (2.37).

$$\dot{x}_c(x) = A_c x_c(x) + B_c v_c \quad (2.37)$$

$$y = C_c x_c(x) \quad (2.38)$$

em que

$$A_c x_c(x) = \mathcal{L}_f x_c(x) + \mathcal{L}_g x_c(x)\alpha_c(x) \quad (2.39)$$

$$B_c = \mathcal{L}_g x_c(x)\beta_c(x) \quad (2.40)$$

Observe que (2.37) está representada em termos de variáveis de estado originais  $x$ . Para transformar para as novas variáveis de estado, basta utilizar  $x = T^{-1}(x_c)$ . A linearização por realimentação clássica possui duas desvantagens ligadas à forma de Brunovsky:

- Perda do sentido físico dos estados do sistema, pois as novas variáveis de estado  $x_c$  nem sempre têm uma interpretação física, o que pode dificultar a síntese de controladores lineares, já que é mais difícil estabelecer objetivos e especificações para essas variáveis abstratas. Além disso, qualquer sistema de mesma ordem, terá a mesma forma canônica de Brunovsky.
- O sistema não-linear e o linearizado apresentam comportamentos que podem ser diferentes devido ao cancelamento simples das não linearidades, além disso, pequenas variações de parâmetros podem impedir o cancelamento de alguma não linearidade. Assim, se o controlador linear for projetado com a suposição de que todos os cancelamentos foram realizados, é possível que se obtenha uma resposta inesperada, ou seja, que haja uma falha do controlador, conseqüentemente, o sistema em malha fechada não é, em geral, robusto.

### 2.1.3 Abordagens de Linearização Clássica

Na técnica de linearização clássica, pode-se buscar realizar a linearização Entrada-Estados ou Linearização Entrada-Saída. A escolha depende de quais são as variáveis de interesse, os estados ou a saída. As seções a seguir apresentam as diferenças e o emprego de cada abordagem em especial.

### Linearização Entrada-Estado

Alguns resultados teóricos do emprego da técnica de linearização por realimentação entrada-estados, bem como o descritivo do método e as etapas de solução para o sistema não-linear são apresentados na sequência. São considerados sistemas não-lineares com uma única entrada ( $m = 1$ ) e sempre é utilizada a notação na forma normal. Considerando o sistema não-linear seguinte:

$$\dot{x} = f(x) + g(x)u \quad (2.41)$$

em que  $f(\cdot)$  e  $g(\cdot)$  são dois campos vetoriais suaves. O sistema será dito linearizável se existir,

- Uma região  $\mathbb{D}$  de  $\mathbb{R}^n$
- Um dioformismo  $T(x)$  de  $\mathbb{D}$  em  $\mathbb{R}^n$
- Uma realimentação de estados não-linear

$$u = \alpha(x) + \gamma(x)v \quad (2.42)$$

Que reduz o sistema não linear (2.41) para a forma

$$\dot{z} = Az + Bv \quad (2.43)$$

com  $z = T(x)$  e

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & . \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

**Teorema 1:** O sistema não-linear (2.41) em que  $f$  e  $g$  são dois campos vetoriais suaves possui a entrada-estado linearizável somente se existe uma região  $\mathbb{D}$  tal que [SL91],

1. Os campos de vetores  $g, ad_f g, ad_f^2 g, \dots, ad_f^{m-1} g$ , são linearmente independentes (L.I) em  $D$ .
2. O conjunto de campos de vetores  $g, ad_f g, ad_f^2 g, \dots, ad_f^{n-2} g$ , são involutivos em  $\mathbb{D}$ .

A compreensão desses resultados se faz necessária, considerando a abordagem da condição 1 do teorema 1. Dito que os vetores  $g, ad_f g, ad_f^2 g, \dots, ad_f^{m-1} g$ , são L.I em  $\mathbb{D}$ , é o mesmo que dizer que:

- $\text{posto} [g, ad_f g, ad_f^2 g, \dots, ad_f^{m-1} g] = n, \forall x \in \mathbb{D}$ .

Considerando o sistema (2.41), em que  $f$  e  $g$  são:

$$f(x) = Ax; \quad g(x) = B$$

A condição 1 do teorema 1 equivale a

$$\text{posto} [g \text{ } ad_f g \text{ } ad_f^2 g \text{ } \dots \text{ } ad_f^{n-1} g] = \text{posto} [B \text{ } AB \text{ } A^2 B \text{ } \dots \text{ } A^{n-1} B] = n.$$

Essa condição corresponde a uma condição linear de controlabilidade que garante a sua invertibilidade. Também verifica-se que a condição 2 do teorema 1 também é satisfeita. A condição 1 pode ser vista como uma generalização da condição de controlabilidade. Considerando uma aproximação linear do sistema não-linear em uma região relacionada  $\mathbb{D}$ :

- Essa condição pode trazer o sistema de um ponto de operação em  $\mathbb{D}$  para outro ponto de operação em  $\mathbb{D}$ .
- Contudo é importante observar, que a parte não-linear do sistema pode ser controlável enquanto sua parte linear não é. Isso é uma outra particularidade de sistemas não-lineares.

**Procedimento de Linearização entrada-estado** A seguir é apresentado um procedimento para a linearização entrada-estado de um sistema não-linear.

1. Construir os campos vetoriais  $g, ad_f g, ad_f^2 g, \dots, ad_f^{n-1} g$
2. Verificar as duas condições apresentadas no teorema 1, verificando as condições de controlabilidade e involutividade.
3. Se as duas condições são satisfeitas, então encontra-se a função  $T(x)$  tal que,

$$\nabla T_1 ad_f^i g = 0; \quad i = 0, \dots, n-2 \quad (2.44)$$

$$\nabla T_1 ad_f^{n-1} g \neq 0 \quad (2.45)$$

4. Calcular a transformação linearizante

$$T(x) = \begin{bmatrix} T_1(x) \\ \mathcal{L}_f T_1(x) \\ \vdots \\ \mathcal{L}_f^{n-1} T_1(x) \end{bmatrix}$$

5. Calcula o sinal de comando linearizante  $u = \alpha(x) + \gamma(x)v$ , no caso SISO, com

$$\alpha(x) = -\frac{\mathcal{L}_f^n T_1(x)}{\mathcal{L}_g \mathcal{L}_f^{n-1} T_1(x)}; \quad \gamma(x) = \frac{1}{\mathcal{L}_g \mathcal{L}_f^{n-1} T_1(x)}$$

## Linearização Entrada-Saída

Quando as variáveis de saída são de interesse, o modelo considerado é descrito por equações de entrada e saída. A linearização da equação de entrada-estado não é necessariamente uma linearização da equação de entrada-saída. Problemas desse tipo podem ser complexos, pois a saída pode ser dependente da entrada e/ou de um estado de maneira complexa. Não sendo

tão fácil calcular a entrada linear para se obter a saída desejada. Assim a ideia de linearização entrada-saída consiste em, se possível, encontrar uma simplificação da relação entrada/saída. Essa técnica utiliza as derivadas da saída(s), como é exemplificado no sistema não-linear de [SL91], a seguir:

Exemplo 2.1.3.1

$$\dot{x}_1 = \sin(x_2) + (x_2 + 1)x_3 \quad (2.46)$$

$$\dot{x}_2 = x_1^5 + x_3 \quad (2.47)$$

$$\dot{x}_3 = x_1^2 + u \quad (2.48)$$

$$y = x_1 \quad (2.49)$$

A primeira derivada da saída,

$$\dot{y} = \sin(x_2) + (x_2 + 1)x_3 \quad (2.50)$$

A segunda derivada nos fornece uma relação de entrada-saída,

$$\ddot{y} = (x_2 + 1)u + (x_1^5 + x_3)(x_3 + \cos(x_2)) + (x_2 + 1)x_3^2 \quad (2.51)$$

A entrada é dada por,

$$u = \frac{1}{(x_2 + 1)}(v - (x_1^5 + x_3)(x_3 + \cos(x_2)) - (x_2 + 1)x_3^2) \quad (2.52)$$

Logo o sistema linearizado obtido:  $\ddot{y} = v$  e com erro definido por  $e = y - y_d$ . Escolhendo uma nova entrada,  $v = \ddot{y}_d - k_1\dot{e} - k_2\ddot{e}$ . Para  $k_1$  e  $k_2$  positivos, o erro verificado

$$\ddot{e} + k_2\dot{e} + k_1e = 0 \quad (2.53)$$

Se as condições iniciais de  $e$  são nulas, então pode-se seguir de maneira exata a trajetória de referência. Senão  $e$  converge exponencialmente para zero.

Para a linearização de sistemas não-lineares entrada-saída, a demonstração teórica irá considerar sistemas com uma única entrada e única saída, ou seja, sistemas SISO como já mencionado ( $m = 1$  e  $p = 1$ ). Como apresentado para a linearização entrada-estado, também será descrito um procedimento de linearização para entrada-saída. Para tal sempre é utilizado a noção de grau relativo, sistemas na forma normal e uma abordagem de dinâmica de zeros. Considerando o sistema não-linear a seguir:

$$\dot{x} = f(x) + g(x)u \quad (2.54)$$

$$y = h(x) \quad (2.55)$$

Em que,  $f(\cdot)$  e  $g(\cdot)$  são dois campos de vetores suaves e a variável  $y$  é a saída. Como apresentado anteriormente, para se encontrar uma relação entrada-saída, foi necessário derivar a saída. O número de derivadas necessárias para se obter a variável de entrada na equação, ou seja, relação entrada-saída é denominado grau relativo do sistema. Em alguns casos de sistemas, o grau relativo não pode ser definido. Para o sistema proposto, tem-se:

$$\mathcal{L}_g \mathcal{L}_f^i h(x) = 0; \quad 0 \leq i < r - 1 \quad (2.56)$$

$$\mathcal{L}_g \mathcal{L}_f^{r-1} h(x) \neq 0; \quad (2.57)$$

para todo  $x_0 \in D$ , é dito que o sistema tem grau relativo  $r$  em  $x_0$ . Derivando a saída do sistema obtêm-se:

$$\begin{aligned}\dot{y} &= \nabla h \dot{x} = \nabla h(f + gu) = \mathcal{L}_{f+gu}h = (\mathcal{L}_gh)u \\ \ddot{y} &= \left(\frac{\partial(\mathcal{L}_{f+gu}.h)}{\partial x}\right)(f + gu) = \nabla(\mathcal{L}_{f+gu}.h)(f + gu) \\ &= \mathcal{L}_{f+gu}^2h = \mathcal{L}_{f+gu}(\mathcal{L}_{f+gu}h) \\ &= \mathcal{L}_f(\mathcal{L}_{f+gu}h) + (\mathcal{L}_g(\mathcal{L}_ff + guh))u \\ &= \mathcal{L}_f(\mathcal{L}_fh + (\mathcal{L}_gh)u) + (\mathcal{L}_g(\mathcal{L}_fh + (\mathcal{L}_gh)u))u\end{aligned}$$

Logo,

$$y^{(i)} = \mathcal{L}_{f+g}^i u h \quad (2.58)$$

No caso em que,  $\mathcal{L}_g = 0$  a forma é simplificada para:

$$\begin{aligned}\dot{y} &= \mathcal{L}_fh \\ \ddot{y} &= \mathcal{L}_f^2h + \mathcal{L}_g\mathcal{L}_fh u\end{aligned}$$

No caso em que,  $\mathcal{L}_gh = 0$ ,  $\mathcal{L}_g\mathcal{L}_fh = 0$  resulta em:

$$\begin{aligned}\dot{y} &= \mathcal{L}_fh \\ \ddot{y} &= \mathcal{L}_f^2h\end{aligned}$$

É importante salientar esse caso, pois o comando  $u$  não aparece na duas primeiras derivadas. Se for assumido que  $\mathcal{L}_g\mathcal{L}_f^i h(x) = 0$ ,  $0 < i < r - 1$ , então a primeira derivada em que  $u$  aparece é,

$$y^{(r)} = \mathcal{L}_f^r h + \mathcal{L}_g\mathcal{L}_f^{r-1} h u \quad (2.59)$$

Isso permite linearizar o sistema, e o comando linearizante é dado por:

$$u = \frac{1}{\mathcal{L}_g\mathcal{L}_f^{r-1}h}(-\mathcal{L}_f^r h + v) \quad (2.60)$$

Então,  $y^r = v$ . Deve-se derivar a saída  $r$  vezes para que o comando linearizante apareça. É importante salientar que para se obter uma linearização entrada-saída é essencial que  $r < n$ , em que  $n$  é a ordem do sistema. No caso em que  $r = n$  encontra-se a linearização entrada-estado. Considerando o caso de um sistema linear definido pela forma,  $f(x) = Ax$ ,  $g(x) = B$  e  $h(x) = C(x)$ , a derivada  $r$  da saída é escrita definida como:

$$y^{(r)} = CA^r x + CBu^{r-1} + CABu^{r-2} + CA^2Bu^{r-1} + \dots + CA^{r-1}Bu \quad (2.61)$$

Desenvolvendo a função transferência  $G(s) = C(sI - A)^{-1}B$  na série de Laurent, resulta em:

$$G(s) = CBs^{-1} + CABs^{-2} + CA^2Bs^{-3} + \dots + CA^iBs^{-(i+1)} + \dots \quad (2.62)$$

Esse desenvolvimento é válido se  $|s| > \max_i |\lambda_i(A)|$ . Em que chamamos os coeficientes  $CA^iB$  de coeficientes de Markov da função de transferência. Se o sistema for de grau relativo  $r$ , significa que os  $r$  primeiros parâmetros de Markov da função transferência são nulos. A função transferência então é escrita por:

$$G(s) = s^{-r+1}(CA^{r-1}Bs^{-1} + CA^rBs^{-2} + \dots) \quad (2.63)$$

$$= s^{-r+1}CA^{r-1}(sI - A)^{-1}B = \frac{s^{-r+1}CA^{r-1}(\widetilde{sI - A})B}{\det(sI - A)} \quad (2.64)$$

Em que  $\widetilde{(sI - A)}$  é a matriz adjunta de  $(sI - A)$ , composta de polinômios de ordem  $n - 1$ . O numerador da função transferência é de ordem  $n - r$  e  $r$  é a diferença entre o grau do numerador e do denominador.

Alguns pontos importantes devem ser salientados, o grau relativo de um sistema não-linear não pode ser definido no ponto  $x_0$  de  $\mathbb{D}$ . É possível existir  $\mathcal{L}_g \mathcal{L}_f^{r-1} h(x_0) = 0$ , na vizinhança de  $x_0$  em  $\mathcal{L}_g \mathcal{L}_f^{r-1} h(x) \neq 0$ . É por exemplo o caso quando  $\mathcal{L}_g \mathcal{L}_f^{r-1} h(x)$  muda de sinal em uma superfície contendo o ponto  $x_0$ . Quando o grau relativo é tal que  $r < n$  em  $x_0$ , pode-se colocar o sistema em uma forma denominada forma normal. Para tal, o vetor aumentado é definido por,

$$\epsilon = \begin{bmatrix} y \\ \dot{y} \\ \vdots \\ y^{r-1} \end{bmatrix}$$

O vetor  $\epsilon$  representa as primeiras componentes do estado. O vetor de estados é completado com um vetor  $\eta$  de  $n - r$  componentes. Dentro da vizinhança de  $x_0$ , a forma normal é escrita como:

$$\dot{\epsilon}_1 = \epsilon_2 \quad (2.65)$$

$$\vdots = \vdots \quad (2.66)$$

$$\dot{\epsilon}_{r-1} = \epsilon_r \quad (2.67)$$

$$\dot{\epsilon}_r = a(\epsilon, \eta) + b(\epsilon, \eta)u \quad (2.68)$$

$$\dot{\eta} = \omega(\epsilon, \eta) \quad (2.69)$$

$$y = \epsilon_1 \quad (2.70)$$

As funções  $a(...)$  e  $b(...)$  são definidas por:

$$a(\epsilon, \eta) = \mathcal{L}_f^r h(x); b(\epsilon, \eta) = \mathcal{L}_f^{r-1} h(x)$$

Para mostrar que o sistema pode ser colocado neste formato, é necessário provar que a transformação a seguir é um difeomorfismo local.

$$T(x) = \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_r \\ \eta_1 \\ \vdots \\ \eta^{n-r} \end{bmatrix}$$

Quando isso ocorre o jacobiano de  $\nabla T$  é invertível. Isso é equivalente a mostrar que os gradientes  $\nabla \epsilon_i$  e  $\nabla \eta_i$  são linearmente independentes.

Uma formalização do resultado é expressa da seguinte forma:

**Lemma:** Se o sistema proposto é de grau relativo  $r$  em  $\mathbb{D}$ , então os gradientes

$$\nabla \epsilon_1, \nabla \epsilon_2, \dots, \nabla \epsilon_r \quad (2.71)$$

são linearmente independentes em  $\mathbb{D}$ .

## 2.2 Representação de Incertezas

Dizemos que um sistema de controle é robusto se esse é insensível a diferenças entre o sistema real e o modelo do sistema que foi usado para projetar o controlador. Essas diferenças são referidas como erro de modelo/planta ou simplesmente incerteza do modelo. No paradigma de controle robusto  $\mathcal{H}_\infty$  é usado para verificar se as especificações estão sendo atendidas mesmo para o pior caso de incertezas [Sko01]. As incertezas de um modelo podem ser classificadas em incertezas paramétricas ou incertezas de dinâmica não-modelada (incerteza negligenciada), que são descritas a seguir.

### 2.2.1 Incertezas Paramétricas

No caso de incertezas paramétricas, a estrutura do modelo, incluindo a ordem, é conhecida. Contudo alguns parâmetros são incertos. A incerteza paramétrica irá ser quantificada, assumindo que qualquer parâmetro incerto é limitado dentro de alguma região  $[\alpha_{min}, \alpha_{max}]$ . Dessa forma o conjunto de parâmetros possuem o seguinte formato [Sko01],

$$\alpha_p = \bar{\alpha}(1 + r_\alpha \Delta) \quad (2.72)$$

Em que:  $\bar{\alpha}$  é a média do valor paramétrico;  $r_\alpha = (\alpha_{max} - \alpha_{min})/(\alpha_{max} + \alpha_{min})$ , é a incerteza relativa no parâmetro; e que  $\Delta$  é qualquer número real escalar que satisfaça  $|\Delta| < 1$ .

### 2.2.2 Incertezas Negligenciadas ou de Dinâmica Não-modelada

As incertezas negligenciadas são menos precisas, sendo mais difícil de serem quantificadas. Modelos com esse tipo de incerteza apresentam erros referentes a não-modelagem de fatores dinâmicos. Elas são usualmente caracterizadas por altas frequências negligenciadas por deliberaridade ou por uma má compreensão do problema físico. Dessa forma elas estão presentes em qualquer modelo de um sistema real, levando o sistema a perturbações complexas com normalização  $\|\Delta\|_\infty \leq 1$ .

### 2.2.3 Incertezas Concentradas

As incertezas concentradas são uma combinação dos outros dois tipos de incerteza apresentados anteriormente. Aqui a descrição da incerteza representa uma ou várias fontes de parâmetros e/ou incertezas de dinâmicas não modeladas, combinadas dentro de uma única perturbação irregular (*lumped*) da estrutura escolhida. A figura 2.2 apresenta um modelo de um sistema com múltiplas incertezas (Paramétricas e Negligenciadas). Nesse caso,  $\Delta I(s)$  é qualquer função transferência estável que tenha magnitude menor ou igual que um 1 em cada frequência. Alguns exemplos de  $\Delta I(s)$ 's permissíveis com  $\mathcal{H}_\infty$  menor que um, são

$$\frac{1}{(5s + 1)^3}; \quad \frac{0.1}{s^2 + 0.1s + 1} \quad (2.73)$$



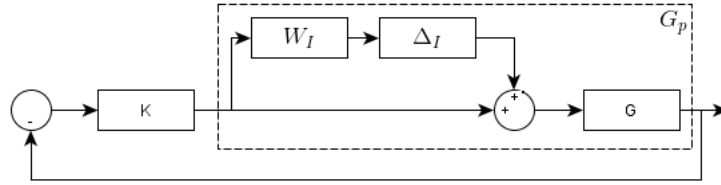


Figura 2.2: Planta com múltiplas incertezas.

## 2.3 Inversão Dinâmica Robusta - IDR

O objetivo desta seção é apresentar de uma forma simples o controle IDR (em inglês: Robust Multi Inversion, RMI) desenvolvido inicialmente em [Lav05]. Para tal, considera-se um sistema com uma entrada e uma saída (em inglês, single-Input single-output) de forma a simplificar a apresentação sobre o controle IDR.

### 2.3.1 Introdução

Existem poucos estudos contendo análise de estabilidade e síntese dentro do contexto de controle robusto não-linear. Dessa forma uma lei de controle não-linear robusta surge como uma alternativa interessante. Os elementos essenciais para a concepção de tal lei de controle consideram as propriedades de robustez da mesma frente às incertezas de parâmetros, erros de modelagem e as perturbações exôgenas. Assim, apesar da presença de incertezas ou perturbações, o sistema compensado deve satisfazer os critérios de estabilidade e de desempenho. Como mencionado na seção 2.2, as incertezas podem ser causadas por erros de aproximação, de modelagem, de variação dos parâmetros ou de problemas de precisão numérica. Todos esses fatores podem induzir desvios importantes entre o comportamento do sistema real e de seu modelo simulado. Afim de evitar ou pelo menos minimizar esses desvios, ao longo dos anos algumas técnicas agrupadas sob o nome de controle robusto foram implementadas [ZW98]. Nos últimos anos a vasta literatura sobre sistemas lineares evidencia que os estudos de controle robusto tem crescido, sendo distintas duas classes entre os métodos mais comumente abordados:

- A primeira classe é constituída pelos chamados métodos de frequência, em que a incerteza é considerada não-estruturada, ou seja, não há informação disponível sobre a estrutura do sistema. A robustez associada aparecem como uma extensão de conceitos clássicos tais como margem de ganho, margem de fase e funções de sensibilidade. Nessa classe podem ser citados como exemplos o controle LQR/LTR, a abordagem  $H_\infty$ , otimização  $H_2/H_\infty$  etc [ZW98].
- Uma segunda classe congrega as abordagens ditas temporais, em que as incertezas são estruturadas, ou seja, que afetam particularmente os parâmetros do sistema. Parte importante desses métodos podem envolver a teoria de Lyapunov, tais como a abordagem quadrática ou abordagens tipos polinomiais [Lav05].

Do ponto de vista de análise de estabilidade, os métodos permitem analisar a robustez do sistema corrigindo-o a certos critérios. Dentro desse contexto, há interesse de determinar os

limites das incertezas admissíveis tais que o sistema pode suportar sem degradar a estabilidade e o desempenho. Do ponto de vista da síntese do controlador, considerar as incertezas é importante para a busca da garantia de estabilidade na fase de concepção da lei de controle. Nesse contexto, os métodos de síntese multi-objetivo são comumente utilizados [Gon06].

Diante disso surge a proposta de um método de controle robusto, denominado de controle IDR (Inversão dinâmica robusta) apresentado em [Lav05]. Esse método baseia-se na inversão dinâmica e consiste em determinar uma lei de controle pela inversão do modelo modificado (controle linearizante). A principal característica dessa nova técnica é que ela tenta mitigar as incertezas de modelagem, o que confere ao sistema controlado uma propriedade de robustez, que não está presente nos métodos de controle não-linear clássicos. Tais incertezas podem ser representadas por incertezas limitadas em norma [Lav05].

### 2.3.2 Principais elementos sobre IDR

O princípio do controle IDR é de dar robustez à inversão da dinâmica e assim atender os requisitos de estabilidade e desempenho do sistema não-linear. O controle por IDR baseia-se na realimentação da diferença entre o sistema real e o modelo usado na linearização por realimentação sujeito ao mesmo sinal de controle. Quando há diferenças não nulas entre o sistema real e o modelo, um integrador com um ganho ajustável é utilizado para compensar o sinal de controle. No entanto, este pode impactar fortemente as margens de estabilidade do sistema, tornando problemático o seu comportamento em regime não-linear.

Na proposta de Lavergne [Lav05] pretende-se que o controle IDR faça uma inversão robusta da dinâmica, quando o modelo considerado é afetado por incertezas.

Seja um sistema dado por <sup>1</sup>:

$$\begin{aligned}\dot{x} &= A(x) + B(x)U \\ y &= Cx\end{aligned}\tag{2.74}$$

Esse sistema tem sua saída derivada  $d_k$  vezes até que o sinal de entrada  $U$  apareça na equação de saída pela primeira vez. Nesse caso pode-se obter:

$$y^{d_k} = \tilde{y} = F(x) + G(x)U\tag{2.75}$$

Em que  $d_k$  é o grau relativo da saída. A matriz  $G(x)$  precisa ter posto completo para que exista uma lei de controle linearizante por realimentação dada por.

$$U = G^{-1}(x)(v - F(x))\tag{2.76}$$

Assuma que exista  $G^{-1}(x)$  e substitua (2.76) em (2.75):

$$\tilde{y} = F(x) + G(x)G^{-1}(x)(v - F(x))\tag{2.77}$$

$$\tilde{y} = v\tag{2.78}$$

---

<sup>1</sup>Note que a dependência temporal é omitida.

Observe que para o uso do conceito de grau relativo é necessário que haja difeomorfismo e a inversa de  $G(x)$ . Satisfeita essas condições, a saída do sistema linearizado é dada por:

$$y = \underbrace{\int \dots \int}_{d_k} \tilde{y} dt = \underbrace{\int \dots \int}_{d_k} v dt \quad (2.79)$$

Portanto, basta escolher  $v$  como a derivada de ordem  $d_k$  de uma trajetória desejada,  $y_d$ , para que a saída  $y$  siga  $y_d$ .

$$v = y_d^{d_k} \quad (2.80)$$

Naturalmente, é necessário que a trajetória desejada seja suficientemente suave para evitar problemas com a(s) derivada(s). A figura 2.3 representa o sistema (2.75) com a lei de controle (2.76).

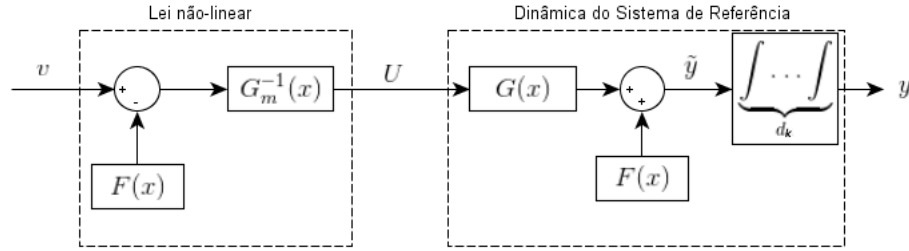


Figura 2.3: Linearização por Realimentação.

Entretanto o controle proposto pela lei (2.76) e ilustrado na figura 2.3 necessita de robustez com relação a presença de incertezas, isto é, quando o modelo utilizado na construção da lei de controle (2.76) não corresponde exatamente ao sistema (2.75). Neste caso, adota-se o índice “m” para denotar o modelo.

$$\tilde{y}_m = F_m(x) + G_m(x)U \quad (2.81)$$

Consequentemente a saída  $\tilde{y}_m$  deve corresponder à saída real do sistema  $\tilde{y}$  acrescida de um erro de modelagem,  $\Delta_m(x, U)$ . Assim, considerando que a saída real,  $\tilde{y}$ , é igual a saída do modelo,  $\tilde{y}_m$ , subtraída do erro de modelagem,  $\Delta_m(x, U)$ , temos

$$\tilde{y}_m = \tilde{y} + \Delta_m(x, U) \text{ ou } \tilde{y} = \tilde{y}_m - \Delta_m(x, U) \quad (2.82)$$

Utilizando (2.75) e (2.81), (2.82) pode ser reescrita como

$$\tilde{y}_m = F(x) + G(x)U + \Delta_m(x, U) \text{ ou } \tilde{y} = F(x) + G_m(x)U - \Delta_m(x, U) \quad (2.83)$$

respectivamente.

No caso em que  $\tilde{y}_m$  e  $\tilde{y}$  não coincidem, objetiva-se uma estratégia de compensação do sinal de controle aplicado ao sistema real,  $U$ , de forma a permitir a utilização do modelo no projeto de

controladores para o sistema (2.75). Para tal situação, a abordagem IDR apresenta-se como uma estratégia de controle bastante útil. A técnica IDR consiste em adicionar um termo de correção  $\Delta_{IDR}$  à lei de controle linearizante clássica, visando neutralizar os efeitos das diferenças entre  $G_m(x)$  e  $G(x)$  e entre  $F_m(x)$  e  $F(x)$ . Por efeito de clareza, a topologia do sistema da figura 2.3 é repetido na figura 2.4 com o acréscimo de incertezas de modelagem e do laço de controle linear.

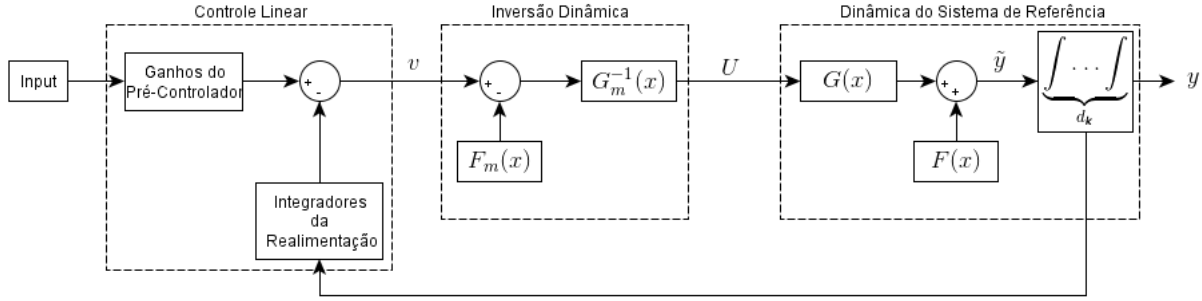


Figura 2.4: Linearização por Realimentação com ganho integral.

O princípio da técnica IDR foi desenvolvido em [Lav05] e a realimentação IDR pode ser definida por:

$$\Delta_{IDR} = K_{IDR} \int_0^t (\tilde{y}_{IDR} - \tilde{y}) d\tau \quad (2.84)$$

Em que  $\tilde{y}$  é obtido a partir da saída real do sistema e  $\tilde{y}_{IDR}$  é a saída estimada a partir do modelo nominal. Esse último sinal será detalhado mais adiante. Note que é possível investigar outras construções para o sinal  $\Delta_{IDR}$ , porém essa investigação não será feita nessa dissertação. Em [Lav05] é proposto que  $K_{IDR}$  seja uma matriz diagonal que deve ser sintetizada. Entretanto, não há nenhuma restrição matemática ou teórica que impeça o uso de uma matriz cheia como ganho IDR. Dessa forma, diferentemente de [Lav05] no presente trabalho o  $K_{IDR}$  é uma matriz cheia a ser sintetizada.

Uma vez que  $\tilde{y}$  é dado em (2.75), pode-se utilizar (2.83) para obter uma estimativa desse sinal a partir do modelo nominal e o sinal de compensação  $\Delta_{IDR}$ , veja (2.84) - conforme a seguinte expressão:

$$\tilde{y}_{IDR} = F(x) + G(x)U + \Delta_m(x, U) - \Delta_{IDR} \quad (2.85)$$

Conforme (2.78), deseja-se que  $\tilde{y}_{IDR}$  siga o sinal linear  $v$ :

$$\tilde{y}_{IDR} = v$$

Assim a compensação  $\Delta_{IDR}$  é definida por

$$\Delta_{IDR} = K_{IDR} \int_0^t (v - \tilde{Y}) d\tau \quad (2.86)$$

e o novo sinal de controle é modificado a partir de (2.76) para  $U = G_m^{-1}(x)(v - F_m(x) + \Delta_{IDR})$ . Importante salientar que  $\tilde{Y}$  é a  $d_k$ -ésima derivada de  $y$ , ou seja, as derivadas sucessivas da saída para que o sinal de controle apareça. Em suma o conjunto de equações apresentadas anteriormente para a abordagem quando aplicadas ao diagrama da figura 2.4, resultam no diagrama completo apresentado na figura 2.5.

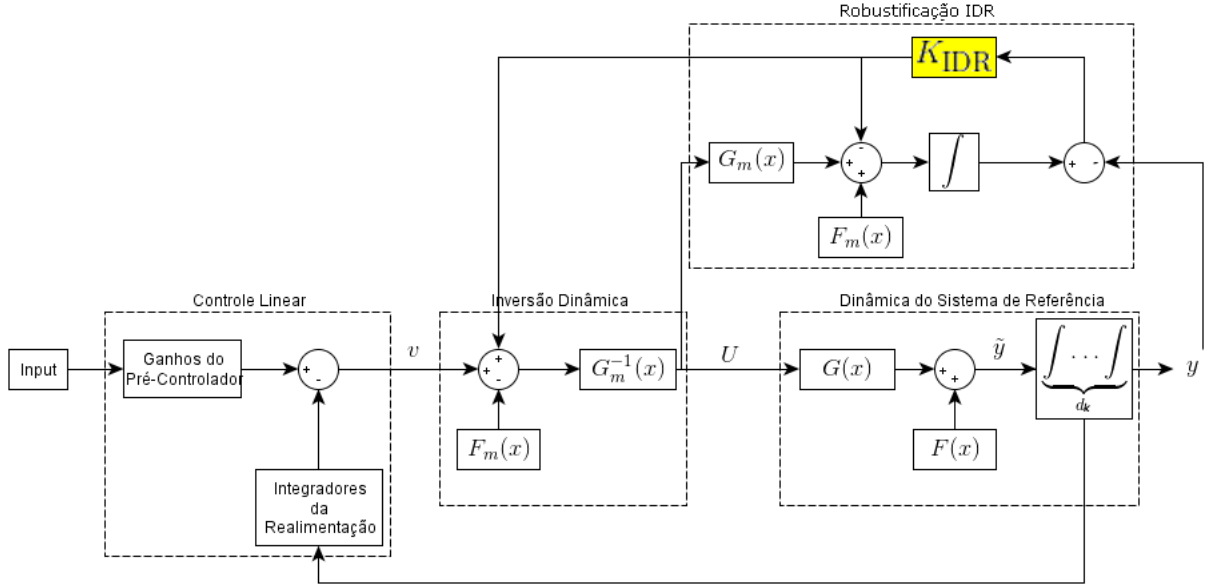


Figura 2.5: Abordagem IDR completa aplicada.

A lei de controle obtida com a compensação  $\Delta_{IDR}$ , pode ser reescrita da seguinte forma:

$$U = G_m(x)(v - F_m(x) + \underbrace{K_{IDR} \int \dots \int (\tilde{y}_{\Delta_c} - \tilde{y}) dt \dots dt}_{d_k}) \quad (2.87)$$

Em que  $\tilde{y}_{\Delta_c}$  é a saída desejada do sistema derivada  $d_k$  vezes. Nesse caso, não somente o grau de integração é superior à ordem necessária para o controle (perda de margens de estabilidade), mas também é sensível às não-linearidades uma vez que integra a diferença entre a saída do sistema e saída desejada.

### Aplicação ao Caso Padrão

Considera-se um sistema com um sinal de controle e uma saída, SISO, descrito por:

$$\dot{x} = Ax + B\gamma(x)(u - \alpha(x)) \quad (2.88)$$

com  $x \in \mathbb{R}^n, u \in \mathbb{R}^n$ . O par  $(A, B)$  é supostamente controlável e a função não-linear  $\gamma(x)$  é invertível para  $x \in D$ .

O sistema pode ser reescrito da seguinte forma,

$$\dot{x} = f(x) + g(x)u$$

com  $f(x) = Ax - B\gamma(x)\alpha(x)$  e  $g(x) = B\gamma(x)$ . O sinal linearizante é da seguinte forma.

$$u = \gamma(x)^{-1}v + \alpha(x) \quad (2.89)$$

que aplicado ao sistema (2.88), resulta no seguinte sistema não-linear:

$$\dot{x} = Ax + Bv \quad (2.90)$$

em que  $v$  é o comando linear que pode ser determinado a partir de técnicas de controle clássicas tais como alocação de pólos, controle robusto, LQR,  $H_2$ ,  $H_\infty$  etc. As matrizes  $A, B$  são definidas por:

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & . \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \\ 0 & 0 & 0 & \dots & 0 \end{bmatrix}; \quad B = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

A saída considerada é definida como:

$$y = Cx = \begin{bmatrix} 1 & 0 & 0 & \dots & 0 \end{bmatrix} x \quad (2.91)$$

Supondo que as incertezas de modelagem afetam o sistema, ou seja, as não-linearidades em  $\gamma(x)$  e  $\alpha(x)$  podem ser descritas como:

$$\gamma(x) = \gamma_n(x) + \Delta\gamma(x) \quad (2.92)$$

$$\alpha(x) = \alpha_n(x) + \Delta\alpha(x) \quad (2.93)$$

em que  $\gamma_n(x)$  e  $\alpha_n(x)$  são as partes nominais e  $\Delta\gamma(x)$  e  $\Delta\alpha(x)$  contém as incertezas. Assim o sistema pode ser escrito da seguinte forma:

$$\dot{x} = Ax + B(\gamma_n(x) + \Delta\gamma(x))(u - (\alpha_n(x) + \Delta\alpha(x))) \quad (2.94)$$

ou de maneira equivalente:

$$\dot{x} = Ax + B\gamma_n(x)(u - \alpha_n(x)) + B\Delta_s \quad (2.95)$$

com

$$\Delta_s = \Delta\gamma(x)(u - (\alpha_n(x) + \Delta\alpha(x))) - \gamma_n(x)\Delta\alpha(x) \quad (2.96)$$

É importante salientar que somente a parte nominal  $\gamma_n(x)$  é considerado invertível para todo  $x$  dentro do domínio de operação  $\mathbb{D}$ . Outro ponto importante, é que a incerteza  $\Delta_s$  é assumida delimitada para todo  $x$  dentro do domínio de operação  $\mathbb{D}$ .

Assim, devido à presença da incerteza sobre  $\gamma(x)$  e  $\alpha(x)$ , a lei de controle linearizante definida por (2.89) não resulta na linearização exata do sistema. Somente as partes nominais,  $\gamma_n(x)$  e  $\alpha_n(x)$  podem ser utilizadas pelo controle. Uma lei de controle possível é:

$$u = \gamma_n(x)^{-1}v + \alpha_n(x) \quad (2.97)$$

O sistema (2.95) realimentado com a lei linearizante (2.97), é

$$\dot{x} = Ax + Bv + D\Delta_s \quad (2.98)$$

em que  $\Delta_s$  é definido por (2.96), ou seja:

$$\Delta_s = \Delta\gamma(x)\gamma_n(x)^{-1}v - (\gamma_n(x) + \Delta\gamma(x))\Delta\alpha(x) \quad (2.99)$$

Afim de garantir as propriedades de robustez, então considera-se a adição da malha de realimentação *IDR*. Dessa forma o sinal de controle é redefinido à luz de (2.87) resultando em

$$u = \gamma_n(x)^{-1}(v + \Delta_{IDR}) + \alpha_n(x) \quad (2.100)$$

A nova variável  $\Delta_{IDR}$ , é dada por

$$\Delta_{IDR} = K_{IDR} \left( \int_0^t v d\tau - y \right) \quad (2.101)$$

em que  $K_{IDR}$  é um ganho à ser determinado e  $y$  é definido em (2.91). O sistema realimentado obtido aplicando-se a lei (2.100) e (2.101) no sistema (2.95), é então escrito da seguinte forma:

$$\dot{x} = (A - BK_{IDR}C)x + Bv + BK_{IDR} \int_0^t v d\tau + B\Delta_s \quad (2.102)$$

em que  $\Delta_s$  definido por (2.96)

$$\Delta_s = \Delta\gamma(x)\gamma_n(x)^{-1} + (v + \Delta_{IDR}) - (\gamma_n(x) + \Delta\gamma(x))\Delta\alpha(x) \quad (2.103)$$

O problema de se encontrar um resultado, pode ser formalizado da seguinte maneira:

Considerando o sistema (2.102), determine  $K_{IDR}$  e a lei de controle  $v$  tais que esse sistema (2.95) seja localmente assintoticamente estável.

Pode-se simplificar esse problema com a suposição que a lei de controle  $v$  é conhecida e que seu ganho *IDR*,  $K_{IDR}$ , é uma variável a ser determinada.

### 2.3.3 Comentários sobre a abordagem IDR

Com base na lei de controle com IDR apresentada em (2.87), a vantagem do IDR é ser uma abordagem de simples e fácil aplicação. A técnica consiste em se fazer a inversão dinâmica do sistema e uma comparação entre as saídas do sistema real com as saídas do sistema modelado. Assim, gerando um sinal de compensação a ser adicionado ao sinal de controle para atenuar as diferenças entre o sistema modelado usado no projeto da lei de linearização e dos sistema real a ser controlado. Esse sinal de compensação traz a grande vantagem de dar maior robustez em malha-fechada, reduzindo os efeitos negativos das incertezas. Outras considerações adicionais podem ser feitas sobre a abordagem IDR [Lav05]:

- O controle IDR não é um controle adaptativo. Sendo necessário fornecer um modelo de referência.

- Aumento da robustez do sistema frente a presença de incertezas com adição de um integrador de precisão à uma determinada lei de controle clássica.
- Necessidade de acesso ao valor do controle  $U$  e um compromisso entre a velocidade da malha IDR e erro do modelo.
- A utilização de perturbações singulares permite demonstrar a estabilidade local do esquema de controle não-linear proposto.

Dentre as vantagens da abordagem IDR, é que essa apresenta insensibilidade em relação às não-linearidades e dinâmica negligenciadas dos dispositivos de controle, robustez para erros de modelagem e simplicidade de implementação. Entretanto a técnica apresenta desvantagens, por exemplo, o IDR baseia-se em um observador de erro de modelagem de alto ganho.

## 2.4 Comentários Gerais

Nesse capítulo foi apresentada a técnica de linearização por realimentação clássica e as duas ferramentas possíveis: entrada-estado e entrada-saída. Na sequência do trabalho é utilizada a linearização por realimentação clássica, entrada-estado que é aplicada a um modelo não-linear. Em relação às incertezas, são apresentados dois tipos: paramétricas e de dinâmica não-modelada, a representação de ambas em um sistema com incertezas é denominada incertezas concentradas. As incertezas concentradas afetam o modelo a ser utilizado no capítulo 4 deste trabalho, contudo as incertezas de dinâmica não-modelada são inerentes a qualquer sistema. Consequentemente, as incertezas paramétricas é que serão manipuladas neste trabalho para utilização com a topologia de controle de compensação de incertezas, IDR. Como mencionado, o IDR será utilizado em conjunto com uma técnica de controle clássica aplicada ao modelo linearizado por realimentação clássica, entrada-estado, para a compensação de incertezas paramétricas via algoritmos apresentados no capítulo 3 em estudos de casos propostos no capítulo 4.



## Estratégias para Determinação do Ganho $K_{\text{IDR}}$

Neste capítulo é estudada a topologia apresentada seção 2.3 visando a proposição de estratégias para determinação do ganho referente à compensação do sinal de controle devido às diferenças entre o modelo nominal e o sistema real. Essas estratégias são formalizadas por duas técnicas para determinação do ganho  $K_{\text{IDR}}$ : heurística e meta-heurística, que se mostram eficientes na determinação do ganho do compensador. O algoritmo heurístico consiste em determinar um valor de  $K_{\text{IDR}}$  que reduza a norma do erro entre o comportamento do sistema não-linear controlado com incertezas em relação ao sistema nominal. Posteriormente é apresentada a meta-heurística, que diferentemente do algoritmo heurístico, que é projetado para a solução de um problema específico, a meta-heurística é uma estratégia de alto nível para solução de problemas de otimização de uma forma genérica sobre determinadas condições. A meta-heurística, aqui apresentada, é aplicada por meio do uso de um algoritmo genético com intuito de determinar o mesmo  $K_{\text{IDR}}$  com uma técnica diferente, entretanto com o mesmo propósito de minimização da norma do erro. Um estudo detalhado da implementação e uso dessas estratégias é apresentado com o objetivo de melhorar o desempenho do robô frente a presença de incertezas. Parte dos resultados aqui apresentados relativo a estratégia Heurística pode ser encontrado em [SLN14a] e [SLN14b].

### 3.1 Heurística

Nesta seção é apresentado uma heurística para determinação do valor de  $K_{\text{IDR}}$  que reduza o erro entre o comportamento do sistema não-linear controlado em relação aos valores nominais de seus parâmetros. Portanto, pretende-se buscar  $K_{\text{IDR}}$ , que atenda a equação (2.101), de maneira que a resposta do sistema real, considerando as incertezas  $\Delta_m(x, u)$  apresentadas no capítulo 2, se aproxime do comportamento esperado para o sistema nominal assumido na etapa de projeto da lei de linearização.

#### 3.1.1 Características

Na computação as teorias de algoritmos, combinações e otimização possuem duas propriedades na criação e concepção das mesmas:

- O algoritmo deve ter um tempo de execução aceitável.
- O algoritmo deve obter a solução ótima ou provavelmente boa para o problema em todos os casos.

Contudo, um algoritmo heurístico não precisa atender necessariamente à segunda propriedade, podendo ser um algoritmo que encontra boas soluções na maior parte dos casos, mas sem garantir que isso irá se repetir para todos os casos ou que deva sintetizar a solução da forma mais rápida e otimizada, podendo ser eficiente para alguns casos e ineficiente em outros [Wik14a].

Por muitos anos as heurísticas foram consideradas modelos cognitivos por excelência, sendo formadas de regras baseadas na experiência e no planejamento substituindo as anteriores baseadas na procura algorítmica que chega às soluções corretas depois de ter combinado o problema com um conjunto suficientemente grande das soluções possíveis. Os métodos heurísticos buscam a maior aderência possível de uma ação em relação a uma situação. Dessa forma, englobando estratégias, procedimentos, métodos de aproximação tentativa/erro, sempre na procura da melhor forma de chegar a um determinado fim. Muitas vezes, os processos heurísticos exigem menor tempo de processamento que algoritmos convencionais e determinísticos, sendo caracterizado por se aproximar da forma como o ser humano raciocina e chega às resoluções dos problemas, e garantem soluções eficientes. Pesquisas heurísticas são realizadas por meio da quantificação de proximidade a um determinado objetivo. Sendo qualificadas na característica de possuírem uma boa (ou alta) heurística se o objeto de avaliação está muito próximo do objetivo, ou ditas de má (ou baixa) heurística se o objeto avaliado estiver muito longe do objetivo. Etimologicamente a palavra heurística vem da palavra grega *Heuriskein*, que significa descobrir (e que deu origem também ao termo *Eureka*) [Per14].

Um algoritmo aproximativo (ou algoritmo de aproximação) é dito heurístico quando utiliza informação e intuição a respeito da instância do problema e da sua estrutura para resolvê-lo de forma rápida. Entretanto, nem todo algoritmo heurístico é aproximativo, ou seja, nem toda heurística tem uma razão de qualidade comprovada matematicamente ou prova formal de convergência. Por este motivo, em várias referências bibliográficas distingue-se os termos algoritmo aproximativo e heurística [Wik14a]:

- Aproximativo é a denominação do algoritmo que fornece soluções dentro de um limite de qualidade absoluto ou assintótico, assim como um limite assintótico polinomial de complexidade (pior caso) comprovado matematicamente;
- Heurística e método heurístico são denominações para o algoritmo que fornece soluções sem um limite formal de qualidade, tipicamente avaliado empiricamente em termos de complexidade (média) e qualidade das soluções.

### 3.1.2 O Algoritmo Heurístico

Nesse capítulo supõe-se que o ganho de realimentação estática de estados, como mencionado no capítulo 2, é pré-definido e não é modificado na busca por  $K_{IDR}$ . O algoritmo aqui apresentado é baseado em encontrar um ganho que verifica a menor norma do erro médio da saída do sistema com incertezas em relação à resposta do sistema não-linear nominal. O procedimento

proposto consiste inicialmente na construção de um politopo  $\mathcal{P}$  que contenha todas as matrizes do modelo gerado da linearização clássica, considerando todos os parâmetros que possuem incertezas. Esse politopo  $\mathcal{P}$ , com  $\kappa$  vértices, é usado para gerar uma sequência de conjuntos convexos indicada por  $\mathcal{P}_\delta = \delta\mathcal{P}$ , com  $\delta \in (0, 1]$ . Portanto, para os valores assumidos por  $\delta$ , é sempre verificado  $\mathcal{P}_\delta \subseteq \mathcal{P}$ . Na heurística proposta, deve-se selecionar um conjunto finito de valores crescentes de  $\delta \in (0, 1]$  de maneira a formar uma lista de  $\ell$  elementos dada por  $\Delta$ :

$$\Delta = \{\delta_1, \delta_2, \dots, \delta_\ell = 1\}, 0 < \delta_1 < \delta_2 < \dots < \delta_\ell = 1. \quad (3.1)$$

Além disso, o usuário deve definir um perfil de teste de interesse para o sinal de referência em que as simulações propostas no procedimento descrito a seguir serão executadas. Esse perfil de testes pode ser selecionado, por exemplo, em função do funcionamento típico do sistema a ser controlado.

Após a construção do politopo  $\mathcal{P}$  e de selecionado o conjunto  $\Delta$ , devem ser executadas 3 etapas:

1. Conjunto de erros e erro limitador
2. Busca do conjunto de ganhos  $K_{i,v}$  menores que o menor erro dentre todos os elementos,  $\underline{e}$ .
3. Determinação do ganho  $K_{\text{IDR}}$  menor que o erro médio máximo,  $\tilde{e}$ .

Essas etapas são detalhadas a seguir:

*Primeira etapa*

A busca pelo ganho  $K_{\text{IDR}}$  deve ser iniciada por  $\delta_1$  com  $K_0$  conhecido e que estabilize o sistema próximo da condição nominal. Note que esse ganho não é difícil de ser encontrado uma vez que para  $\delta_1$  suficientemente pequeno, sempre existe  $K_{\text{IDR}}$  próximo de zero que mantém o sistema estável. Isso é verificado, pois suficientemente próximo à condição nominal — portanto para  $\delta_1 > 0$  suficientemente pequeno — o sinal de compensação provido na topologia mostrada na figura 2.5,  $\Delta_{\text{IDR}}(t)$ , também aproxima-se de zero. Portanto uma escolha razoável é

$$K_0 = \epsilon I_p \quad (3.2)$$

em que  $\epsilon$  é um escalar real suficientemente pequeno e  $I_p$  é a matriz identidade.

Defina o sinal  $\bar{e}_{\delta_i,v}$  com uma equação tal  $e_{\delta_i,j}(t) = \tilde{Y}_j(t) - \tilde{Y}_{m,j}(t)$ . Para cada elemento  $\delta_i \in \Delta$ ,  $i = 1, \dots, \ell$ , deve-se calcular o erro médio em cada vértice do politopo  $\mathcal{P}_{\delta_i}$ ,  $\bar{e}_{\delta_i,v}$ ,  $v = 1, \dots, \kappa$ , que é dado pela seguinte expressão:

$$\bar{e}_{\delta_i,v} = \frac{1}{p} \sum_{j=1}^p \int_0^{t_f} |e_{\delta_i,j}(\tau)| d\tau \quad (3.3)$$

em que  $t_f$  é o tempo final de simulação arbitrado pelo usuário na definição do perfil do teste de interesse, para o  $v$ -ésimo vértice do politopo  $\mathcal{P}_{\delta_i}$ ,  $\delta_i \in \Delta$  e as saídas  $\tilde{Y}_j(t)$  e  $\tilde{Y}_{m,j}(t)$  são as  $j$ -ésimas saídas simuladas para um dado perfil de referência (veja figura 2.5) definido pelo usuário. É importante salientar que os valores obtidos para  $\tilde{Y}_j(t)$ ,  $j = 1, \dots, p$ , devem corresponder às saídas do sistema não-linear avaliado no vértice  $v$  do politopo  $\mathcal{P}_{\delta_i}$ . Com isso, é obtida uma sequência de erros dada por

$$E_{\delta_i} = \{\bar{e}_{\delta_i,1}, \bar{e}_{\delta_i,2}, \dots, \bar{e}_{\delta_i,\kappa}\}$$

que permite a seleção do menor erro dentre todos os elementos de  $E_{\delta_i}$ :

$$\underline{e} = \min_i E_{\delta_i} \quad (3.4)$$

O valor de  $\underline{e}$  servirá de limitador para a busca de melhores valores de  $K_{\text{IDR}}$ .

#### *Segunda etapa*

Para cada vértice  $v$  de cada politopo  $\mathcal{P}_{\delta_i}$ ,  $i = 1, \dots, \ell$ , busca-se um ganho  $K_{i,v}$  tal que (3.3) seja o menor possível e menor que  $\underline{e}$ . A busca por esse valor  $K_{i,v}$  deve ser feita a partir de  $K_0$  e pode ser implementada de diversas maneiras. Algumas possibilidades são *i)* uma busca simples incrementando os valores dos elementos de  $K_0$  ou *ii)* utilizar técnicas de busca como algoritmos de busca com discrepância limitada, heurística de construção etc. Portanto, para cada  $\delta_i$  e cada vértice  $v$ , seleciona-se ganho  $K_{i,v}$  que gera o menor erro médio  $\bar{e}_{\delta_i,v}$  que necessariamente satisfaz  $\bar{e}_{\delta_i,v} < \underline{e}$ .

Para cada  $\delta_i$  seleciona-se o ganho  $\tilde{K}_i = K_{i,v}$  que gera o menor erro médio  $\bar{e}_{\delta_i,v}$ . Para todos os vértices de todos os politopos  $\mathcal{P}_{\delta_i}$ , seleciona-se erro médio máximo, isto é,  $\tilde{e} = \max_{i,v} \bar{e}_{\delta_i,v}$ ,  $i = 1, \dots, \ell$ ,  $v = 1, \dots, \kappa$ .

#### *Terceira etapa*

Toma-se cada um dos ganhos  $\tilde{K}_i$  selecionados anteriormente e aplica-se esses ganhos em todos os demais vértices de todos os politopos  $\mathcal{P}_{\delta_i}$ . O ganho  $\tilde{K}_i$  é descartado sempre que gerar um erro médio maior que  $\tilde{e}$ . Se para todos os vértices de todos os politopos  $\mathcal{P}_{\delta_i}$  o ganho  $\tilde{K}_i$  gerar erros médios menores ou iguais a  $\tilde{e}$ , o ganho é classificado como viável e o menor erro verificado nessas avaliações é armazenado. Finalmente, dentre os ganhos selecionados, toma-se aquele que possuir o menor erro médio armazenado e faz-se  $K_{\text{IDR}}$  igual ao  $\tilde{K}_i$  correspondente. As etapas do algoritmo heurístico são apresentadas na forma do pseudo-código da figura 3.1.

### 3.1.3 Análise do Algoritmo Heurístico

Note que o procedimento proposto na seção 3.1.2 falha para os casos em que não seja encontrado um  $K_{i,v}$  que torne o vértice estável e com  $\bar{e}_{\delta_i,v} < \underline{e}$ , não sendo possível continuar a execução da busca. Na etapa 3 é possível que nenhum  $\tilde{K}_i$  encontrado seja estável para os demais vértices e/ou que atenda a relação  $\bar{e}_{\delta_i,v} < \tilde{e}$ , dessa forma nenhum  $K_{\text{IDR}}$  é determinado pelo algoritmo proposto.

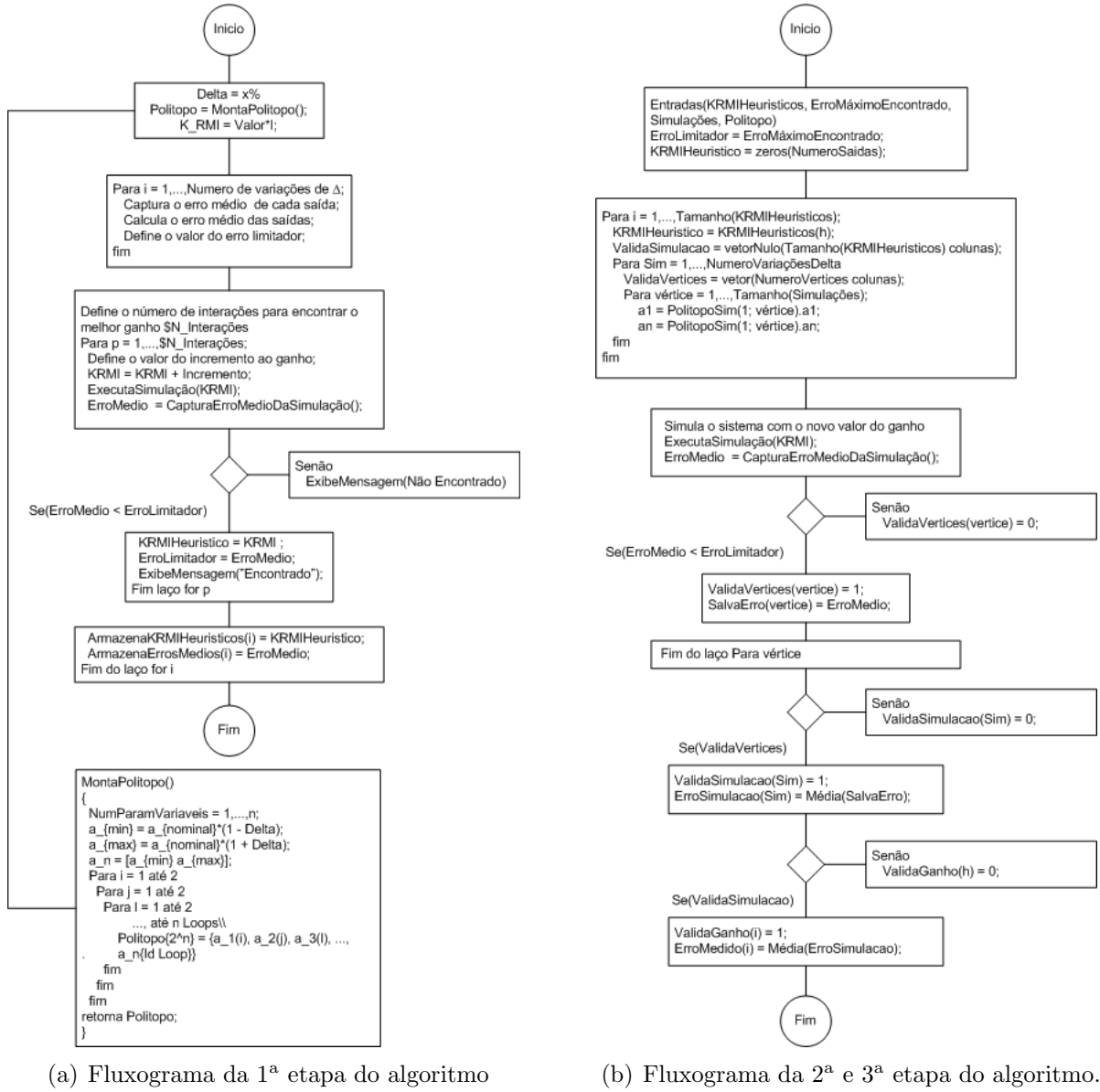


Figura 3.1: Fluxograma do algoritmo heurístico

## 3.2 Meta-Heurística

Essa seção apresenta uma abordagem semelhante à apresentada na seção 3.2 para determinação do ganho  $K_{IDR}$ . Entretanto não se trata de uma busca tradicionalmente heurística, mas de acordo com [Suc04] de uma meta-heurística evolutiva que utiliza um algoritmo genético de minimização do erro para determinação do  $K_{IDR}$  de menor erro encontrado. Meta-heurísticas são heurísticas genéricas mais sofisticadas, em que um procedimento simples é gerenciado por um procedimento que visa explorar de forma inteligente o problema e seu espaço de soluções. Da mesma forma que o algoritmo heurístico, o algoritmo genético busca encontrar o ganho  $K_{IDR}$  que resulte no menor erro em cada vértice definido pela equação (3.3).

### 3.2.1 Características

As técnicas de busca e otimização tradicionais iniciam-se com um único candidato que, iterativamente, é manipulado utilizando algumas heurísticas (estáticas) diretamente associadas ao problema a ser solucionado. Geralmente, esses processos heurísticos não são algorítmicos e sua simulação em computadores pode ser muito complexa. Apesar desses métodos não serem determinísticos, mas utilizam regras de transição probabilísticas, isso não implica que eles sejam inúteis. Na prática, eles são amplamente utilizados, com sucesso, em inúmeras aplicações: *simulated annealing*, busca tabu, etc. Por outro lado, as técnicas de computação evolucionária operam sobre uma população de candidatos em paralelo. Assim, elas podem fazer a busca em diferentes áreas do espaço de solução, alocando um número de membros apropriado para a busca em várias regiões. As meta-heurísticas evolutivas lidam com uma população, essencialmente um conjunto de soluções (possivelmente inviáveis) que evolui através da interação entre seus elementos, de forma a preservar as características pertinentes. Portanto, de forma a melhorar ao longo de suas gerações (interações) a qualidade média das soluções, não comprometendo a diversidade da população. Exemplos de algoritmos de meta-heurísticas evolutivas são os algoritmos genéticos, algoritmos meméticos, os algoritmos de estimação de distribuição e de busca dispersa [Suc04].

### 3.2.2 Algoritmo Genético para Determinação do Ganho $K_{IDR}$

Os Algoritmos Genéticos (AGs) diferem dos métodos tradicionais de busca e otimização, principalmente em quatro aspectos [Leo14]:

- AGs geralmente trabalham com uma codificação do conjunto de parâmetros e nem sempre com os próprios parâmetros.
- AGs trabalham com uma população e não com um único ponto.
- AGs utilizam informações de custo ou recompensa e não derivadas ou outro conhecimento auxiliar.
- AGs utilizam regras de transição probabilísticas e não determinísticas.

Algoritmos Genéticos são muito eficientes para busca de soluções ótimas, ou aproximadamente ótimas em uma grande variedade de problemas, pois não impõem muitas das limitações encontradas nos métodos de busca tradicionais. Além de ser uma estratégia de gerar-e-testar muito elegante, por serem baseados na evolução biológica, são capazes de identificar e explorar fatores ambientais e convergir para soluções ótimas, ou aproximadamente sub-ótima em níveis globais. Inicialmente, é gerada uma população formada por um conjunto aleatório de indivíduos que podem ser vistos como possíveis soluções do problema. Durante o processo evolutivo, esta população é avaliada: para cada indivíduo é dada uma nota, ou índice, refletindo sua habilidade de adaptação a determinado ambiente. Uma porcentagem dos mais adaptados são mantidos, enquanto os outros são descartados (darwinismo). Os membros mantidos pela seleção podem sofrer modificações em suas características fundamentais através de mutações e cruzamento (crossover) ou recombinação genética gerando descendentes para a próxima geração. Este

processo, chamado de reprodução, é repetido até que uma solução satisfatória seja encontrada [Luc02].

### 3.2.3 O Algoritmo Genético

Neste trabalho é utilizado na meta-heurística o algoritmo genético fornecido pelo `ToolBox Optimization` do Matlab [Mat14]. Esse `ToolBox` de otimização possui a função de minimização denominada `fminsearch`, cujo objetivo é encontrar o menor valor possível de retorno de uma função para uma dada família de entradas [Mat14]. A função a ser minimizada (Denominada `BuscaIDRGenetica`), cujo código fonte é apresentado no apêndice C.4.1, consiste em calcular o erro da simulação do modelo para o valor de  $K_{IDR}$  utilizado. De uma forma simplificada, as etapas do algoritmo utilizado em torno dessa função é apresentado a seguir.

1. Iniciar os parâmetros do modelo.
2. Determinar o Politopo de incertezas  $P$ .
3. Determinar o  $K_{IDR}$  inicial com um valor aleatório ou definido por experiência.
4. Executar a função de minimização `fminsearch` para minimizar a função `BuscaNDIGenet-ica` com o  $K_{IDR}$  inicial.
5. Se  $\bar{e}_{\delta_i, v} < \tilde{e}$  e interações  $< \text{MaxInterações}$ , volta a etapa anterior até que se encontre um erro menor. Senão o ganho selecionado é o último ganho que atende a condição  $\bar{e}_{\delta_i, v} < \tilde{e}$ .

### 3.2.4 Análise do Algoritmo Genético

No caso do algoritmo genético que utiliza Meta-Heurística, os valores iniciais utilizados para se determinar o  $K_{IDR}$  de menor erro encontrado são fundamentais no tempo de execução do algoritmo. Caso o valor inicial utilizado esteja muito distante do valor de menor erro, o algoritmo pode ser interrompido por atingir o número máximo de interações e o resultado encontrado para  $K_{IDR}$  esteja muito distante de um valor aceitável. Em contrapartida, poderia-se aumentar o número máximo de interações, mas de toda o tempo de busca continuaria muito alto. Ou seja, para que seja praticável o uso da Meta-Heurística é importante se determinar um bom  $K_{IDR}$  inicial. Entretanto isso não é muito difícil, pois para um  $\delta_1$  suficientemente pequeno, o quanto menor for possível determinar, possivelmente há um  $K_{IDR}$  próximo de zero que mantém o sistema estável.

As duas técnicas apresentadas são bastantes úteis para a determinação de  $K_{IDR}$  de forma prática e eficiente.

## 3.3 Comentários Gerais

Uma característica negativa das abordagens Heurística e Meta-Heurística é a não garantia de estabilidade robusta para o  $K_{IDR}$  determinado, mas são eficazes para um  $\mathcal{P}$  com incertezas precisamente conhecidas. A estabilidade é garantida a somente para os vértices de  $\mathcal{P}$  e alguns

pontos internos mas não para toda região do politopo. As duas estratégias apresentadas são bastante úteis para a determinação de  $K_{\text{IDR}}$ , pois são práticas e requerem menor rigor matemático do que a estratégia de restrição convexa por exemplo, mas sem perder eficiência.

O comparativo entre essas duas estratégias na determinação de  $K_{\text{IDR}}$ , aplicada à um modelo não-linear visando a compensação de incertezas de modelagem serão apresentadas no capítulo 4.



# Abordagem IDR aplicada ao modelo do robô móvel F180

Neste capítulo, os controladores obtidos a partir de técnicas discutidas no capítulo 3 e utilizando o comando `Place` do `Matlab` são aplicados ao modelo matemático desenvolvido em [Soa11] e apresentado no Apêndice A. Inicialmente realiza-se a linearização do modelo do robô omnidirecional utilizando a técnica de linearização clássica apresentada na seção 2.1. Em seguida é apresentada a topologia de controle com a inclusão da abordagem IDR proposta no capítulo 2. Posteriormente são adicionadas a descrição do sistema incertezas de modelagem, conforme discutido na seção 2.2. As estratégias de síntese propostas no capítulo 3 são utilizadas separadamente para sintetizar o ganho  $K_{IDR}$ . Finalizando, experimentos numéricos são apresentados para comparar o desempenho dos controladores e evidenciar as vantagens de utilização da abordagem IDR.

## 4.1 Introdução

O projeto de um controlador e de um compensador de incertezas IDR é concebido por etapas. A construção inicia-se pela linearização do modelo, seguida da definição do politopo de incertezas associado à inclusão de incertezas paramétricas. Concluída essa etapa, é possível utilizar um método de controle linear para projetar um controle para o sistema resultante da linearização. No entanto, é necessário definir a topologia de controle que atenda aos requisitos do projeto. Por se tratar do mesmo problema proposto em [Soa11], uma partida de futebol de robôs em que o objetivo é alcançar e simultaneamente desviar-se dos demais robôs, a topologia de controle será a mesma de [Soa11]: seguimento de referência com a inclusão de ação integral no ramo direto [Che99]. Essa topologia é denominada de rastreamento robusto, do inglês *Robust Tracking*, entretanto não significa garantia de estabilidade robusta. Definida a topologia de controle, a próxima etapa é a síntese dos controladores, determinando-se o ganho estático de realimentação de estados e o ganho do compensador IDR.

## 4.2 Linearização do modelo

Na seção 2.1 é apresentada a linearização clássica para linearização do sistema por meio da linearização entrada-estados. Ao utilizar a linearização por realimentação clássica, nenhuma robustez é garantida na presença de incertezas paramétricas. Pois o sistema linearizado na forma de Brunovsky pode ter um comportamento diferente do sistema original quando algum parâmetro está fora do valor nominal. Em [Soa11], foi adotada uma linearização robusta para se valer da robustez perante a variação dos parâmetros garantida nessa abordagem. Todavia o atual trabalho visa mostrar a eficiência do IDR frente as incertezas paramétricas, dessa forma adota-se a abordagem por realimentação clássica do capítulo 2 que é mais vulnerável as variações paramétricas e tende a apresentar respostas inferiores a da realimentação robusta utilizada em [Soa11].

Os ângulos da disposição das rodas do robô, são os mesmos adotados em [Soa11], ou seja,  $\phi_d = \phi_t = 45^\circ$  ( $\phi_d$  e  $\phi_t$  são definidos em [Soa11]). Substituindo esses ângulos no modelo não-linear dado pela equação (A.22), tem-se:

$$\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} -a_1 & a_2\dot{\theta} & 0 \\ -a_2\dot{\theta} & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} -b_1 & -b_1 & b_1 & b_1 \\ b_1 & -b_1 & -b_1 & b_1 \\ b_2 & b_2 & b_2 & b_2 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (4.1)$$

em que:

$$a_1 = 2 \frac{n^2(b_o R_a + K_t K_{em})}{R_a(r^2 m + 2n^2 I_o)} \quad (4.2)$$

$$a_2 = \frac{r^2 m}{r^2 m + 2n^2 I_o} \quad (4.3)$$

$$a_3 = 4 \frac{n^2 L^2 (b_o R_a + K_t K_{em})}{R_a(4n^2 I_o L^2 + r^2 I_w)} \quad (4.4)$$

$$b_1 = \frac{nr K_t V_{ref} r}{\sqrt{2} R_a (r^2 m + 2n^2 I_o)} \quad (4.5)$$

$$b_2 = \frac{n L K_t V_{ref} r}{R_a (I_w r^2 + 4 I_o n^2 L^2)} \quad (4.6)$$

A saída do sistema é definida como seus estados, então:

$$y = \underbrace{\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}}_{\lambda(x)} \quad (4.7)$$

O modelo dado por (4.1) não atende aos requisitos para que seja utilizado o método de linearização por realimentação de estados, porque o modelo não possui o mesmo número de

entradas e saídas. Dessa forma é necessário fazer uma transformação conforme abaixo

$$\underbrace{\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -a_1 & a_2\dot{\theta} & 0 \\ -a_2\dot{\theta} & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix}}_{f(x)} \underbrace{\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}}_{\dot{x}} + \underbrace{\begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix}}_{g(x)} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \quad (4.8)$$

em que:

$$\begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} = \begin{bmatrix} -1 & -1 & 1 & 1 \\ 1 & -1 & -1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (4.9)$$

Observe que a equação (4.9) encontra-se no formato  $\mu = A\rho$ , em que  $A \in \mathbb{R}^{3 \times 4}$ . Consequentemente há mais equações do que variáveis e  $\rho$  é subdeterminado, o que implica que muitas escolhas de  $\rho$  podem levar ao mesmo  $\mu$ . O trabalho atual herda a escolha de [Soa11] e utiliza a solução particular que minimiza a norma do vetor  $\rho$ ,  $\|\rho\|$ , :

$$\rho = A^T(AA^T)^{-1}\mu \quad (4.10)$$

Para um vetor de entradas  $[\mu_1 \ \mu_2 \ \mu_3]^T$ , a equação (4.10) é utilizada para encontrar o vetor de entradas  $[\rho_1 \ \rho_2 \ \rho_3 \ \rho_4]^T$  que possui a menor norma. Como  $\rho_i$  representa as tensões aplicadas aos motores do robô, o uso de um vetor que possui menor norma implica em menor consumo energético pelo mesmo. Feita a transformação, o novo sistema possui o mesmo número de entradas e de saídas, sendo possível ser submetido a linearização por realimentação. Esse sistema satisfaz as hipóteses **H1**, **H2** e **H3** apresentadas no capítulo 2, portanto existem funções escalares  $\lambda_1(x) = \dot{X}_m$ ,  $\lambda_2(x) = \dot{Y}_m$  e  $\lambda_3(x) = \dot{\theta}$ , definidas em uma vizinhança  $U$  de  $x_0$ , o grau relativo do sistema é  $r = 1$ . Assim o sistema dado por (4.11) é linearizável por realimentação conforme apresentado em [Isi85] e a saída definida por (4.7).

$$\underbrace{\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix}}_{\dot{x}} = \underbrace{\begin{bmatrix} -a_1 & a_2\dot{\theta} & 0 \\ -a_2\dot{\theta} & -a_1 & 0 \\ 0 & 0 & -a_3 \end{bmatrix}}_{f(x)} \underbrace{\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}}_{\dot{x}} + \underbrace{\begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix}}_{g(x)} \begin{bmatrix} \mu_1 \\ \mu_2 \\ \mu_3 \end{bmatrix} \quad (4.11)$$

Garantidas as condições necessárias e suficientes para a linearização por realimentação, o próximo passo é aplicação da técnica conforme apresentado na Seção 2.1. É aplicada a linearização clássica. O difeomorfismo  $T(x)$  definido por (2.33) aqui é denominado por convenção de  $\phi_c(x)$ , assim seja o difeomorfismo:

$$x_c = T(x) = \phi_c(x) \quad (4.12)$$

e a realimentação de estados

$$u_c(x, v_c) = \alpha_c(x) + \beta_c(x)v_c, \quad (4.13)$$

sendo que

$$\alpha_c(x) = \begin{bmatrix} (a_1\dot{X}_m - a_2\dot{Y}_m\dot{\theta})/b_1 \\ (a_1\dot{Y}_m + a_2\dot{X}_m\dot{\theta})/b_1 \\ (a_3\dot{\theta})/b_2 \end{bmatrix}, \quad (4.14)$$

$$\beta_c(x) = \begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix} e \quad (4.15)$$

$$\phi_c(x) = \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}. \quad (4.16)$$

Obtêm-se o sistema linearizado

$$\dot{x}_c = \underbrace{\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{A_c} x_c + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_{B_c} v_c. \quad (4.17)$$

O sistema (4.17) encontra-se na forma de Brunovsky, dada pela linearização por realimentação clássica. O formato de (4.17) é utilizado para o sistema nominal. Contudo para a associação do sistema com incertezas o formato desejado é o apresentado pela equação (2.88) que é o formato de sistema proposto no capítulo 3 para abordagem IDR. Dessa forma o sistema (4.11) é levado para o formato (2.88) resultando em:

$$\dot{x}_c = \underbrace{\begin{bmatrix} a_1 & 0 & 0 \\ 0 & a_1 & 0 \\ 0 & 0 & a_3 \end{bmatrix}}_A x_c + \underbrace{\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}}_B \underbrace{\begin{bmatrix} b_1 & 0 & 0 \\ 0 & b_1 & 0 \\ 0 & 0 & b_2 \end{bmatrix}}_{\gamma(x)} \left( \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} - \underbrace{\begin{bmatrix} (a_1 \dot{X}_m - a_2 \dot{Y}_m \dot{\theta})/b_1 \\ (a_1 \dot{Y}_m - a_2 \dot{X}_m \dot{\theta})/b_1 \\ (a_3 \dot{\theta})/b_2 \end{bmatrix}}_{\alpha(x)} \right) \quad (4.18)$$

### 4.3 Topologia de Controle

Concluída a etapa de projeto da linearização por realimentação, a próxima etapa é projetar o sinal de controle  $v$  que seja capaz de estabilizar o sistema linearizado. Isso pode ser feito por meio de ferramentas de controle clássicas de controle linear, tal qual o controle por realimentação apresentado em [Che99].

Considerando o sistema dado por (2.17) com o par  $(A_c, B_c)$  controlável, então a estrutura de controle apresentada na figura 4.1 pode ser utilizada para fazer o seguimento de referência, em que aplica-se a ação integral sobre o erro de seguimento, definido como  $\dot{x}_a(t) = r(t) - y(t)$ , sendo  $r(t)$  a referência a ser seguida [Che99].

Observe que a inclusão do integrador, conforme demonstrado na figura 4.1, aumenta a ordem do sistema, entretanto assegura erro nulo em regime permanente para uma entrada em degrau. O erro nulo em regime permanente é de suma importância no futebol de robôs, pois ao longo das partidas é requisitado aos robôs que interceptem a bola e realizem passes para outros robôs. Se houvesse erro residual em regime permanente essas ações não seriam possíveis. Essa topologia é semelhante a um controlador do tipo Proporcional Integral (PI). A representação espaço de

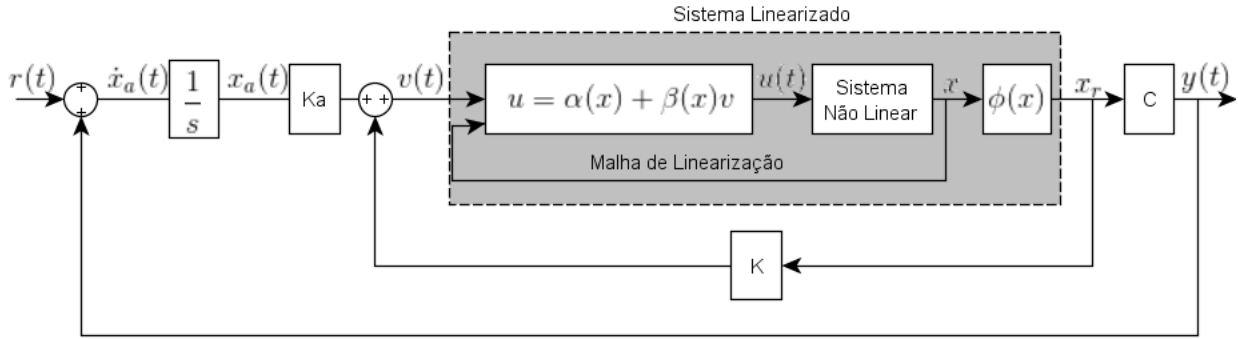


Figura 4.1: Projeto de sistemas de controle

estados dessa estrutura definida em [Che99] é dada por :

$$\underbrace{\begin{bmatrix} \dot{x}_c \\ \dot{x}_a \end{bmatrix}}_{\dot{\tilde{x}}} = \left( \underbrace{\begin{bmatrix} A_c & 0 \\ -C_c & 0 \end{bmatrix}}_{\tilde{A}} + \underbrace{\begin{bmatrix} B_c \\ 0 \end{bmatrix}}_{\tilde{B}} \underbrace{\begin{bmatrix} K & K_a \end{bmatrix}}_{\tilde{K}} \right) \underbrace{\begin{bmatrix} x_c \\ x_a \end{bmatrix}}_{\tilde{x}} + \underbrace{\begin{bmatrix} 0 \\ I \end{bmatrix}}_E r \quad (4.19)$$

em que  $K$  e  $K_a$  são os ganhos de realimentação de estados e os ganhos integrais, respectivamente. É possível alocar os polos de  $(\tilde{A} + \tilde{B}\tilde{K})$  utilizando alocação de polos ou outra ferramenta de sistemas lineares, no **Matlab** isso pode ser feito com a função **Place**.

Dentre os problemas da teoria de controle, um dos fundamentais é o projeto de leis de controle capazes de alocar os polos de malha fechada de um sistema em uma dada região. Dentre as diferentes formas para definição de especificações de desempenho, uma das mais populares é aquela que impõem alguma restrição para a alocação dos polos do sistema em malha fechada, como tempo de acomodação ou máxima sobrepassagem [CG99]. De uma maneira geral a maior parte da literatura relativa à alocação de polos, foca no problema da alocação exata dos mesmos, no qual é necessário que os polos de malha fechada estejam posicionados sobre ou próximos a locais previstos. Na prática a alocação exata não é requerida e é frequentemente suficiente sua alocação em uma região prescrita no lado esquerdo do plano complexo para sistemas contínuos no tempo ou no interior do círculo unitário para sistemas discretos no tempo [HB92]. O presente trabalho segue a abordagem de [Soa11] em que o problema de alocação de polos é considerado para uma região circular conforme a figura 4.2.

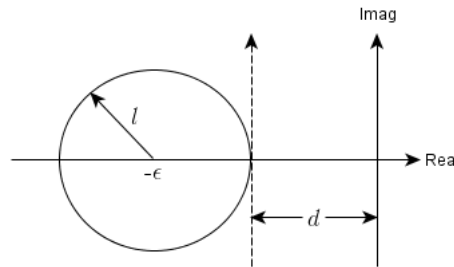


Figura 4.2: Região circular para alocação de polos

As exponenciais com decaimento limitado por  $Ke^{(\ell-\epsilon)t}$  e as frequências (parte imaginária dos polos) menores ou iguais a  $l$  limitam a oscilação e sobredinâmica, isso é garantido pela especificação dos polos para o sistema em malha fechada. No caso de sistema de segunda ordem, a região de alocação garante um fator de amortecimento (no pior caso) dado pelo cosseno do ângulo determinado pela reta que tangencia o círculo e passa pela origem, e máxima frequência natural amortecida limitada pelo raio do círculo. É importante observar que esses valores de pior caso não ocorrem simultaneamente.

Dessa forma o projeto de controle resume-se então, em calcular a matriz  $\tilde{K} = [K \ K_a]$  dada pela equação 4.19, tal que o sistema realimentado dado por

$$\dot{\tilde{x}} = (\tilde{A} + \tilde{B}\tilde{K})\tilde{x} + Er \quad (4.20)$$

possua todos os polos de malha fechada no interior de uma região circular de raio  $l$  e o centro em  $(-\epsilon, 0)$ , conforme mostrado na figura 4.2.

Um dos objetivos deste trabalho é utilizar o controlador IDR para compensação de incertezas de modelagem, dessa forma a topologia apresentada na figura 4.1, que não possui essa compensação, pode ser modificada para a inclusão da abordagem IDR. Essa modificação da topologia é apresentada na figura 4.3, em que a inclusão do compensador IDR, conforme mencionado no capítulo 3, visa gerar um sinal de compensação para as diferenças entre o sistema real e o modelado.

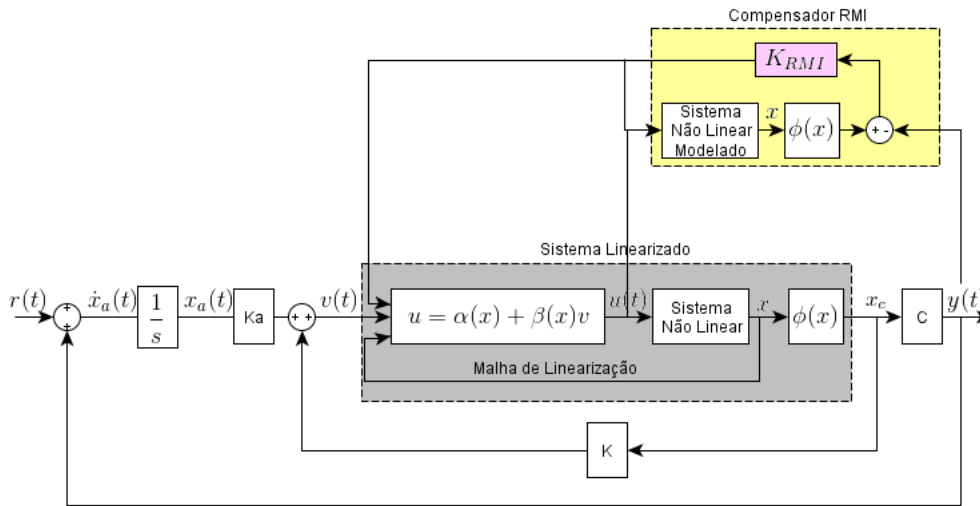


Figura 4.3: Topologia de controle de seguimento com compensação de incertezas

## 4.4 Experimentos Numéricos

Nesta seção são apresentados os experimentos numéricos realizados utilizando a matriz  $A_c$  dada pelo modelo linearizado pela técnica de realimentação linearizante clássica (4.18) substituída no sistema apresentado na equação (4.19). Os parâmetros nominais para esse modelo extraídos de [Soa11], são apresentados na tabela 4.1

Parâmetro	Valor	Descrição
$k_t$	0,0059	Constante de torque dos atuadores
$n$	19	Fator de redução dos atuadores para as rodas
$V_{ref}$	12	Tensão de referência
$R_a$	1,71	Resistência de armadura dos atuadores
$r$	0,024	Raio da roda
$J_0$	$3,88 \times 10^{-7}$	Momento de inércia dos atuadores
$J_w$	2,125	Momento de inércia do robô
$b_0$	$2,4 \times 10^{-6}$	Constante de atrito viscoso dos atuadores
$K_{em}$	0,0059	Constante de força contra eletromotriz dos atuadores
$L$	0,09	Distância entre a roda e o centro do robô
$m$	3,4	Massa do robô
$\phi_d$	45°	Ângulo dianteiro do robô
$\phi_t$	45°	Ângulo traseiro do robô

Tabela 4.1: Parâmetros nominais do modelo dinâmico

Após a substituição dos parâmetros, as matrizes  $\tilde{A}$  e  $\tilde{B}$  do modelo nominal aumentado para o robô F180 são dadas por (4.21).

$$\tilde{A} = \begin{bmatrix} -7,3398 & 0,8749\dot{\theta} & 0 & 0 & 0 & 0 \\ -0,8749\theta & -7,3398 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0,2167 & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 & 0 & 0 \end{bmatrix} \quad \tilde{B} = \begin{bmatrix} 5,9638 & 0 & 0 \\ 0 & 5,9638 & 0 \\ 0 & 0 & 1,3831 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (4.21)$$

A partir desse ponto são aplicadas as estratégias para determinação de  $K_{IDR}$  propostas no capítulo 3 para o sistema (4.21) e as incertezas associadas ao mesmo.

#### 4.4.1 Sistema com Incertezas

Em [Soa11] as incertezas foram associadas somente aos parâmetros lineares do modelo:  $(K_t, K_{em}$  e  $b_0)$  com uma variação máxima e mínima de  $\pm 20\%$  do valor nominal. Isso gera um politopo com menor número de vértices e de comportamento do sistema mais próximo do nominal. Entretanto, no mundo real, as incertezas podem estar associadas a qualquer parâmetro do modelo e com variações muito superiores a 20%. Por tal motivo, o presente trabalho adota uma bordagem diferente de [Soa11], incluindo incertezas de modelagem a todos os parâmetros (lineares e não-lineares) do modelo, os parâmetros são apresentados na tabela 4.1.

Contudo essa abordagem gera um politopo de  $2(2^n)$  vértices, em que  $n$  é o número de parâmetros, ou seja, 16.384 vértices, algo extremamente complexo de ser aplicado a um caso de estudo. Com intuito de reduzir o número de vértices do politopo, os parâmetros são variados de forma indireta. Os elementos  $a_1, a_2, a_3, b_1$  e  $b_2$  das equações (4.2), (4.3), (4.4), (4.5) e (4.6) respectivamente, são funções dos parâmetros da tabela 4.1. Assim, qualquer variação desses parâmetros tem que ser resultante da variação dos parâmetros do modelo. Dessa forma os elementos  $a_1, a_2, a_3, b_1$  e  $b_2$  serão utilizados para gerar o politopo de incertezas, acrescido do estado

$\dot{\theta}$  que também faz parte da matriz de dinâmica do sistema, (4.19), isso reduz consideravelmente o número de vértices do politopo. Complementando, o percentual de variação desses elementos não será de  $\pm 20\%$ , mas sim dos valores máximos e mínimos possíveis de  $a_1, a_2$  e  $a_3$ ,  $\dot{\theta}$  dentro de uma faixa pré-definida, e os elementos  $b_1$  e  $b_2$  não possuem incertezas associadas. Para se determinar os valores máximos e mínimos de  $a_1, a_2$  e  $a_3$  foi desenvolvido um algoritmo de busca, veja C.1, que varia todas as combinações possíveis dos parâmetros da tabela 4.1 entre valores máximo, mínimo e nominal de forma a determinar a combinação que resulte nos máximos e mínimos de  $a_1, a_2$  e  $a_3$ . Os elementos  $a_1, a_2, a_3, \dot{\theta}, b_1$  e  $b_2$  após o processo de inclusão de incertezas de modelagem são apresentados na tabela 4.2.

Tabela 4.2: Máximos e mínimos dos parâmetros

Valor	$a_1$	$a_2$	$a_3$	$\dot{\theta}$
Nominal	7,3398	0,8749	0,2167	0°
Min	1,640326	0,674413	0,028910	-30°
Max	31,001811	0,959340	1,511401	30°
Valor	$b_1$	$b_2$	—	—
Nominal	5,963798	1,383107	—	—

Realizando as combinações dos valores da tabela 4.2 e aplicando ao sistema (4.21), é gerado o politopo de incertezas formado por 16 vértices dado pelas matrizes

$$\tilde{A}_1 = \begin{bmatrix} 1,6403 & -20,2324 & 0 \\ 20,2324 & 1,6403 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_2 = \begin{bmatrix} 1,6403 & -20,2324 & 0 \\ 20,2324 & 1,6403 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.22)$$

$$\tilde{A}_3 = \begin{bmatrix} 1,6403 & -28,7802 & 0 \\ 28,7802 & 1,6403 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_4 = \begin{bmatrix} 1,6403 & -28,7802 & 0 \\ 28,7802 & 1,6403 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.23)$$

$$\tilde{A}_5 = \begin{bmatrix} 31,0018 & -20,2324 & 0 \\ 20,2324 & 31,0018 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_6 = \begin{bmatrix} 31,0018 & -20,2324 & 0 \\ 20,2324 & 31,0018 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.24)$$

$$\tilde{A}_7 = \begin{bmatrix} 31,0018 & -28,7802 & 0 \\ 28,7802 & 31,0018 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_8 = \begin{bmatrix} 31,0018 & -28,7802 & 0 \\ 28,7802 & 31,0018 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.25)$$

$$\tilde{A}_9 = \begin{bmatrix} 1,6403 & 20,2324 & 0 \\ -20,2324 & 1,6403 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_{10} = \begin{bmatrix} 1,6403 & 20,2324 & 0 \\ -20,2324 & 1,6403 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.26)$$

$$\tilde{A}_{11} = \begin{bmatrix} 1,6403 & 28,7802 & 0 \\ -28,7802 & 1,6403 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_{12} = \begin{bmatrix} 1,6403 & 28,7802 & 0 \\ -28,7802 & 1,6403 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.27)$$



$$\tilde{A}_{13} = \begin{bmatrix} 31,0018 & 20,2324 & 0 \\ -20,2324 & 31,0018 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_{14} = \begin{bmatrix} 31,0018 & 20,2324 & 0 \\ -20,2324 & 31,0018 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.28)$$

$$\tilde{A}_{15} = \begin{bmatrix} 31,0018 & 28,7802 & 0 \\ -28,7802 & 31,0018 & 0 \\ 0 & 0 & 0,0289 \end{bmatrix} \quad \tilde{A}_{16} = \begin{bmatrix} 31,0018 & 28,7802 & 0 \\ -28,7802 & 31,0018 & 0 \\ 0 & 0 & 1,5114 \end{bmatrix} \quad (4.29)$$

Tais vértices formam o polítopo de incertezas  $\mathcal{A}$ . O próximo passo é selecionar controladores com desempenho semelhantes ao caso nominal e avaliar o desempenho na presença de incertezas paramétricas.

#### 4.4.2 Síntese dos Controladores

Nesta seção, são apresentados os controladores obtidos pela utilização do algoritmo heurístico proposto na seção 3.1.2 e do algoritmo genético proposto na seção 3.2, ambos propostos no capítulo 3. Contudo como referência de comparação dos resultados entre os controladores, é realizada a síntese de um controlador utilizando a técnica de alocação exata de polos fornecida pelo comando `Place` do `MATLAB`. Importante salientar que o sistema possui polos lentos próximos a origem e é instável para os polos em 0,  $-7.3398$  e  $-0.2167$ . Diferente do proposto em [Soa11], todos os polos serão alocados em malha fechada sobre o eixo real no semi-plano esquerdo com intuito de garantir uma dinâmica rápida o suficiente. O conjunto de polos escolhidos é definido como

$$Pd = [-70 \quad -75 \quad -80 \quad -85 \quad -90 \quad -95] \quad (4.30)$$

O ganho fornecido pela alocação exata gerado pelo `Place` ao sistema expandido é dado por (4.31).

$$K_p = \begin{bmatrix} 31.0205 & 0 & 0 & -1433.6502 & 0 & 0 \\ 0 & 25.9901 & 0 & 0 & -1006.0703 & 0 \\ 0 & 0 & 112.0665 & 0 & 0 & -4301.9081 \end{bmatrix} \quad (4.31)$$

Observe que a matriz de ganho dado por (4.31) é obtido para os polos do sistema nominal, contudo a inclusão de incertezas faz com que os polos de dos vértices de  $\mathcal{A}$  sejam diferentes. Consequentemente, (4.31) não é capaz de realizar a compensação de incertezas e garantir a estabilidade do sistema para  $\mathcal{A}$ . Essa característica faz com que a resposta do sistema em  $\mathcal{A}$  utilizando o ganho (4.31) seja pior que a resposta do sistema nominal. A matriz de ganhos (4.31) representa as matrizes de ganhos  $K_a$  e  $K$  presentes nas figuras 4.1 e 4.3, ou seja,  $K_p = [K \quad K_a]$ .

#### Topologia IDR para compensação de incertezas

Frente ao problema das incertezas presentes em  $\mathcal{A}$ , como mencionado, a topologia apresentada na figura 4.1 foi adaptada com base no proposto na seção 2.3 para a topologia apresentada na figura 4.3. Essa topologia deve ser capaz de compensar as incertezas presentes em  $\mathcal{A}$  por meio de um sinal de controle  $v$  condicionado ao erro das incertezas que é adicionado ao sinal de controle  $u$  associado aos ganhos de (4.31). Para que o compensador IDR atenuar as incertezas de  $\mathcal{A}$  de forma que para cada vértice o sistema tenha resposta o mais próximo da resposta

nominal é necessário determinar um conjunto de ganhos  $K_p$  e  $K_{IDR}$  para tal. Contudo a ideia da proposta atual é que, para uma planta previamente projetada para um dado  $K_p$ , somente um  $K_{IDR}$  seja projetado, sem modificar o ganho  $K_p$ . Isso pode ser executado com a utilização dos algoritmos heurísticos e meta-heurísticos propostos no capítulo 4.

O algoritmo Heurístico irá fazer uma busca pela matriz de ganho  $K_{IDR}$  que apresente o menor erro definido por (3.3) para  $\mathcal{A}$ . O ganho  $K_{IDR}$  obtido do algoritmo heurístico para  $\mathcal{A}$  é definido por

$$K_{IDR\text{Heuristico}} = \begin{bmatrix} 0.0486 & -0.0068 & -0.0065 \\ -0.0068 & 0.0485 & -0.0065 \\ -0.0454 & -0.0455 & 0.2694 \end{bmatrix} \quad (4.32)$$

No mesmo sentido, o algoritmo genético realiza a busca pelo ganho  $K_{IDR}$  que apresente o menor erro definido por (3.3) para  $\mathcal{A}$ . O Ganho encontrado pela busca genética é dado por

$$K_{IDR\text{Genetico}} = \begin{bmatrix} 0.0472 & -0.0084 & -0.0079 \\ -0.0084 & 0.0470 & -0.00079 \\ -0.0554 & -0.0584 & 0.3175 \end{bmatrix} \quad (4.33)$$

Com os ganhos das três abordagens definidos, as respostas dos estados  $\dot{X}_m$ ,  $\dot{Y}_m$  e  $\dot{\theta}$  ao degrau unitário referentes ao Place, Heuristica e Genético são apresentados respectivamente pela figura 4.4.

Observa-se que pela resposta ao degrau, os estados dos três controladores possuem comportamento aproximado. Outros perfis de teste são apresentados na seção 4.4.4.

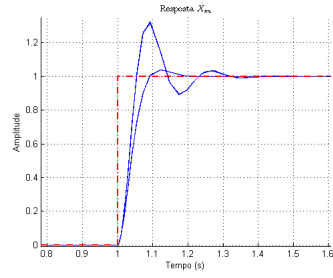
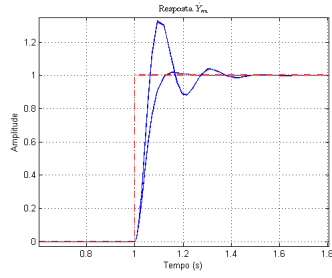
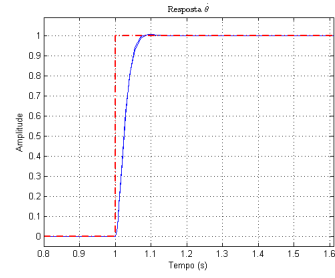
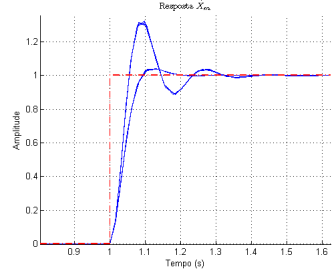
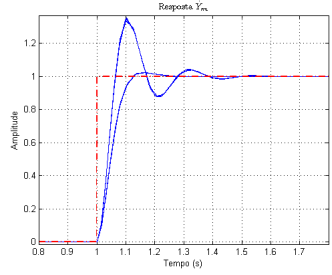
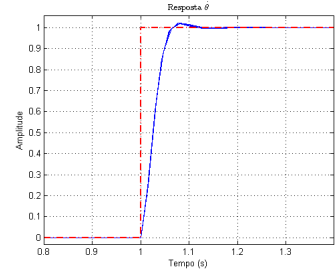
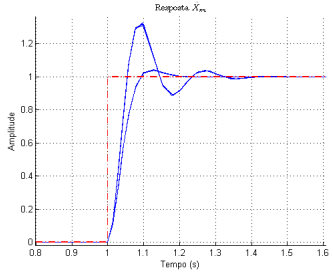
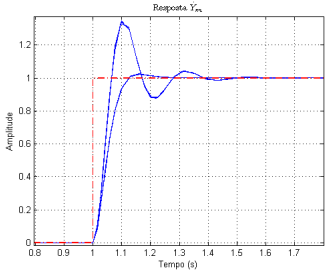
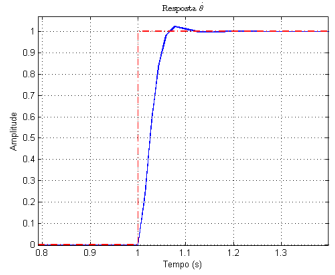
(a) Resposta  $\dot{X}_m$  ao degrau (Place)(b) Resposta  $\dot{Y}_m$  ao degrau (Place)(c) Resposta  $\dot{\theta}$  ao degrau (Place)(d) Resposta  $\dot{X}_m$  ao degrau (Heurístico)(e) Resposta  $\dot{Y}_m$  ao degrau (Heurístico)(f) Resposta  $\dot{\theta}$  ao degrau (Heurístico)(g) Resposta  $\dot{X}_m$  ao degrau (Genético)(h) Resposta  $\dot{Y}_m$  ao degrau (Genético)(i) Resposta  $\dot{\theta}$  ao degrau (Genético)

Figura 4.4: Respostas temporal dos estados  $\dot{X}_m$ ,  $\dot{Y}_m$  e  $\dot{\theta}$  para uma entrada em degrau unitário aplicada em  $r$  para todos os vértices de  $\mathcal{A}$  para o controlador usando os três ganhos (4.31), (4.32) e (4.33) respectivamente.

### 4.4.3 Critérios de desempenho

Concluída a análise das respostas temporais do sistema em malha fechada utilizando os controladores propostos, são apresentados alguns estudos de caso para o seguimento de referência. O objetivo desses estudos é demonstrar a diferença de desempenho dos controladores na presença de incertezas paramétricas. Um dos critérios de desempenho adotado nesse trabalho é o valor máximo absoluto do erro de seguimento das velocidades passadas como referência para o sistema.

$$|e_v| = \sqrt{e_{vx}^2 + e_{vy}^2 + e_{\dot{\theta}}^2} \quad (4.34)$$

em que  $e_{vx}$ ,  $e_{vy}$  e  $e_{\dot{\theta}}$  são dados por (3.3) para cada velocidade. O erro definido por (4.34) é utilizado como métrica para se determinar os desvios em relação a referência de cada controlador com os ganhos dados por (4.31), (4.32) e (4.33). Como critério de comparação de desempenho

entre os ganhos dados por (4.32), (4.33) em relação ao ganho (4.31) (sem IDR), é utilizada uma normalização em função do controlador sem IDR (4.31). Dessa forma o erro dos demais controladores é dado por (4.35)

$$e_p = \left( \frac{e_c}{e_{cp}} \right) * 100 \quad (4.35)$$

Em que  $e_c$  é o erro do controlador usando (4.32) ou (4.33),  $e_{cp}$  é o erro do controlador dado por **Place** e  $e_p$  é o erro percentual relativo do controlador.

O segundo critério de desempenho adotado neste trabalho também baseia-se no erro em relação à referência, contudo em relação a posição  $x$  e  $y$  do robô no campo de jogo. O desvio de posição em centímetros é dado por

$$e_{xy} = p_{ref} - e_{xyc} \quad (4.36)$$

em que  $p_{ref}$  é a posição de referência e  $e_{xyc}$ , que é o erro de posição do controlador, é dado por

$$|e_{vc}| = \sqrt{e_x^2 + e_y^2} \quad (4.37)$$

sendo  $e_x$  e  $e_y$  dados pela equação (3.3) para cada posição em  $x$  e  $y$ . O erro definido por (4.36) também é utilizado como métrica para se determinar os desvios em relação a referência de cada controlador com os ganhos dados por (4.31), (4.32) e (4.33). O terceiro critério utilizado neste trabalho é referente a capacidade do robô em atingir a bola em uma trajetória desejada. Considera-se que a bola foi atingida somente se a seguinte condição for alcançada:

$$e_{xy} \leq (r_{bola} * 0,65) \quad (4.38)$$

em que  $r_{bola}$  é o raio da bola.

#### 4.4.4 Estudos de Caso

O robô omnidirecional tem por capacidade realizar movimentos de rotação e translação em torno do seu centro de gravidade independentemente no plano bi-direcional. Essa propriedade é chamada de holonômica e para a comprovar essa característica trajetórias serão definidas realizando movimentos de translação e sem translação com rotação simultânea. Como prerrogativa assume-se que o valor inicial do sistema em coordenadas móvel  $S_m$  seja igual ao sistema de coordenadas global  $S_r$  e o valor inicial das variáveis de estado é dado por  $\tilde{x} = [0 \ 0 \ 0 \ 0 \ 0 \ 0]^T$ . Note que uma combinação de valores intermediários aos da tabela 4.2 é possível para as incertezas paramétricas do sistema, gerando infinitas combinações. Contudo de forma a simplificar a abordagem e reduzir consideravelmente o número de simulações somente os vértices do politopo de incertezas,  $\mathcal{A}$ , são utilizados.

Serão propostos oito estudos de casos para quatro trajetórias distintas, com intuito de comparar as abordagens para determinação de ganho IDR. Os estudos de casos propostos são: Trajetória reta, quadrada, circular e em oito, em que cada trajetória será testada com e sem rotação. Para as trajetórias reta, quadrada, circular e em oito sem rotação serão apresentados os gráficos de respostas da trajetória, o gráfico de erros para a simulação de todos os vértices, e a tabela comparativa de erros. A trajetória em oito com rotação será o caso de estudo mais explorado,

em que serão apresentados os gráficos de respostas da trajetória, as respostas de cada estado, gráfico de erros e a tabela comparativa de erros. Entretanto os mesmos resultados apresentados para a trajetória em oito com rotação são apresentados no apêndice D.1 para os demais estudos de caso.

### Caso 1 - Trajetória Reta

No estudo de caso da trajetória reta, é pré-suposto que o robô esteja se movimentando em linha reta na direção  $\psi = 45^\circ$ . Para o ângulo de rotação são assumidas duas situações, a primeira consiste em  $\theta = 0^\circ$  durante o intervalo de 10 segundos, ou seja, o robô realiza uma trajetória reta sem rotação. A segunda situação consiste em um  $\theta$  igual a zero no intervalo de 0 a 2 segundos, e  $\theta = 135^\circ$  no intervalo de 2 a 10 segundos, caracterizando a trajetória reta com rotação. As velocidades nas direções  $x$  e  $y$  são definidas em  $0,3m/s$ , em que cada componente de velocidade é definida como uma onda quadrada de forma a exercitar melhor a dinâmica do sistema.

As figuras (4.5(a)) e (4.5(b)) representam o gráfico da trajetória sem rotação e com rotação respectivamente e as figuras (4.5(c)) e (4.5(d)) são os gráficos dos erros de cada abordagem no respectivos vértice, sem rotação e com rotação. Os demais gráficos referentes a este estudo de caso estão presentes no apêndice D.1.

As tabelas 4.3 e 4.4 representam os erros de cada simulação, trajetória reta com rotação e trajetória reta sem rotação respectivamente.

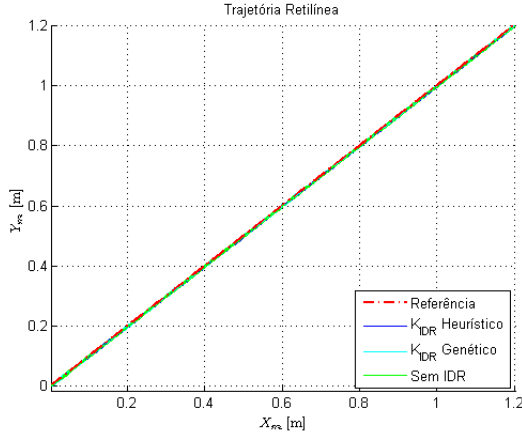
Tabela 4.3: Erro para trajetória reta sem rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	<b>92,177588</b>	92,362222
Erro bruto	$ e_v $	2,539526	2,340874	2,345562
	$e_{vx}$	0,030563	0,023206	0,023304
	$e_{vy}$	0,034737	0,025461	0,031360
	$e_{\dot{\theta}}$	2,539104	2,340620	2,345237

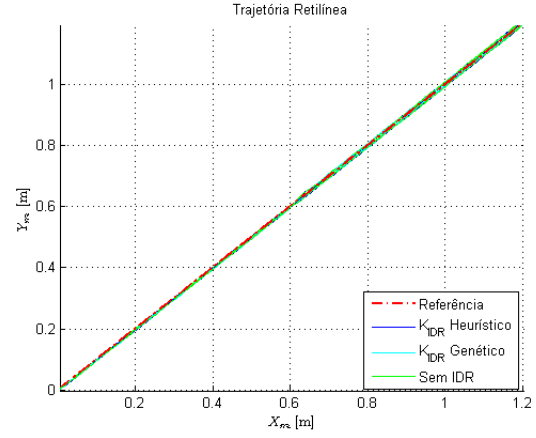
Tabela 4.4: Erro para trajetória reta com rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	<b>102,383093</b>	103,171453
Erro bruto	$ e_v $	1,086148	1,112032	1,120595
	$e_{vx}$	0,026957	0,027190	0,026939
	$e_{vy}$	0,030483	0,030847	0,030480
	$e_{\dot{\theta}}$	1,085386	1,111272	1,119857

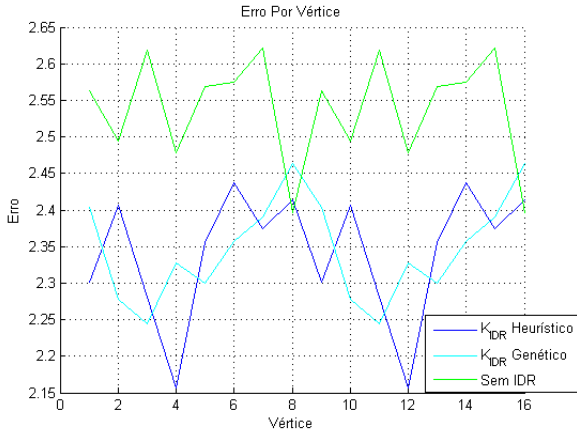
Em uma análise da trajetória reta sem rotação nota-se pela figura 4.5(a) que a resposta para os quatro controladores o comportamento é semelhante, não sendo possível determinar qual apresenta melhor resposta. Contudo a figura 4.5(c) e a tabela 4.3 mostram que o controlador Place possui um erro maior, e dentre os que utilizam a técnica IDR o controlador Heurístico



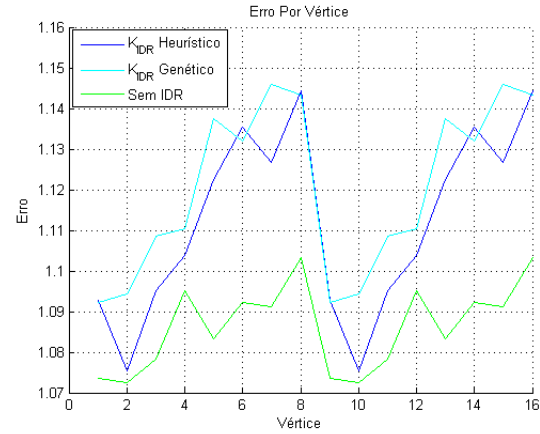
(a) Resposta Trajetória Reta Sem Rotação



(b) Resposta Trajetória Reta Com Rotação



(c) Erro global (Traj. Sem Rotação)



(d) Erro global (Traj. Com Rotação)

Figura 4.5: Resposta trajetória reta aplicada em  $r$  para todos os vértices de  $\mathcal{A}$  para o controlador usando os três ganhos (4.31), (4.32) e (4.33) respectivamente.

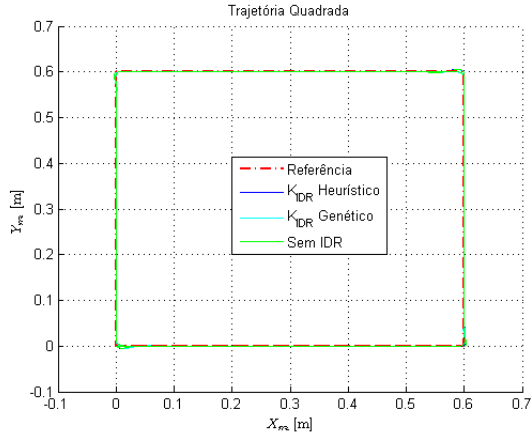
possui o menor erro em todos os vértices do polítopo  $\mathcal{A}$ . Para a trajetória reta com rotação a mesma análise não se mostra válida, o controlador Place apresenta uma resposta melhor em relação aos controladores Heurístico e Genético.

## Caso 2 - Trajetória Quadrada

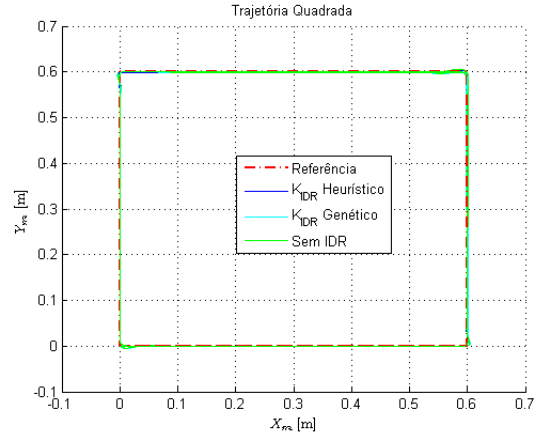
Nesse estudo de caso, uma trajetória quadrada com 0.6 metros de aresta foi gerada. As velocidades nas direções  $x$  e  $y$  são definidas como sendo  $0.3 \text{ m/s}$ . Para alcançar a trajetória quadrada desejada assume-se que o robô deve descolar-se no intervalo 0 a 2 segundos na direção  $\psi = 0^\circ$ , de 2 a 4 segundos na direção  $\psi = 90^\circ$ , de 4 a 6 segundos na direção  $\psi = 180^\circ$ , de 6 a 8 segundos na direção  $\psi = 270^\circ$  e de 8 a 10 segundos na direção  $\psi = 0^\circ$ . Para o ângulo de rotação são assumidas duas situações, a primeira consiste em  $\theta = 0^\circ$  durante os 10 segundos. Na segunda situação,  $\theta = 0^\circ$  no intervalo de 0 a 2 segundos,  $\theta = 30^\circ$  de 2 a 6 segundos e novamente  $\theta = 0^\circ$  no intervalo de 6 a 10 segundos.

As figuras (4.6(a)) e (4.6(b)) representam o gráfico da trajetória quadrada sem rotação e com rotação respectivamente e as figuras (4.6(c)) e (4.6(d)) são os gráficos dos erros de cada

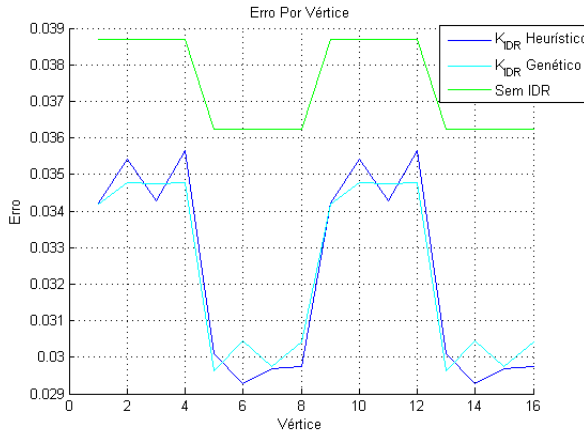
abordagem no respectivos vértice, sem rotação e com rotação. Os demais gráficos referentes a este estudo de caso estão presentes no apêndice D.1.



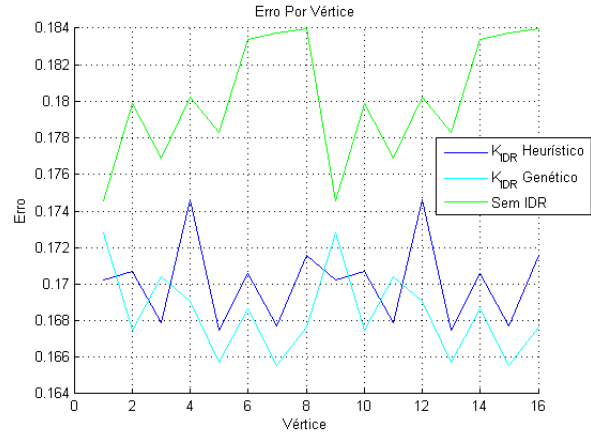
(a) Resposta Trajetória Quadrada Sem Rotação



(b) Resposta Trajetória Quadrada Com Rotação



(c) Erro global (Traj. Sem Rotação)



(d) Erro global (Traj. Com Rotação)

Figura 4.6: Resposta trajetória quadrada aplicada em  $r$  para todos os vértices de  $\mathcal{A}$  para o controlador usando os três ganhos (4.31), (4.32) e 4.33 respectivamente.

As tabelas 4.5 e 4.6 representam os erros de cada simulação, trajetória quadrada com rotação e trajetória quadrada sem rotação respectivamente.

Tabela 4.5: Erro para trajetória quadrada sem rotação.

—	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	—	<b>86,208749</b>	86,297879
Erro bruto	$ e_v $	0,037456	0,032290	0,032324
	$e_{vx}$	0,027463	0,024576	0,024345
	$e_{vy}$	0,025471	0,020944	0,021263
	$e_{\dot{\theta}}$	0,000000	0,000204	0,000251

Tabela 4.6: Erro para trajetória quadrada com rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	94,444605	<b>93,513756</b>
Erro bruto	$ e_v $	0,180090	0,170085	0,168409
	$e_{vx}$	0,029341	0,027464	0,027067
	$e_{vy}$	0,028008	0,024689	0,024756
	$e_{\dot{\theta}}$	0,175462	0,166027	0,164365

A figura 4.6(a) apresenta a resposta à trajetória quadrada sem rotação, em que o comportamento dos quatro controladores é semelhante. As diferenças em resposta dentre os controladores podem ser notadas pela figura 4.6(c) e pela tabela 4.5, em que é possível observar que o erro do controlador **Place** é maior que as demais abordagens, concluindo que o compensador IDR melhora o desempenho dos controladores nesse estudo de caso. Na figura 4.6(b) é possível observar que o controlador **Place** apresenta um desvio da trajetória de referência, esse erro é apresentado na figura 4.6(d) em que é possível visualizar que o desempenho do controlador **Place** é pior. Esse erro é quantificado na tabela 4.6 demonstrando que os demais controladores apresentam um desempenho em torno de 5% melhor que o controlador **Place**. Observe que para as duas situações (com rotação e sem rotação) da trajetória quadrada, o controlador genético apresenta melhor desempenho em relação aos demais.

### Caso 3 - Trajetória Circular

É considerado neste estudo de caso uma trajetória circular, em que foi escolhida o formato de elipse para a trajetória com  $a = 5$  e  $b = 4$  metros. As velocidades em  $x$  e  $y$  são dadas por

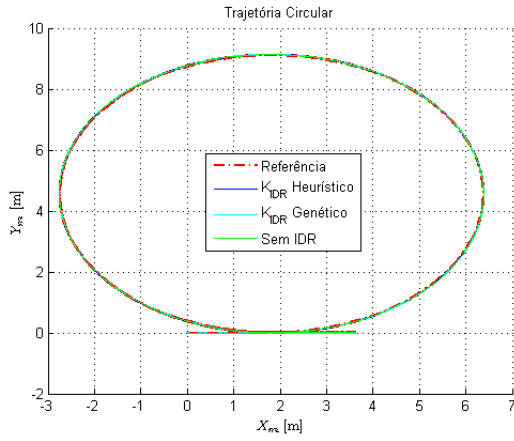
$$v_x = v \cos(\psi) \quad (4.39)$$

$$v_y = v \sin(\psi) \quad (4.40)$$

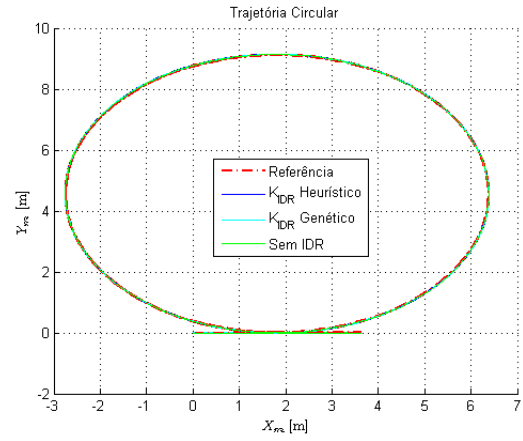
Como realizado nos estudos de caso anteriores,  $\theta = 0^\circ$  durante todo o trajeto para a trajetória circular sem rotação. Para a trajetória circular com rotação,  $\theta = 0^\circ$  no intervalo de 0 a 10 segundos e uma rampa crescente de 10 a 20 segundos. Os gráficos de cada estado estão presentes no apêndice D.1.

Os resultados das simulações são dados pelas figuras 4.7(a) e 4.7(b) em que as trajetórias sem rotação e com rotação são apresentadas respectivamente, e pelas figuras 4.7(c) e 4.7(d) que representam o erro para cada vértice das simulações sem rotação e com rotação. As tabelas 4.7 e 4.8 quantificam o erro para cada abordagem.

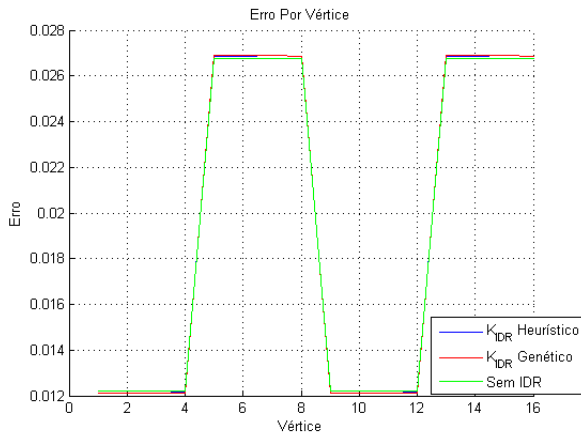




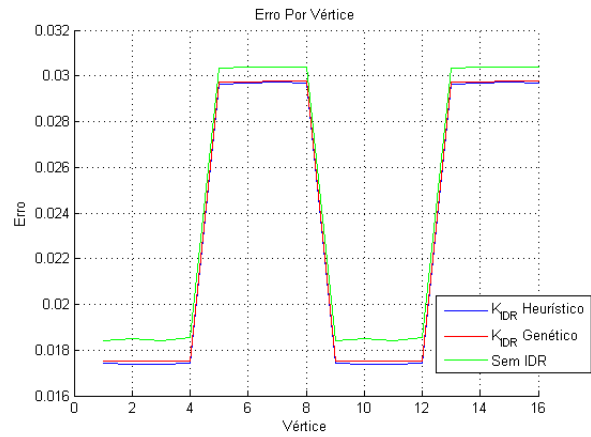
(a) Resposta Trajetória Circular Sem Rotação



(b) Resposta Trajetória Circular Com Rotação



(c) Erro global (Traj. Sem Rotação)



(d) Erro global (Traj. Com Rotação)

Figura 4.7: Resposta trajetória circular aplicada em  $r$  para todos os vértices de  $\mathcal{A}$  para o controlador usando os três ganhos (4.31), (4.32) e 4.33 respectivamente.

Tabela 4.7: Erro para trajetória circular sem rotação.

—	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	—	<b>100,15630</b>	100,198533
Erro bruto	$ e_v $	0,019461	0,019491	0,019499
	$e_{vx}$	0,015408	0,015077	0,015073
	$e_{vy}$	0,011887	0,012351	0,012369
	$e_{\dot{\theta}}$	0,000000	0,000183	0,000216

Este estudo de caso mostra conforme a tabela 4.7 que o comportamento dos três controladores é bem semelhante, sendo o **Place** com melhor desempenho, para a trajetória circular sem rotação. Contudo para a trajetória circular com rotação o mesmo não ocorre, conforme a figura 4.7(d) e a tabela 4.8 o controlador **Place** apresenta o pior desempenho e o controlador Heurístico demonstra melhor desempenho. Importante evidenciar que apesar do melhor desempenho, o controlador Heurístico não apresenta robustez para todo politopo  $\mathcal{A}$ .

Tabela 4.8: Erro para trajetória circular com rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	<b>96,539240</b>	96,907046
Erro bruto	$ e_v $	0,024025	0,023193	0,023282
	$e_{vx}$	0,014811	0,015071	0,015045
	$e_{vy}$	0,012515	0,012354	0,012385
	$e_{\dot{\theta}}$	0,014184	0,012577	0,012739

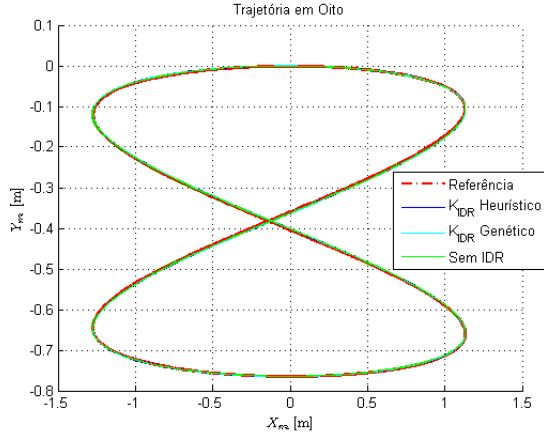
#### Caso 4 - Trajetória em Forma de Oito

A presente seção trata de uma trajetória em forma de oito sem rotação e com rotação. A trajetória em oito sem rotação será apresentada como os estudos de caso apresentados anteriormente. A trajetória em oito com rotação é o estudo de caso a ser detalhado com todos os gráficos e uma discussão dos resultados mais profunda. Essa trajetória foi escolhida por ser uma variação conjunta das demais trajetória, sendo que de forma completa ou parcial é a trajetória mais presente em uma situação real de jogo. A trajetória em forma de oito é utilizada para verificar o desempenho dos controladores no seguimento de referência quando as referências  $\dot{X}_m$  e  $\dot{Y}_m$  mudam frequentemente. Na trajetória sem rotação o robô movimenta-se com um ângulo de rotação  $\theta = 0^\circ$ , mas com uma direção de deslocamento  $\psi$  variando linearmente de  $0^\circ$  a  $360^\circ$ . As componentes das velocidades nas direções  $x$  e  $y$ , para ambas as situações (com rotação e sem rotação) são dadas por

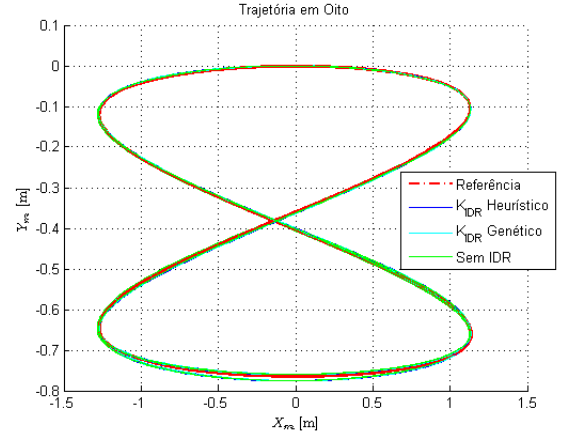
$$v_x = -1.2(\psi \sin(2\psi) + \cos(2\psi)) \quad (4.41)$$

$$v_y = -0.3 \cos(\psi) \quad (4.42)$$

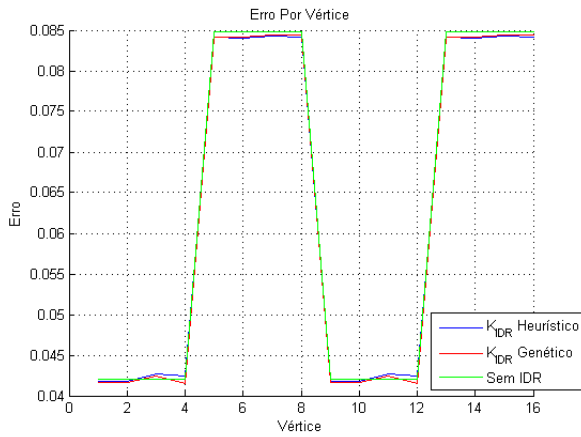
Analisando a trajetória em oito sem rotação, nota-se pela figura 4.8(a) que os três controladores apresentam desempenho bem aproximado e satisfatórios para o seguimento de referência proposto. Este fato é evidenciado pela figura 4.8(c) e pela tabela 4.9 em que é possível visualizar que o erro é também aproximado para todos os controladores. Sendo o controlador **Place** o de pior desempenho, seguido pelos controladores Heurístico e Genético.



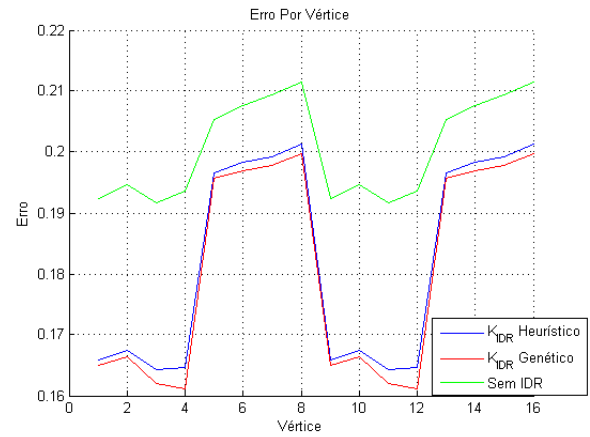
(a) Resposta Trajetória em Oito Sem Rotação



(b) Resposta Trajetória em Oito Com Rotação



(c) Erro global (Traj. Sem Rotação)



(d) Erro global (Traj. Com Rotação)

Figura 4.8: Resposta trajetória em Oito aplicada em  $r$  para todos os vértices de  $\mathcal{A}$  para o controlador usando os três ganhos (4.31), (4.32) e 4.33 respectivamente.

Em uma análise dos resultados da trajetória em oito com rotação, nota-se pela figura 4.8(b) que o controlador **Place**, que não utiliza a técnica de compensação IDR, diverge da trajetória de referência próximo a posição 0 de  $X_m$ . Isso é evidenciado pela figura 4.8(d) que apresenta os erros de cada controlador em cada vértice de  $\mathcal{A}$ . Observe que o controlador **Place** possui um erro maior que os demais. A tabela 4.10 quantifica esse erro em que é visto que os controladores Heurístico e Genético possuem desempenho de aproximadamente 10% melhor que o controlador com **Place**. Essa melhora de desempenho é um ganho significativo e demonstra a vantagem na utilização da técnica IDR para compensação de incertezas de  $\mathcal{A}$ . Na figura 4.9 contém a resposta de  $\dot{X}_m$ , em que as respostas dos controladores são bem semelhantes, não sendo perceptível uma diferença com o uso do IDR. Contudo nas figuras 4.10 e 4.11 em que as respostas de  $\dot{Y}_m$  e  $\dot{\theta}$  são apresentadas, o desempenho dos controladores com o IDR é melhor. Esse melhor desempenho é quantificado na tabela 4.10, em que para os três controladores, os dois que utilizam a compensação IDR apresentam melhor desempenho nas respostas  $\dot{Y}_m$  e  $\dot{\theta}$ . Dentre os controladores com IDR, o controlador de melhor desempenho é o Genético, entretanto os dois apresentam respostas muito próximas.

Tabela 4.9: Erro para trajetória em oito sem rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	99,553390	<b>99,328347</b>
Erro bruto	$ e_v $	0,063461	0,063177	0,063034
	$e_{vx}$	0,063235	0,062954	0,062811
	$e_{vy}$	0,005343	0,005306	0,005298
	$e_{\dot{\theta}}$	0,000000	0,000230	0,000279

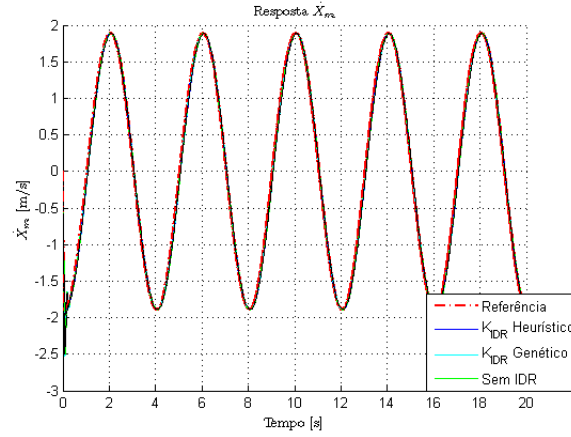
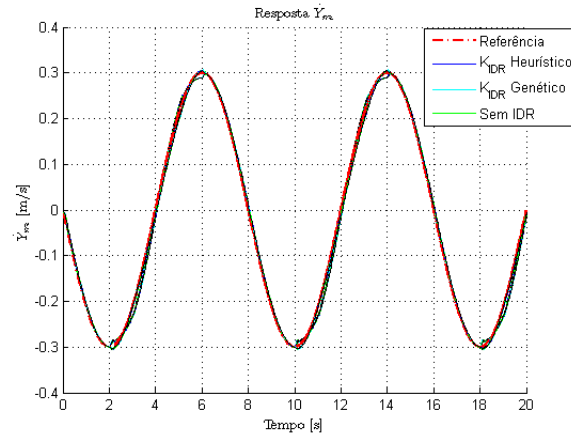
Figura 4.9: Resposta  $\dot{X}_m$  para trajetória em oito com rotaçãoFigura 4.10: Resposta  $\dot{Y}_m$  para trajetória em oito com rotação

Tabela 4.10: Erro para trajetória em oito com rotação.

–	Erro	Sem IDR	Heurístico	Genético
Erro Relativo Percentual	$ e_p $	–	90,850859	<b>90,036468</b>
Erro bruto	$ e_v $	0,199775	0,181497	0,179870
	$e_{vx}$	0,059766	0,062016	0,061769
	$e_{vy}$	0,007112	0,006953	0,006881
	$e_{\dot{\theta}}$	0,190492	0,170431	0,168791

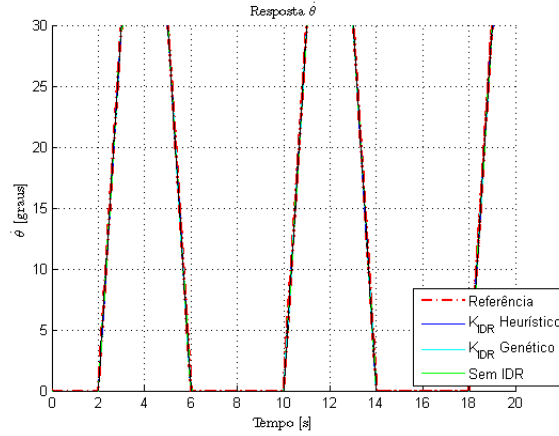


Figura 4.11: Resposta  $\dot{\theta}$  para trajetória em oito com rotação

#### 4.4.5 Simulação de Jogadas

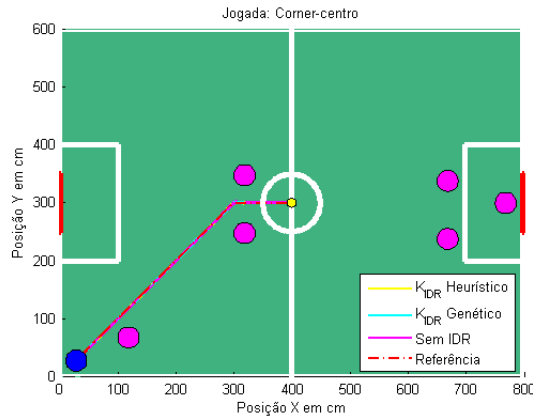
Nessa seção são apresentadas quatro possíveis jogadas em uma partida da liga F-180 da Robocup. Essas jogadas representam uma combinação das trajetórias apresentadas na seção 4.4.4, ou seja, uma trajetória mista para uma situação real de jogo. As trajetórias mistas apresentadas são: Corner-Centro, Lateral-Área, Centro-Ponta e Condução-Bola. As trajetórias (jogadas) são aplicadas o politopo  $\mathcal{A}$  para efeito comparativo de desempenho das abordagens para determinação de ganho IDR. O critério de desempenho adotado nesta seção é o erro acumulado em centímetros ao longo da trajetória e o erro da posição final da trajetória, conforme apresentado na seção 4.4.3. Dessa forma é possível demonstrar se a bola foi efetivamente acertada ao fim da jogada. Para cada trajetória são apresentados os gráficos de trajetória e os gráficos de erros em centímetros de desvio de trajetória e posição. Nos gráficos de erro da posição final os valores positivos ou negativos dos erros, indicam o sentido do erro em relação ao raio da bola, ou seja, distância posterior ou anterior ao centro.

A Small Size League é formada por robôs de 18 cm de diâmetro e não podem ser maiores que 15 cm, cada equipe é formada por 6 jogadores (robôs), a bola possui 46 gramas e 4,3 cm de diâmetro. Para o torneio de 2015 as dimensões serão de 890cm  $\times$  605cm [WB13]. As simulações de jogadas são baseadas em informações de posição de referência que em uma competição oficial são passadas ao F-180 pelo computador do sistema de visão compartilhada, ou seja, essas simulações não possuem algoritmo de colisão ou previsão. Nos gráficos de trajetória o tamanho da bola foi aumentado em 4 vezes para facilitar a visualização, contudo o tamanho real da mesma é utilizado nos critérios e demais cálculos. O critério utilizado de alcance da bola na trajetória é dado pela equação (4.38), considerando o raio da bola de 2,15 cm. Dessa forma  $e_{xy}$  deve ser menor que 1,3975 cm. O outro critério de desempenho de controladores adotado é dado pela equação (4.37).

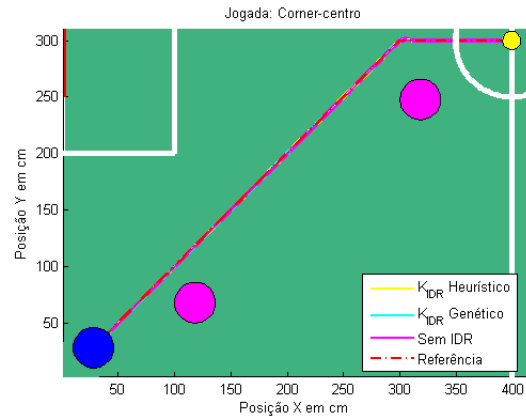
##### Jogada 1 - Corner-Centro

Essa jogada consiste em partir com o jogador da posição próxima ao corner até o centro do campo onde se encontra a bola. A ideia é que a trajetória seja o mais retilínea possível. O

resultado da trajetória dessa jogada é apresentado na figura 4.12(a) e aplicado uma aproximação na 4.12(b) em que se pode ver mais detalhadamente. O erro acumulado em centímetros ao longo da trajetória é mostrado na figura 4.13(a), o erro da posição final do robô é mostrado na figura 4.13(b).

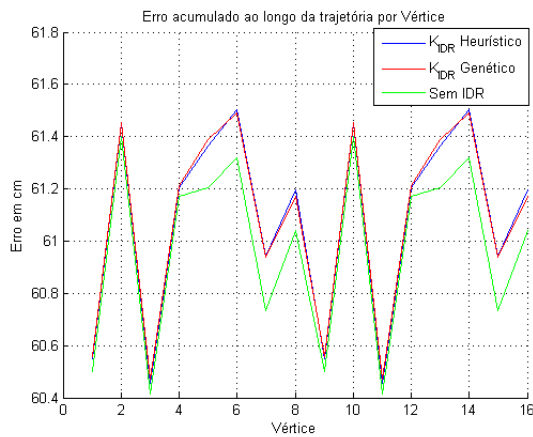


(a) Resposta da jogada Centro-Corner

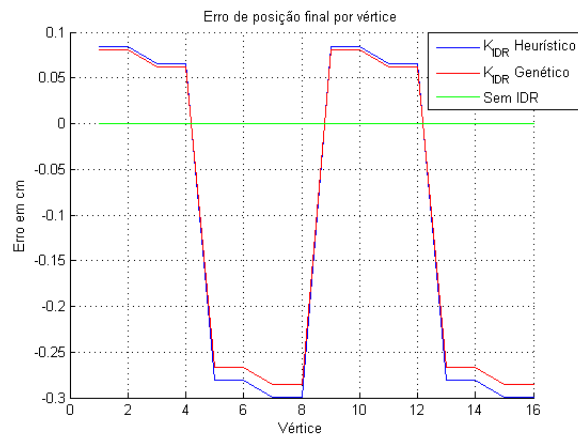


(b) Resposta da jogada Centro-Corner (Zoom)

Figura 4.12: Resposta da jogada Centro-Corner



(a) Erro acumulado da jogada Corner-centro



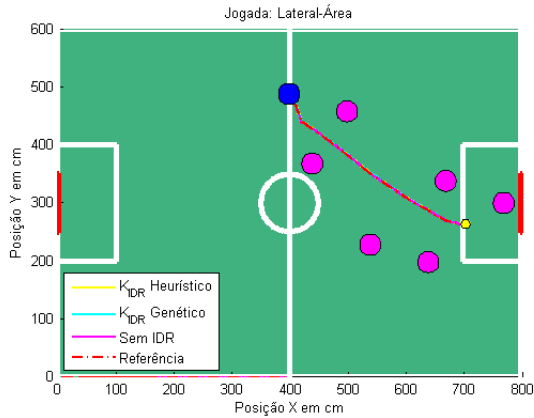
(b) Erro da posição final da jogada Corner-centro

Figura 4.13: Erros da jogada Centro-Corner

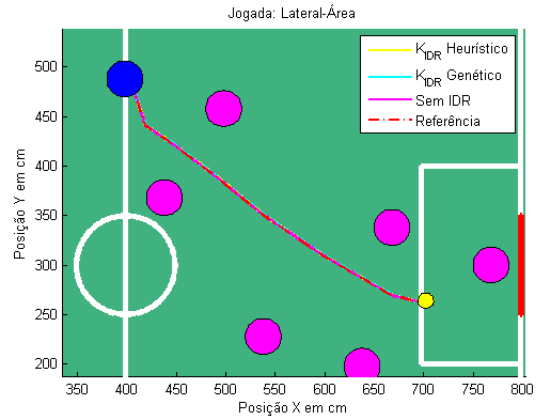
Com base na figura 4.12(a) nota-se que os três controladores seguem a referência sem grandes desvios, conseguindo atingir a bola ao meio campo. Com base no gráfico de erro acumulado da figura 4.13(a) e no gráfico de erro de posição final da figura 4.13(b) o controlador sem o IDR apresenta um melhor desempenho, mesmo que com uma diferença pequena em relação aos demais. O erro na posição final para o controlador sem IDR é próximo de zero, para os controladores com IDR o maior erro é menor que 0,3 cm. O melhor desempenho do controlador sem IDR é devido a característica mais retilínea da jogada, semelhante ao resultado da trajetória retilínea com rotação.

## Jogada 2 - Lateral-Área

Nessa jogada o F-180 no campo adversário próximo a linha lateral tem por objetivo passar entre os robôs adversários até a grande área onde se localiza a bola. O resultado da trajetória dessa jogada é apresentado na figura 4.14(a) e aplicado uma aproximação na 4.14(b) em que se pode ver mais detalhadamente. O erro acumulado em centímetros ao longo da trajetória é mostrado na figura 4.15(a), o erro da posição final do robô é mostrado na figura 4.15(b).

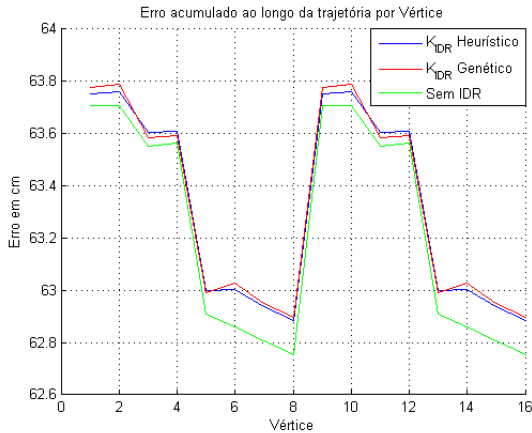


(a) Resposta da jogada Lateral-Área

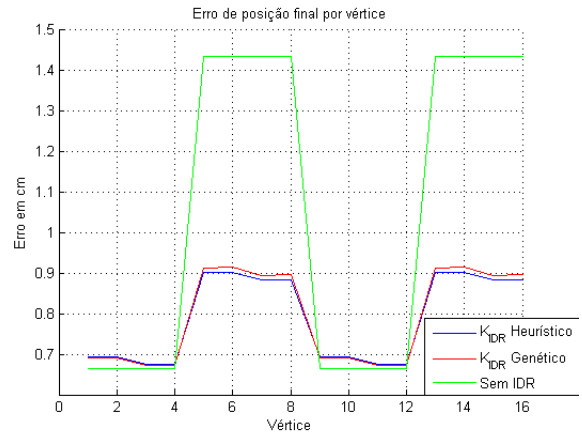


(b) Resposta da jogada Lateral-Área (Zoom)

Figura 4.14: Resposta da jogada Lateral-Área



(a) Erro acumulado da jogada Lateral-Área



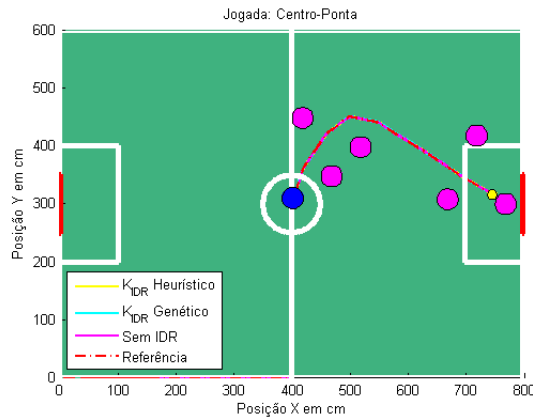
(b) Erro da posição final da jogada Lateral-Área

Figura 4.15: Erros da jogada Lateral-Área

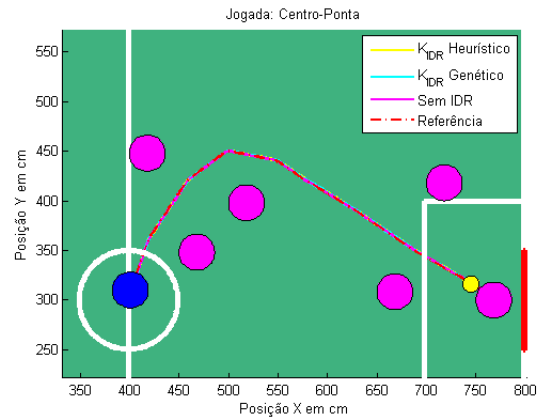
A jogada Lateral-Área possui características de trajetória retilínea e circular com e sem rotação. De acordo com as figuras 4.14(a) e 4.14(b) todos os controladores aparentemente acompanham a trajetória de referência e possuem um erro acumulado bem semelhante conforme mostra a figura 4.15(a). Entretanto a figura 4.15(b) mostra que o controlador erra a bola em até 1,43 cm, indicando que a bola não é alcançada de acordo com o critério adotado.

### Jogada 3 - Centro-Ponta

A jogada centro-ponta é baseada no deslocamento do F-180 do meio campo desviando de robôs adversários em uma combinação de trajetória retilínea, circular e em oito para alcançar a bola dentro da grande área em frente ao goleiro. As figuras 4.16(a) e 4.16(b) representam a trajetória do robô no campo de jogo e as figuras 4.17(a) e 4.17(b) os erros acumulados e de posição final em centímetros respectivamente.

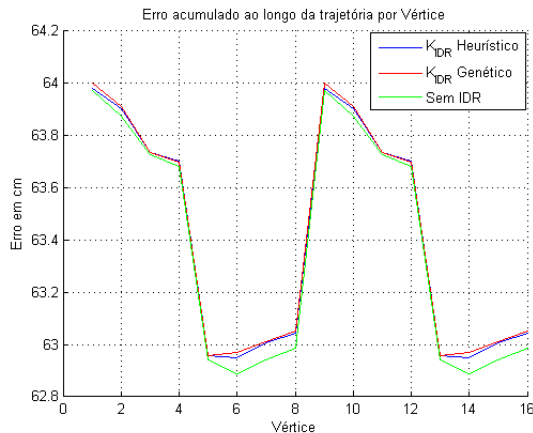


(a) Resposta da jogada Centro-Ponta



(b) Resposta da jogada Centro-Ponta (Zoom)

Figura 4.16: Respostas da jogada Centro-Ponta



(a) Erro acumulado da jogada Centro-Ponta



(b) Erro da posição final da jogada Centro-Ponta

Figura 4.17: Erros da jogada Centro-Ponta

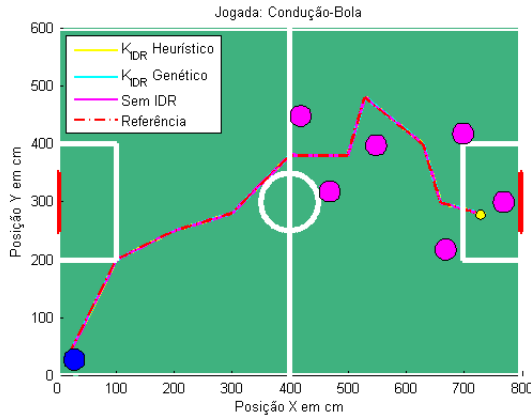
A figura 4.16(a) mostra que os controladores seguem a referência para essa jogada que combina trajetórias circulares e retilíneas sem grandes desvios da referência. Embora a figura 4.17(a) mostra que o erro acumulado em centímetros ao longo da trajetória é muito semelhante para os três controladores, nota-se pela figura 4.17(b) que o controlador sem IDR possui erro de até 2,47 centímetros para alguns vértices. Esse erro da posição final representa que a bola



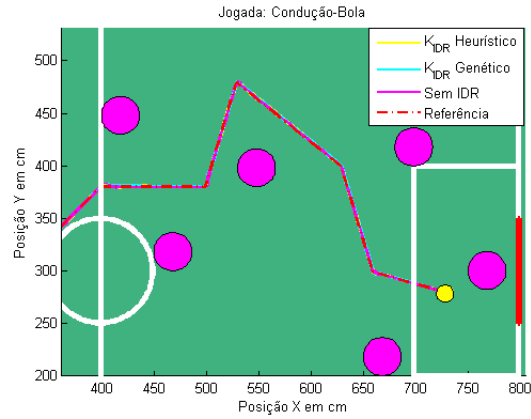
não foi atingida ao final da trajetória de acordo com o critério proposto, ou seja, o controlador sem IDR não é capaz de realizar a jogada com sucesso.

### Jogada 4 - Condução-Bola

A jogada Condução-Bola difere das demais jogadas apresentadas em que o F-180 se desloca para alcançar a bola, nessa jogada a ideia é que o F-180 inicia a trajetória com a bola dominada e a conduza até o ponto final da trajetória. Obviamente em uma situação real de jogo, essa trajetória seria modificada diversas vezes devido a interação dos demais robôs. Com intuito de simular essas variações essa jogada alterna movimentos e dribles em diversos pontos do campo. O resultado da trajetória da jogada Condução-Bola é apresentado nas figuras 4.18(a) e 4.18(b) e com o respectivo erro demonstrado nas figuras 4.19(a) e 4.19(b).

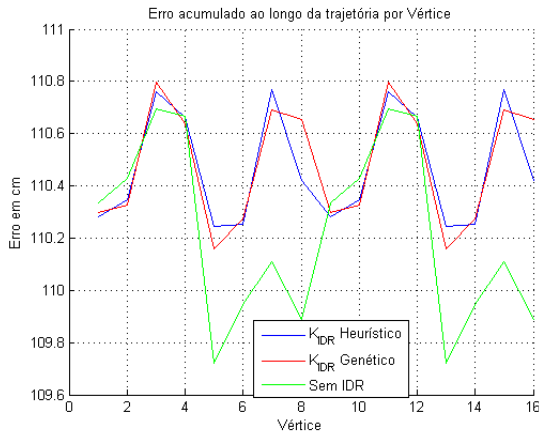


(a) Resposta da jogada Condução-Bola

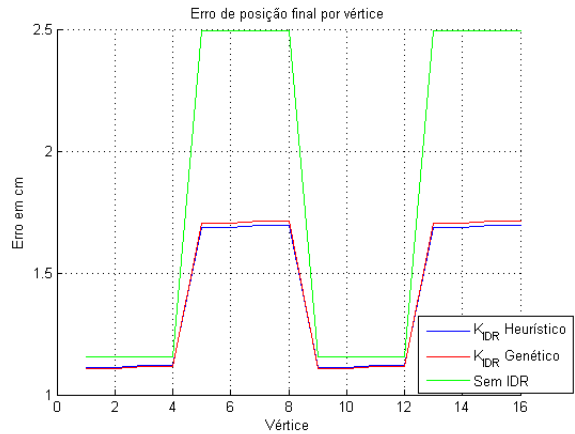


(b) Resposta da jogada Condução-Bola (Zoom)

Figura 4.18: Respostas da jogada Condução-Bola



(a) Erro acumulado da jogada Condução-Bola



(b) Erro da posição final da jogada Condução-Bola

Figura 4.19: Erros da jogada Condução-Bola

Não é possível notar desvios significantes na trajetória da jogada com base nas figuras 4.18(a) e 4.18(b), em que os três controladores aparentemente possuem desempenho semelhante. Entretanto com base no gráfico apresentado na figura 4.19(b), nota-se que para os vértices 4,5,6,7,8,9,12,13,14 e 15 o controlador sem IDR apresentou um erro de 2,5 centímetros. Isso indica que a bola não foi conduzida ao ponto final, ou seja, a jogada não foi sucedida com o uso do controlador sem IDR. Os demais controladores também apresentam um erro acima do critério máximo adotado para os mesmos vértices, contudo ainda assim são capazes de atingir a bola, já que o erro é menor que o raio da mesma.

#### 4.4.6 Análise dos Resultados

O objetivo deste capítulo era apresentar as vantagens de utilização da técnica de compensação de incertezas IDR frente a um modelo de um robô omnidirecional com as incertezas dadas por um Politopo  $\mathcal{A}$  em relação a um controlador **Place** sem a utilização do IDR. Para tal, foram propostos dois controladores com o IDR, cuja diferença entre eles é a forma de sintonia do ganho  $K_{IDR}$ . O controlador Heurístico utiliza o algoritmo heurístico proposto no capítulo 3 para determinar uma matriz de ganho  $K_{IDR}$ , de uma forma fácil e rápida que atenda os critérios de desempenho mínimo em cada vértice. O controlador Genético utiliza um algoritmo genético de minimização para determinação de  $K_{IDR}$ , em que o critério de desempenho também é baseado no erro mínimo encontrado para um dado ganho como o Heurístico. Contudo utiliza técnicas mais complexas de determinação da matriz candidata ao ganho  $K_{IDR}$ . Esses dois controladores possuem um problema em comum que apesar de determinarem um ganho  $K_{IDR}$ , esse ganho não garante estabilidade robusta para todo o politopo  $\mathcal{A}$ , mas somente para os vértices e alguns pontos próximos.

Essas três abordagens de controladores são testadas nos estudos de caso propostos na seção 4.4.4 com intuito de apresentar o comparativo de desempenho de cada controlador para as trajetórias reta, quadrada, circular e em oito, todas com situações de rotação e sem rotação. Para a trajetória reta sem rotação o IDR mostrou ter um desempenho melhor e controlador Heurístico apresentou menor erro. Contudo o mesmo não ocorreu para a trajetória reta com rotação, em que o IDR se mostrou pior que a ausência deste, aumentando o erro na trajetória do robô. Para a trajetória quadrada o IDR mostrou desempenho superior tanto com o rotação ou sem rotação, reduzindo a influência das incertezas nas respostas do sistema. Na trajetória sem rotação o controlador Heurístico obteve melhor desempenho, mas ao realizar a trajetória com rotação o controlador Genético apresentou menor erro e conseqüentemente melhor desempenho. No caso de estudo da trajetória circular, em situação sem rotação o IDR não apresentou vantagem e teve desempenho muito próximo do controlador **Place**, com a rotação o IDR demonstra um ganho de desempenho e o controlador Heurístico se sobressai frente aos demais com um melhor desempenho. No último estudo de caso, trajetória em oito, na situação sem rotação o desempenho dos três controladores é bem semelhante e o uso do IDR, mesmo tendo melhor desempenho, não traz vantagens significativas. Contudo com rotação o IDR apresenta um ganho extremamente significativo de aproximadamente 10%. O controlador Genético apresenta melhor desempenho nesse estudo de caso, com e sem rotação.

No caso das simulações de jogadas da seção 4.4.5, em que foram testados os três contro-

ladores com objetivo de apresentar um comparativo de desempenho entre os controladores em situações mais próximas de uma partida da Robocup, os controladores com IDR confirmaram o melhor desempenho frente ao controlador sem IDR. Para a jogada Corner-Centro devido às características da trajetória o controlador sem IDR obteve um resultado numericamente superior, mas não significativo em termos de sucesso da jogada em relação aos controladores com IDR. Na jogada Lateral-Área os controladores com IDR apresentaram melhor desempenho e o controlador sem IDR falhou em alcançar a bola ao final da jogada. O mesmo ocorreu em relação à jogada Centro-Ponta em que os controladores com IDR mostraram melhor desempenho, enquanto que o controlador sem IDR não foi capaz de atingir a bola. No caso da jogada Condução-Bola o erro do controlador sem IDR chegou a ser o dobro dos controladores com IDR para alguns vértices. Contudo de acordo com o critério de alcance a bola, em alguns vértices todos os três controladores falharam.

Concluídos os estudos de casos e as simulações de jogadas, os resultados oriundos desses mostram que a técnica de compensação de incertezas IDR se mostrou vantajosa na maioria dos testes em relação ao não uso da mesma frente a um modelo com a presença de incertezas. Um ponto importante na comparação entre as abordagens com IDR e sem IDR é que as diferenças podem ser maiores para um modelo não tão bem comportado como é o caso do modelo do robô omnidirecional utilizado, pois trata-se praticamente de uma modelagem sobre um motor CC, em um modelo cujas não linearidades são mais acentuadas, como o modelo de uma aeronave apresentado em [Lav05], o IDR apresenta resultados melhores de compensação. De toda forma os ganhos no uso do mesmo nos estudos de caso são extremamente satisfatórios ao critério de desempenho proposto, proporcionando que a trajetória do robô seja bem mais próxima do seguimento de referência em cada caso, diminuindo a influência das incertezas no modelo em relação as respostas  $\dot{X}_m$ ,  $\dot{Y}_m$  e  $\dot{\theta}$ . Em relação aos controladores que utilizam o IDR, nota-se que o melhor desempenho foi alternado dentre eles entre os estudos de caso, não assegurando a nenhum uma posição distinta de melhor desempenho em todos os estudos de caso. Entretanto, nenhum apresentou o desempenho muito pior em detrimento de outro, mas sim sempre desempenhos muito próximos. Dessa forma o controlador Heurístico se mostra a melhor opção já que apresenta um custo computacional menor que o Genético.



## Conclusões e Perspectivas

A tarefa de projetar um controlador não linear para um sistema com incertezas paramétricas deve ser realizada com muito cuidado, para que se garanta os critérios de desempenho. Como discutido neste trabalho, diversos fatores podem influenciar nesse projeto, divergências entre modelo e sistema real, região de operação, critérios de desempenho etc. Este trabalho tratou questões relacionadas à abordagem não linear de linearização por realimentação clássica, de compensação de incertezas ao modelo matemático por meio de um compensador de Inversão Dinâmica Robusta, sintonia de controlador utilizando heurística e Meta-Heurística para determinação de ganho e ao controle de trajetórias de robôs móveis omnidirecionais. Aspectos relativos a linearização por realimentação do modelo foram discutidos de forma a se obter o modelo no formato desejado. A técnica de compensação IDR foi apresentada e discutida, trata-se de uma abordagem relativamente recente para compensação de incertezas, auxiliando consideravelmente na compensação da influência das incertezas na resposta do sistema em relação a referência.

Quanto às estratégias heurística e meta-heurística apresentadas para determinação do ganho  $K_{IDR}$ , mostrou-se a viabilidade de duas técnicas de busca de ganho com base em critérios de minimização do erro normalizado. Fator fundamental para a construção do compensador IDR aplicado ao modelo, já que em [Lav05] não é apresentada nenhuma técnica para determinação de  $K_{IDR}$ . Em relação à aplicação de controle, mostrou-se a viabilidade de um controlador não linear que combina a linearização por realimentação de estados clássica a um controlador linear em conjunto com um compensador de incertezas. A associação desses três métodos gera um sistema de controle capaz de compensar as incertezas de parâmetros em regiões lineares e não lineares. Foi utilizado para a síntese dos controladores o método de alocação exata de polos, denominado controlador clássico. Essa escolha foi proposital para que o sistema seja mais susceptível a variação dos parâmetros de forma a evidenciar a ação do compensador IDR. A síntese do compensador IDR foi realizada utilizando os métodos heurístico e meta-heurístico apresentados, obtendo-se dois ganhos  $K_{IDR}$ . O desempenho dos compensadores foi comparado com a topologia sem IDR, ou seja, para o mesmo ganho de realimentação estática de estados. Mostrou-se que na presença de incertezas os compensadores são capazes de garantir o menor erro em relação a trajetória desejada comparado ao sistema de controle sem IDR. Os resultados das simulações para seguimento de referência realizadas para o robô F180 com quatro rodas evidenciam as vantagens do sistema de controle com o compensador IDR em relação à topologia

com ausência deste.

## 5.1 Contribuições da Dissertação

Nesta dissertação foi apresentada uma topologia de utilização da abordagem de compensação de Inversão Dinâmica Robusta associada à uma técnica de controle não linear, linearização por realimentação. Foram propostos duas estratégias para determinação do ganho do compensador IDR,  $K_{IDR}$ , baseadas em heurística e meta-heurística, aplicados sobre o modelo linearizado do robô F180 considerado incerto, mostraram-se bem sucedidas. Os compensadores projetados com as técnicas foram capazes de manter o sistema estável e atender os critérios de desempenho definidos para os casos de estudo apresentados para a trajetória do robô.

Dentre as contribuições deste trabalho de dissertação estão dois artigos publicados no Congresso Brasileiro de Automática 2014 e na 19ª Conferência Internacional sobre Métodos e Modelos em Automação e Robótica (MMAR-IEEE):

- F. P. Silva., V. J. S. Leite. e E. G. Nepomuceno. Heurística para Sintonia de Ganho de Compensação de Incertezas de Modelagem Via Multi-Inversão Robusta. *Congresso Brasileiro de Automática*, Belo Horizonte, 2014.
- F. P. Silva., V. J. S. Leite. e E. G. Nepomuceno. Heuristic to tune the compensation gain of modeling uncertainties through the robust multi inversion. *19th International Conference on Methods and Models in Automation and Robotics - MMAR/IEEE*, Miedzyzdroje - Polônia, 2014.

## 5.2 Trabalhos Futuros

Para trabalhos futuros são propostas as seguintes direções de pesquisa:

- Reproduzir os casos de estudo propostos a um robô real, aplicando os controladores desenvolvidos.
- Pesquisar a síntese de um controlador robusto utilizando síntese convexa, LMI. De forma a garantir robustez para todo o politopo de incertezas.
- Utilizar um outro modelo que não o do robô F180 para aplicação da topologia proposta, Linearização por realimentação clássica e compensador IDR. De forma que em um modelo mais não-linear os efeitos dos benefícios do compensador IDR sejam mais notórios, conforme os resultados apresentados em [Lav05].

## Modelo do Robô Omnidirecional - F-180

Este apêndice apresenta o sistema de locomoção de um robô móvel seguido da demonstração para se obter o modelo não-linear do robô F-180 que foi desenvolvido em [Soa11], modelo este que é utilizado ao longo desta dissertação.

### A.1 Descrição do modelo do robô F180 Omnidirecional

O comportamento de um robô, sua capacidade de locomoção e as equações que regem seu movimento são influenciadas pela sua geometria e pela sua dinâmica. Dessa forma, aspectos de caráter mecânico, elétrico e a disposição física dos elementos que o compõem, são extremamente relevantes na escolha e implementação dos controladores. O modelo do robô pode ser dividido em modelo cinemático, que contém aspectos geométricos, e em modelo dinâmico que contém os aspectos dinâmicos.

O modelo cinemático descreve o movimento do robô em função das velocidades das rodas desconsiderando as forças que atuam sobre as mesmas. Todavia, o modelo dinâmico descreve as relações dinâmicas entre as coordenadas de postura e a orientação do robô e os torques desenvolvidos pelos seus atuadores [Soa11]. A formulação do modelo dinâmico pode ser concebida de acordo com o formalismo de Newton-Euler ou segundo o formalismo de Euler-Lagrange. As equações de Newton tratam cada corpo rígido separadamente e modelam explicitamente as restrições de cada corpo através das forças que atuam sobre esses. Lagrange e d'Alembert fornecem um procedimento sistemático para eliminar restrições mecânicas, através de um sistema de equações geral. O modelo a ser desenvolvido neste relatório, vale-se da formulação de Newton-Euler.

#### A.1.1 Modelo Geométrico do Robô

A figura A.1 representa o modelo geométrico do robô F-180 da *RoboCup* de quatro rodas que é abordado neste trabalho. A figura apresenta dois sistemas de coordenadas, sendo que  $S_r = [X_r \ Y_r]$  representa o sistema de referência espacial ou global e  $S_m = [X_m \ Y_m]$  representa o sistema de movimento do robô ou móvel, sendo que o eixo  $Y_m$  indica a frente do robô. Como pode ser visualizado na figura A.1 do modelo geométrico, há as seguintes definições:

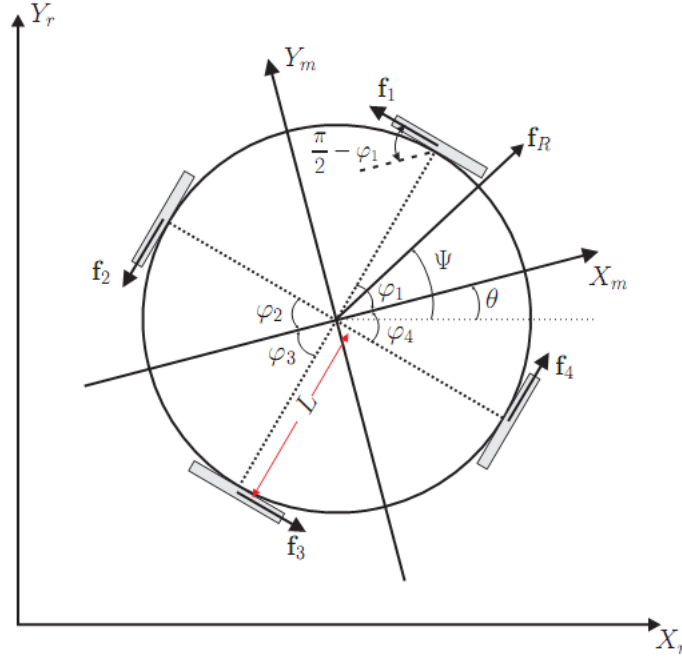


Figura A.1: Modelo Geométrico do Robô. Fonte: [Soa11]

- $S_m = [X_m \ Y_m]$  sistema de coordenadas de movimento do robô.
- $S_r = [X_r \ Y_r]$  sistema de coordenadas de referência espacial ou global.
- $\theta$  Ângulo de rotação do robô em relação a  $S_r$ .
- $\psi = \theta + \gamma$  é o azimute no  $S_r$ .
- $\gamma$  é o ângulo entre  $X_m$  e  $f_r$ , é a azimute no  $S_r$ .
- $\phi_i$  ( $i = 1, 2, 3$  e  $4$ ) é os ângulo das rodas com  $X_m$ .

Como o ângulo de rotação do robô é  $\theta = \phi - \gamma$ , percebe-se que mesmo que o azimute do robô mude arbitrariamente, o robô pode realizar movimentos de translação sem rotacionar. Quando os quatro motores estão ativados, obtém-se quatro forças de tração  $f_1, f_2, f_3$  e  $f_4$  que criam uma força de translação e um binário de rotação. Cada força  $f_i$  é o torque do motor multiplicado pelo raio da roda. A força resultante  $f_r$  depende da posição exata em que as rodas estão dispostas no robô e da direção do movimento do robô.

### A.1.2 Matriz de acoplamento das forças

Analizando o movimento do robô ao longo das direções  $X_m$  e  $Y_m$ , considerando-se a velocidade e aceleração instantâneas do robô no seu próprio sistema de referência,  $S_m = [X_m, Y_m]$ . Assume-se que o centro de massa do robô está localizado no seu centro geométrico, valendo-se da formulação de Newton-Euler:

$$\Sigma F = m \cdot a \rightarrow (f_1 + f_2 + f_3 + f_4) = m \cdot a \longrightarrow a = \frac{(f_1 + f_2 + f_3 + f_4)}{m}$$



$$\Sigma M_\theta = J_w \ddot{\theta} \longrightarrow M_\theta = L \cdot f_A.$$

Fazendo,

$$\Sigma M_\theta = L_1 f_1 + L_2 f_2 + L_3 f_3 + L_4 f_4,$$

sendo  $L_1 = L_2 = L_3 = L_4 = L$ , obtêm-se

$$\Sigma M_\theta = L(f_1 + f_2 + f_3 + f_4) = J_w \ddot{\theta}$$

. Logo,

$$\ddot{\theta} = \frac{L}{J_w}(f_1 + f_2 + f_3 + f_4),$$

para

$$\Sigma F_x = m \cdot a_x \quad \Sigma F_y = m \cdot a_y$$

. Expandi-se para,

$$a_x = \frac{(f_1 + f_2 + f_3 + f_4)_x}{m}$$

$$a_y = \frac{(f_1 + f_2 + f_3 + f_4)_y}{m}$$

. Baseando-se na decomposição vetorial possível da figura A.1, as forças em  $x$  e  $y$  são dadas por

$$\sin(\phi_1) = \frac{f_{1x}}{f_1} \longrightarrow f_1^p = f_1 \longrightarrow f_{1x} = f_1 \sin(\phi_1)$$

$$\cos(\phi_i) = \frac{f_{iy}}{f_i} \longrightarrow f_{iy} = f_i \cos(\phi_i)$$

A aceleração linear do robô pode ser decomposta em suas componentes  $x$  e  $y$  analisando a geometria do robô. Essas componentes podem ser apresnetadas conforme as equações:

$$\Sigma F_x = m \cdot a_x = -f_1 \sin \phi_1 - f_2 \sin \phi_2 + f_3 \sin \phi_3 + f_4 \sin \phi_4$$

$$\Sigma F_y = m \cdot a_y = f_1 \cos \phi_1 - f_2 \cos \phi_2 - f_3 \cos \phi_3 + f_4 \cos \phi_4$$

Considerando que o robô tenha distribuição de massa homogênea,  $J_w = \frac{1}{2}mL^2$ , para um robô que tenha distribuição de massa periférica,  $J_w = mL^2$ . Para qualquer distribuição de massa entre a concentração de massa no centro geométrico do robô e sua periferia, pode-se escrever  $J_w = \sigma mL^2$ , em que  $0 < \sigma \leq 1$  representa a constante de distribuição de massa.

Reescrevendo as equações para a forma matricial:

$$\begin{bmatrix} a_x \\ a_y \\ \ddot{\theta} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin \phi_1 & -\sin \phi_2 & \sin \phi_3 & \sin \phi_4 \\ \cos \phi_1 & -\cos \phi_2 & -\cos \phi_3 & \cos \phi_4 \\ \frac{mL}{J_w} & \frac{mL}{J_w} & \frac{mL}{J_w} & \frac{mL}{J_w} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (\text{A.1})$$

Substituindo  $J_w = \sigma mL^2$  na equação matricial (A.1),

$$\begin{bmatrix} a_x \\ a_y \\ \ddot{\theta} \end{bmatrix} = \frac{1}{m} \begin{bmatrix} -\sin \phi_1 & -\sin \phi_2 & \sin \phi_3 & \sin \phi_4 \\ \cos \phi_1 & -\cos \phi_2 & -\cos \phi_3 & \cos \phi_4 \\ \frac{mL}{\sigma L} & \frac{mL}{\sigma L} & \frac{mL}{\sigma L} & \frac{mL}{\sigma L} \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix}$$

### A.1.3 Modelagem Cinemática

Para o sistema de coordenadas  $S_r$  têm-se que:

$$\epsilon_r = \begin{bmatrix} X_r \\ Y_r \\ \Theta \end{bmatrix}$$

Para o sistema de coordenadas  $S_m$  têm-se que:

$$\epsilon_m = \begin{bmatrix} X_m \\ Y_m \\ \Theta \end{bmatrix}$$

Em que  $\epsilon$  representa o centro de massa, postura do robô. Dessa forma  $\dot{\epsilon}_r = R(\theta)\dot{\epsilon}_m$ , em que

$$R(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

As velocidades lineares das rodas são dadas por:

$$\begin{bmatrix} V_1 \\ V_2 \\ V_3 \\ V_4 \end{bmatrix} = \begin{bmatrix} -\sin \phi_1 & \cos \phi_1 & L \\ -\sin \phi_2 & \cos \phi_2 & L \\ \sin \phi_3 & -\cos \phi_3 & L \\ \sin \phi_4 & \cos \phi_4 & L \end{bmatrix} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}$$

As velocidades anteriormente apresentadas representam as velocidades lineares dos sistema. Entretanto é necessário representar essas velocidades em função das velocidades angulares de forma que seja possível acrescentar a dinâmica dos atuadores. Sendo  $v_i = \phi_i r$ , em que  $\phi_i$  é a velocidade angular da roda e  $r$  é o raio da mesma.  $\omega_{mi} = \phi_i n$  em que  $n$  é o coeficiente de redução da caixa de redução.

$$\phi_i = \frac{v_i}{r} \quad \omega_{mi} = \frac{v_i}{r} n$$

$$\begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \\ \omega_{m4} \end{bmatrix} = \frac{n}{r} D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix}$$

$$D = \begin{bmatrix} -\sin \phi_1 & \cos \phi_1 & L \\ -\sin \phi_2 & \cos \phi_2 & L \\ \sin \phi_3 & -\cos \phi_3 & L \\ \sin \phi_4 & \cos \phi_4 & L \end{bmatrix}$$

Logo o modelo cinemático direto é dado por,

$$\begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = \frac{r}{n} D^+ \begin{bmatrix} \omega_{m1} \\ \omega_{m2} \\ \omega_{m3} \\ \omega_{m4} \end{bmatrix}$$

Em que  $D^+$  é a pseudo inversa da matriz  $D$ .

### A.1.4 Modelo Dinâmico de base Móvel

Considerando

$$F_r = M_m \ddot{\epsilon}_r \quad (\text{A.2})$$

$$F_r = R(\theta) f_m \quad (\text{A.3})$$

Logo

$$F_r = [F_x \ F_y \ \sum M_o]^T_{Global} \quad (\text{A.4})$$

$$f_m = [f_x \ f_y \ \sum M_o]^T_{Movel} \quad (\text{A.5})$$

$$M_m = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_w \end{bmatrix}$$

$$R(\theta) f_m = M_m (R(\theta) \ddot{\epsilon}_m + \dot{R}(\theta) \dot{\epsilon}_m) \rightarrow$$

$$f_m = M_m R(\theta)^T (R(\theta) \ddot{\epsilon}_m + \ddot{\epsilon}_m) \rightarrow$$

$$f_m = M_m (R(\theta)^T \dot{R}(\theta) \dot{\epsilon}_m + \ddot{\epsilon}_m)$$

$$\begin{bmatrix} f_x \\ f_y \\ \sum M_o \end{bmatrix} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_w \end{bmatrix} \underbrace{\begin{bmatrix} 0 & -\dot{\theta} & 0 \\ \dot{\theta} & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}}_{R(\theta)^T \dot{R}(\theta)} \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix}$$

$$f_x = -f_1 \sin \phi_1 - f_2 \sin \phi_2 + f_3 \sin \phi_3 + f_4 \sin \phi_4 \quad (\text{A.6})$$

$$f_y = f_1 \cos \phi_1 - f_2 \cos \phi_2 - f_3 \cos \phi_3 + f_4 \cos \phi_4 \quad (\text{A.7})$$

$$\sum M_o = (f_1 + f_2 + f_3 + f_4) L \quad (\text{A.8})$$

Logo,

$$\begin{aligned} m(\ddot{X}_m - \dot{\theta} \dot{Y}_m) &= -f_1 \sin \phi_1 - f_2 \sin \phi_2 + f_3 \sin \phi_3 + f_4 \sin \phi_4 \\ m(\dot{\theta} \dot{X}_m + \ddot{Y}_m) &= f_1 \cos \phi_1 - f_2 \cos \phi_2 - f_3 \cos \phi_3 + f_4 \cos \phi_4 \\ J_w \ddot{\theta} &= (f_1 + f_2 + f_3 + f_4) L \end{aligned}$$

$$\ddot{X}_m = \frac{-f_1 \sin \phi_1 - f_2 \sin \phi_2 + f_3 \sin \phi_3 + f_4 \sin \phi_4}{m} + \dot{\theta} Y_m \quad (\text{A.9})$$

$$\ddot{Y}_m = \frac{f_1 \cos \phi_1 - f_2 \cos \phi_2 - f_3 \cos \phi_3 + f_4 \cos \phi_4}{m} - \dot{\theta} X_m \quad (\text{A.10})$$

$$\ddot{\theta} = \frac{(f_1 + f_2 + f_3 + f_4)}{J_w} L \quad (\text{A.11})$$

Assim,

$$M_m \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} + M_\theta \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = M_f \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} \quad (\text{A.12})$$

Em que,

$$M_\theta = \begin{bmatrix} 0 & -\dot{\theta}_m & 0 \\ \dot{\theta}_m & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} M_f = \begin{bmatrix} -\sin \phi_1 & -\sin \phi_2 & \sin \phi_3 & \sin \phi_4 \\ \cos \phi_1 & -\cos \phi_2 & -\cos \phi_3 & \cos \phi_4 \\ L & L & L & L \end{bmatrix}$$

### A.1.5 Modelo dos atuadores

$$u = L_a \frac{di_a}{dt} + R_a i_a + e \quad (\text{A.13})$$

$$J_o \dot{W}_m + b_0 W_m + \frac{r}{n} f = \tau \quad (\text{A.14})$$

$$\tau = K + i_a \quad (\text{A.15})$$

$$e = K_{em} W_m \quad (\text{A.16})$$

Como  $\frac{di_a}{dt}$  é aproximadamente igual a 0, pois a constante de tempo elétrica é muito menor que a constante de tempo mecânica do motor. Dessa forma ela é desconsiderada. Logo,

$$u = R_a i_a + K_{em} W_m \quad (\text{A.17})$$

Realizando a substituição, têm-se que

$$J_o \dot{W}_m + b_0 W_m + \frac{r}{n} f = k_t \left( \frac{u - K_{em} W_m}{R_a} \right) \rightarrow \frac{R_a}{k_t} J_o \dot{W}_m + \frac{R_a}{k_t} b_0 W_m + \frac{R_a r}{n k_t} f = u - K_{em} W_m$$

$$u = \frac{R_a}{k_t} J_o \dot{W}_m + \frac{R_a}{k_t} b_0 W_m + \frac{R_a r}{k_t n} f + K_{em} W_m \quad (\text{A.18})$$

$$k + u - K_{em} W_m = R_a J_o \dot{W}_m + R_a b_0 W_m + R_a \frac{r}{n} f \rightarrow \frac{k_t}{R_a} u - \frac{K_{em}}{R_a} W_m = J_o \dot{W}_m + b_0 W_m + \frac{r}{n} f$$

$$\frac{k_t}{R_a} u - \frac{K_{em}}{R_a} W_m - J_o \dot{W}_m - b_0 W_m = \frac{r}{n} f \rightarrow f = \frac{k_t}{R_a} \frac{n}{r} u - \frac{K_{em}}{R_a} \frac{n}{r} W_m - \frac{n}{r} J_o \dot{W}_m - \frac{n}{r} b_0 W_m$$

Observe que os motores são acionados por uma largura de pulso  $\rho = \frac{u}{V_{ref}}$ , logo  $u = \rho V_{ref}$ . Dessa forma,

$$f = \frac{k_t}{R_a} \frac{n}{r} \rho V_{ref} - \frac{K_{em}}{R_a} \frac{n}{r} W_m - \frac{n}{r} J_0 \dot{W}_m - \frac{n}{r} b_0 W_m \quad (A.19)$$

Assim a dinâmica dos atuadores do robô é dada por,

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{-n}{r} J_0 \begin{bmatrix} \dot{W}_{m1} \\ \dot{W}_{m2} \\ \dot{W}_{m3} \\ \dot{W}_{m4} \end{bmatrix} - \frac{n}{r} \left( \frac{k_{em}}{R_a} + b_0 \right) \begin{bmatrix} W_{m1} \\ W_{m2} \\ W_{m3} \\ W_{m4} \end{bmatrix} + \frac{n}{r} \frac{k_t}{R_a} V_{ref} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (A.20)$$

### A.1.6 Modelo Dinâmico da Base Móvel com Inclusão dos Atuadores

$$\begin{bmatrix} W_{m1} \\ W_{m2} \\ W_{m3} \\ W_{m4} \end{bmatrix} = \frac{n}{r} D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} \rightarrow \begin{bmatrix} \dot{W}_{m1} \\ \dot{W}_{m2} \\ \dot{W}_{m3} \\ \dot{W}_{m4} \end{bmatrix} = \frac{n}{r} D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix}$$

$$\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \end{bmatrix} = \frac{-n^2}{r^2} J_0 D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} - \frac{n^2}{r^2} D \left( b_0 + \frac{k_t}{R_a} K_{em} \right) \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \frac{n}{r} \frac{k_t}{R_a} V_{ref} \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (A.21)$$

Fazendo

$$\beta = \frac{n}{r} \frac{k_t}{R_a} V_{ref} \quad X = \frac{n^2}{r^2} J_0 \quad \phi = \frac{n^2}{r^2} \left( b_0 + \frac{k_t}{R_a} K_{em} \right).$$

Realizando as substituições,

$$M_m \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} + M_\theta \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} = M_f \left[ -X D \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} - \phi D \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + \beta \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \right]$$

Em que,

$$M_m = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & J_w \end{bmatrix} \quad M_\theta = \begin{bmatrix} 0 & -\dot{\theta}_m & 0 \\ \dot{\theta}_m & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$M_f = \begin{bmatrix} -\sin \phi_1 & -\sin \phi_2 & \sin \phi_3 & \sin \phi_4 \\ \cos \phi_1 & -\cos \phi_2 & -\cos \phi_3 \cos \phi_4 & \\ L & L & L & L \end{bmatrix}$$

Dessa forma a formulação do modelo é dado por:

$$M_1 \begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = M_2 \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + M_3 \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (A.22)$$

E o modelo resultante, desenvolvido em [Soa11], é dado por:

$$\begin{bmatrix} \ddot{X}_m \\ \ddot{Y}_m \\ \ddot{\theta} \end{bmatrix} = M_1^{-1} M_2 \begin{bmatrix} \dot{X}_m \\ \dot{Y}_m \\ \dot{\theta} \end{bmatrix} + M_1^{-1} M_3 \begin{bmatrix} \rho_1 \\ \rho_2 \\ \rho_3 \\ \rho_4 \end{bmatrix} \quad (\text{A.23})$$

Em que,

$$M_1 = \begin{bmatrix} -(Xe_1 + m) & -Xe_2 & -Xe_3 \\ -Xe_2 & -(Xe_4 + m) & -Xe_5 \\ -Xe_3 & -Xe_5 & -(4L^2X + J_w) \end{bmatrix} \quad M_2 = \begin{bmatrix} \phi e_1 & (\psi e_2 - \dot{\theta}_m) & \psi e_3 \\ (\phi e_2 + \dot{\theta}_m) & \phi e_4 & \psi e_5 \\ \psi e_3 & \psi e_5 & 4L^2\phi \end{bmatrix}$$

$$M_3 = \begin{bmatrix} \beta \sin \phi_1 & \beta \sin \phi_2 & -\beta \sin \phi_3 \\ -\beta \cos \phi_1 & \beta \cos \phi_2 & \beta \cos \phi_3 \\ -\beta L & -\beta L & -\beta L \end{bmatrix}$$

## Ferramentas

### B.1 Álgebra de Lie

A álgebra de Lie trata-se de uma estrutura algébrica de uso principal no estudo dos chamados grupos de Lie e das variedades diferenciáveis [Mar14]. O objetivo das álgebras de Lie é ser uma ferramenta para o estudo das rotações infinitesimais. Duas álgebras de Lie muito conhecidas e utilizadas neste trabalho, colchete de Lie e derivada de Lie são apresentadas nas seções seguintes.

#### B.1.1 Colchete de Lie

Seja  $\mathfrak{g}$  um tipo de álgebra de Lie sobre um corpo, ou seja, corresponde a um espaço vetorial sobre um corpo  $F$  juntamente com uma operação binária  $([\cdot, \cdot] : \mathfrak{g} \times \mathfrak{g} \rightarrow \mathfrak{g})$ , conhecida como colchete de Lie, que satisfaz os seguintes axiomas [Wik14b]:

- Bilinearidade:  
 $[ax + by, z] = a[x, z] + b[y, z], \quad [z, ax + by] = a[z, x] + b[z, y]$  para todos escalares  $a, b$  em  $F$  e todos elementos  $x, y, z$  em  $\mathfrak{g}$ .
- Anticomutatividade:  
 $[x, y] = -[y, x]$ , para todos elementos  $x, y$  em  $\mathfrak{g}$ .
- A identidade de Jacobi:  
 $[x, [y, z]] + [y, [z, x]] + [z, [x, y]] = 0$  para todo  $x, y, z$  em  $\mathfrak{g}$ .

O colchete de Lie de  $L(A)$  é definido como sendo o seu comutador em  $A$ :  $[a, b] = a * b - b * a$ . A álgebra associativa das matrizes  $n' \times n$  sobre um corpo  $F$  dá origem ao grupo linear geral  $\mathfrak{gl}_n(F)$ . A álgebra associativa  $A$  é chamada de uma álgebra envolvente da álgebra de Lie  $L(A)$ .

#### B.1.2 Derivada de Lie

A derivada de Lie é uma derivação na álgebra de funções diferenciais sobre uma variação diferenciável  $\mathcal{M}$ , cuja definição pode estender-se a álgebra de tensores de variação. Para definir a derivada de Lie sobre o conjunto de tensores do tipo  $(r, s)$  basta definir sua ação sobre funções

e sobre campos vetoriais. Assim, se  $x$  é um campo diferencial de vetores, define-se a derivada de Lie com relação a  $x$  como a única derivada tal que:

- $\mathcal{L}_x f = x(f)$ . Para toda função diferenciável  $f$ .
- $\mathcal{L}_x Y = [x, Y]$ . Para todo campo diferenciável  $Y$ . Onde  $[,]$  é o colchete de Lie.

A derivada assim definida satisfaz a regra do produto:

$$\mathcal{L}_X(S \otimes T) = (\mathcal{L}_X S) \otimes T + S \otimes (\mathcal{L}_X T). \quad (\text{B.1})$$

No espaço vetorial de todas as derivadas de Lie em  $M$ , formam uma álgebra de Lie de dimensão infinita com relação ao colchete de Lie [Wik14b].



## Código Fonte

### C.1 Algoritmo de busca de máximos e mínimos dos vértices

```

clc
clear all
% Variaveis de dinâmica do sistema
K_t = 0.0059; % Constante de torque dos atuadores
n = 19; % Fator de redução dos atuadores para as rodas
V_ref = 12; % Tensão de referência
R_a = 1.71; % Resistência da armadura dos atuadores
r = 0.024; % Raio da roda
I_o = 3.88E-7; % Momento de inércia dos atuadores
I_w = 2.125; % Momento de inércia do robô
b_o = 2.4E-6; % Constante de atrito viscoso dos atuadores
K_em = 0.0059; % Constante de força contra eletromotriz dos atuadores
L = 0.09; % Distância entre a roda e o centro do robô
m = 3.4; % Massa do robô
fi_d = 45; % Ângulo dianteiro do robô
fi_t = 45; % ângulo traseiro do robô
% Percentual de variação de cada parâmetro da planta
unct_K_t = 0.2;
unct_n = 0.2;
unct_V_ref = 0.2;
unct_R_a = 0.2;
unct_r = 0.2;
unct_I_o = 0.2;
unct_I_w = 0.2;
unct_b_o = 0.2;
unct_K_em = 0.2;
unct_L = 0.2;

```

```
unct_m      = 0.2;
unct_fi_d   = 0.2;
unct_fi_t   = 0.2;
% Numero de pontos com variacöes  $\frac{1}{2}$  e  $\frac{1}{2}$ 
pnts        = 2;
% Parâmetros com variacöes  $\frac{1}{2}$  e  $\frac{1}{2}$ 
Kt_max = K_t*(1 + unct_K_t);
Kt_min = K_t*(1 - unct_K_t);
Kt_nom = [K_t linspace(Kt_min, Kt_max, pnts)]
n_max   = n*(1 + unct_n);
n_min   = n*(1 - unct_n);
n_nom   = [n linspace(n_min, n_max, pnts)];
V_ref_max = V_ref*(1 + unct_V_ref);
V_ref_min = V_ref*(1 - unct_V_ref);
V_ref_nom = [V_ref linspace(V_ref_min, V_ref_max, pnts)];
R_a_max  = R_a*(1 + unct_R_a);
R_a_min  = R_a*(1 - unct_R_a);
R_a_nom  = [R_a linspace(R_a_min, R_a_max, pnts)];
r_max    = r*(1 + unct_r);
r_min    = r*(1 - unct_r);
r_nom    = [r linspace(r_min, r_max, pnts)];
I_o_max  = I_o*(1 + unct_I_o);
I_o_min  = I_o*(1 - unct_I_o);
I_o_nom  = [I_o linspace(I_o_min, I_o_max, pnts)];
I_w_max  = I_w*(1 + unct_I_w);
I_w_min  = I_w*(1 - unct_I_w);
I_w_nom  = [I_w linspace(I_w_min, I_w_max, pnts)];
b_o_max  = b_o*(1 + unct_b_o);
b_o_min  = b_o*(1 - unct_b_o);
b_o_nom  = [b_o linspace(b_o_min, b_o_max, pnts)];
K_em_max = K_em*(1 + unct_K_em);
K_em_min = K_em*(1 - unct_K_em);
K_em_nom = [K_em linspace(K_em_min, K_em_max, pnts)];
L_max    = L*(1 + unct_L);
L_min    = L*(1 - unct_L);
L_nom    = [L linspace(L_min, L_max, pnts)];
m_max    = m*(1 + unct_m);
m_min    = m*(1 - unct_m);
m_nom    = [m linspace(m_min, m_max, pnts)];
fi_d_max = fi_d*(1 + unct_fi_d);
fi_d_min = fi_d*(1 - unct_fi_d);
fi_d_nom = [fi_d linspace(fi_d_min, fi_d_max, pnts)];
```



```
fprintf('a1_max: %f a1_min: %f \n a2_max: %f a2_min: %f \n a3_max: %f a3_min: %f
\n', max(Vc_a1), min(Vc_a1), max(Vc_a2), min(Vc_a2), max(Vc_a3), min(Vc_a3));
disp('Finished')
```

## C.2 InitMIMO.m

```
clc;
clear all;
versao = '2014.03.12.01';
fprintf('\n===== \n');
fprintf('***** Relatório de execução do sistema MIMO com abordagem RMI para
robo F-180 ***** \n');
fprintf('Data: %s \n', date );
fprintf('Versão de software: %s \n', versao);
fprintf('\n===== \n');
fprintf('Etapas: \n 1) Simulação sem RMI, sem incertezas \n');
fprintf('2) Gera Politopo de 16 vértices \n');
fprintf('3) Calcula as novas A e B com incertezas \n');
fprintf('4) Simulação sem RMI, com incertezas \n');
fprintf('5) Calcula Ganhos pela LMI \n');
fprintf('6) Simulação com Ganhos RMI, Incertezas e K da alocação antiga \n ');
fprintf('7) Simulação com Ganhos RMI, Incertezas e novos ganhos K da
realimentação \n');
fprintf('\n===== \n');

% Percentual de divergência entre o sistema real e o modelado. Para fazer
% com que o modelo seja idêntico ao sistema real (planta), ou seja,
%  $F_m(x) = F(x)$  faça com que o erro percentual seja o menor possível. O uso
% de zero irá provocar falha na simulação.
erroPercentual = 0.000001;

% Define o ganho RMI
K_RMI = zeros(3);
%K_RMI = [2 -2 -2; -1 1 -1; -10 -10 20];
%K_RMI = [0.0464 -0.0039 -0.0033; -0.0040 0.0463 -0.0032; -0.0262 -0.0263 0.3131];
% Estados do sistema MIMO
syms Xm
syms Ym
syms dotTeta

% Variaveis de dinâmica do sistema
K_t = 0.0059; % Constante de torque dos atuadores
```

---

```

n = 19;           % Fator de redução dos atuadores para as rodas
V_ref = 12;      % Tensão de referência
R_a = 1.71;      % Resistência da armadura dos atuadores
r = 0.024;       % Raio da roda
I_o = 3.88E-7;   % Momento de inércia dos atuadores
I_w = 2.125;     % Momento de inércia do robô
b_o = 2.4E-6;    % Constante de atrito viscoso dos atuadores
K_em = 0.0059;   % Constante de força contra eletromotriz dos atuadores
L = 0.09;        % Distância entre a roda e o centro do robô
m = 3.4;         % Massa do robô
fi_d = 45;       % Ângulo dianteiro do robô
fi_t = 45;       % ângulo traseiro do robô

fprintf('\n===== \n');
fprintf('@[Etapa: 01/Passo: 1]: Calculando parâmetros das matrizes do modelo e
da planta\n');
%
[a1, a2, a3, b1, b2] = CalcParamMatr(K_t, n, V_ref, R_a, r, I_o, I_w, b_o, K_em,
L, m, fi_d, fi_t);
% Matriz An
An = [-a1 a2*dotTeta 0; -a2*dotTeta -a1 0; 0 0 -a3];
% Matriz Bn
Bn = [b1 0 0; 0 b1 0; 0 0 b2];
%
a1_m = a1;
a2_m = a2;
a3_m = a3;
b1_m = b1;
b2_m = b2;
% Matriz An_m
An_m = [-a1_m a2_m*dotTeta 0; -a2_m*dotTeta -a1_m 0; 0 0 -a3_m];
% Matriz Bn_m
Bn_m = [b1_m 0 0; 0 b1_m 0; 0 0 b2_m];
% Matrizes G e Gm (3x3)
G = [b1 0 0; 0 b1 0; 0 0 b2];
Gm = [b1_m 0 0; 0 b1_m 0; 0 0 b2_m];
%Gm = G; %fazendo planta igual ao modelo
InvGm = inv(Gm);

% Matriz C,
C = eye(3); % MIMO Case

```

```

fprintf('\n=====\\n');
fprintf('@[Etapas: 01/Passo: 2]: Calculando ganhos da realimentação estática de
estados\\n');
%
% Ganhos da realimentação estática de estados
Ap = [zeros(3) zeros(3); -C zeros(3)];
Bp = [Bn; zeros(3)];
An = [a1 a2 0; a2 a1 0; 0 0 a3];
%Kp = place(Ap, Bn, [-10 -20 -30])
%K = Kp;
%Kp = place(Ap, Bp, [-15 -20 -32 -41 -50 -65]);
%Ap = [An zeros(3); -C zeros(3)];
%Pd = [-14+7i -14-7i -15 -25 -30 -33];
%Kp = place(Ap, Bp, Pd);
Kp = place(Ap, Bp, [-70 -75 -80 -85 -90 -95]);
K = Kp(:,1:3);
Ka = Kp(:,4:6);

%Ganho K ajustado para valores nominais para o menor erro possível
% Entrada onda quadrada, 10seg de simulação
%K = [ 7.7551 0 0; 0 6.4975 0; 0 0 0.8755 ];

```

## C.3 Algoritmo Heurístico de sintonia do ganho $K_{IDR}$

### C.3.1 CalculatePolytopeNLMaxMin

```

function [Polytope] = CalculatePolytopeNL_MaxMin(a1_nom, a2_nom, a3_nom, b1_nom,
b2_nom, b3_nom, c1_nom, c2_nom, c3_nom)
%
%Cria as matrizes para armazenar os vertices
a1max = 31.001811;
a1min = 1.640326;
a2max = 0.959340;
a2min = 0.674413;
a3max = 1.511401;
a3min = 0.028910;
b1max = b1_nom;
b1min = b1_nom;
b2max = b2_nom;
b2min = b2_nom;
dotTeta_max = 30;
dotTeta_min = -30;
%

```

---

```

a1 = [a1min a1max];
a2 = [a2min a2max];
a3 = [a3min a3max];
b1 = [b1min b1max];
b2 = [b2min b2max];
dotTeta = [dotTeta_min dotTeta_max];
%
%
cnt = 1;
    for i = 1: 2
        for j = 1: 2
            for l = 1: 2
                for k = 1: 2
                    d1 = a1_nom - a1(j);
                    d2 = a2_nom - a2(l);
                    d3 = a3_nom - a3(k);
                    dTeta = 0 - dotTeta(i);
                    %Va{cnt} = [d1 -d2*dTeta 0; d2*dTeta d1 0; 0 0 d3];
                    Va{cnt} = [(a1_nom - d1) -(a2_nom - d2)*dTeta 0; (a2_nom -
d2)*dTeta (a1_nom - d1) 0; 0 0 (a3_nom - d3)];
                    Vb{cnt} = [b1(1) 0 0; 0 b1(1) 0; 0 0 b2(1)];
                    Polytope{cnt} = struct('A', Va{cnt}, 'B', Vb{cnt}, 'C',
eye(3), 'a1', a1(j), 'a2', a2(l), 'a3', a3(k), 'dotTeta', dotTeta(i), 'd1', d1,
'd2', d2, 'd3', d3, 'dTeta', dTeta, 'b1', b1(1), 'b2', b2(1) );
                    cnt = cnt + 1;
                end
            end
        end
    end
end

```

### C.3.2 BuscaGanhoKIDRFinal.m

```
InitModelMIMO
```

```
UsarErroSidney = 0;
```

```
% Monta o Politopo
```

```
Polytope = CalculatePolytopeNL_MaxMin(a1_m, a2_m, a3_m, b1_m, b2_m, 0, 1, 1, 1);
```

```
% Determina um ganho RMI inicial
```

```
K_RMI_Inicial = 0.9*[0.0429 -0.0076 -0.0072; -0.0076 0.0428 -0.0072; -0.0504
-0.0505 0.2882];
```

```
K_RMI = K_RMI_Inicial;
K_RMI_Otimo = K_RMI;

% Executa interações para determinar os erros sem o uso do RMI
for v = 1:length(Polytope) %Varre os vértices
    % Carrega os valores com as incertezas do vértice
    a1 = Polytope{v}.a1;
    a2 = Polytope{v}.a2;
    a3 = Polytope{v}.a3;
    b1 = Polytope{v}.b1;
    b2 = Polytope{v}.b2;
    % Simula sem o RMI
    K_RMI = zeros(3);
    simulation = sim( 'RefatoracaoMod_MIMO_RMI','SaveOutput', 'on' );
    Erro_Xm_RMILess{v} = mean(abs(simulation.get('Erro_Xm')));
    Erro_Ym_RMILess{v} = mean(abs(simulation.get('Erro_Ym')));
    Erro_Theta_RMILess{v} = mean(abs(simulation.get('Erro_Theta')));
    Erro_Medio_RMILess{v} = mean(abs(simulation.get('Erro_Medio')));
end
% Executa interações para determinar o ganho RMI com menor erro
for v = 1:length(Polytope) %Varre os vértices
    K_RMI = K_RMI_Inicial;
    K_RMI_Otimo = K_RMI;
    % Carrega os valores com as incertezas do vértice
    a1 = Polytope{v}.a1;
    a2 = Polytope{v}.a2;
    a3 = Polytope{v}.a3;
    b1 = Polytope{v}.b1;
    b2 = Polytope{v}.b2;
    simulation = sim( 'RefatoracaoMod_MIMO_RMI','SaveOutput', 'on' );
    Erro_XmInicial = mean(abs(simulation.get('Erro_Xm')));
    Erro_YmInicial = mean(abs(simulation.get('Erro_Ym')));
    Erro_ThetaInicial = mean(abs(simulation.get('Erro_Theta')));
    Erro_MedioInicial = mean(abs(simulation.get('Erro_Medio')));
    ErroSetPointMedio = Erro_MedioInicial;
    if(UsarErroSidney)
        ErroSetPointMedio = sqrt((Erro_XmInicial^2) + (Erro_YmInicial^2));
    end
    ErroSetPointXm = Erro_XmInicial;
    ErroSetPointYm = Erro_YmInicial;
    ErroSetPointTheta = Erro_ThetaInicial;
    interacoes = 50;
```



---

```

for j = 1:interacoes
    stepXm = 0.01;
    stepYm = 0.01;
    stepTheta = 0.01;
    step = -0.001;
    %K_RMI = K_RMI + step;
    K_RMI(1) = K_RMI(1) + stepXm;
    K_RMI(5) = K_RMI(5) + stepYm;
    K_RMI(9) = K_RMI(9) + stepTheta;
%    K_RMI(2) = K_RMI(2) + step;
%    K_RMI(3) = K_RMI(3) + step;
%    K_RMI(4) = K_RMI(4) + step;
%    K_RMI(6) = K_RMI(6) + step;
%    K_RMI(7) = K_RMI(7) + step;
%    K_RMI(8) = K_RMI(8) + step;
    simulation = sim( 'RefatoracaoMod_MIMO_RMI', 'SaveOutput', 'on' );
    Erro_Xm = mean(abs(simulation.get('Erro_Xm')));
    Erro_Ym = mean(abs(simulation.get('Erro_Ym')));
    Erro_Theta = mean(abs(simulation.get('Erro_Theta')));
    Erro_Medio = mean(abs(simulation.get('Erro_Medio')));
    if(UsarErroSidney)
        Erro_Medio = sqrt((Erro_Xm^2) + (Erro_Ym^2));
    end
    if(Erro_Medio <= ErroSetPointMedio)
        K_RMI_Otimo = K_RMI;
        ErroSetPointMedio = Erro_Medio;
        fprintf('\n\n###Encontrado###\n\n\n');
%        K_RMI_OtimosVertice{v} = K_RMI_Otimo;
%        Erro_Medio_Vertices{v} = Erro_Medio;
%        Erro_Xm_RMI{v} = mean(abs(simulation.get('Erro_Xm')));
%        Erro_Ym_RMI{v} = mean(abs(simulation.get('Erro_Ym')));
%        Erro_Theta_RMI{v} = mean(abs(simulation.get('Erro_Theta')));
    else
        fprintf('\n\n###NÃO Encontrado###\n\n\n');
    end
    fprintf('### Interações: %d - %d ###', v, j);
end
K_RMI_OtimosVertice{v} = K_RMI_Otimo;
Erro_Medio_Vertices{v} = Erro_Medio;
Erro_Xm_RMI{v} = mean(abs(simulation.get('Erro_Xm')));
Erro_Ym_RMI{v} = mean(abs(simulation.get('Erro_Ym')));
Erro_Theta_RMI{v} = mean(abs(simulation.get('Erro_Theta')));

```

```
end
% Com os conjuntos de ganhos RMI de menor erro por vértices
% Escolhe-se os ganhos de menor erro e o maior erro encontrado
MaxErro_Encontrado = 0;
for v = 1:length(Erro_Medio_Vertices)
    Vec(v) = Erro_Medio_Vertices{v};
end
%[Maximo, IndMax] = max(Vec);
%[Minimo, IndMin] = min(Vec);
[MaxErro_Encontrado, indMaximo] = max(Vec);
[MinErro_Encontrado, indMinimo] = min(Vec);

% Testa todos os ganhos para encontrar o de menor erro
for s = 1:length(Polytope)
    K_RMI = K_RMI_OtimosVertice{s};
    ValidaVertices = [0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0];
    for v = 1:length(Polytope)
        a1 = Polytope{v}.a1;
        a2 = Polytope{v}.a2;
        a3 = Polytope{v}.a3;
        b1 = Polytope{v}.b1;
        b2 = Polytope{v}.b2;
        simulation = sim( 'RefatoracaoMod_MIMO_RMI', 'SaveOutput', 'on' );
        Erro_Medio = mean(abs(simulation.get('Erro_Medio')));
        Erro_Xm = mean(abs(simulation.get('Erro_Xm')));
        Erro_Ym = mean(abs(simulation.get('Erro_Ym')));
        Erro_Theta = mean(abs(simulation.get('Erro_Theta')));
        if(UsarErroSidney)
            Erro_Medio = sqrt((Erro_Xm^2) + (Erro_Ym^2));
        end
        if(Erro_Medio <= MaxErro_Encontrado)
            ValidaVertices(v) = 1;
            SalvaErro_Medio(v) = Erro_Medio;
            SalvaErro_Xm(v) = Erro_Xm;
            SalvaErro_Ym(v) = Erro_Ym;
            SalvaErro_Theta(v) = Erro_Theta;
            TMP_ValidaVertices{s,v} = 1;
        else
            ValidaVertices(v) = 0;
            TMP_ValidaVertices{s,v} = 0;
        end
        TMP_ValidaErroVertices{s,v} = Erro_Medio;
    end
end
```

```

end
%ErroMedido(s) = SalvaErro_Medio;
if(ValidaVertices)
    ValidaGanho(s) = 1;
    ErroMedido(s) = mean(SalvaErro_Medio);
else
    ValidaGanho(s) = 0;
end
end
fprintf('=====\\n');
fprintf('|   Ganho   |   Sem RMI   |   Max Erro   |   Erro Médio   |   Erro
Xm(RMI-SRMI)   |   Erro Ym(RMI-SRMI)   |   Erro Theta(RMI-SRMI)   |   Ganhos
|\\n');
fprintf('-----\\n');
for h = 1:length(Polytope)
    if(ValidaGanho(h))
        fprintf('%d   |   %f   |   %f   |   %f   |   %f-%f   |   %f-%f   |
%f-%f   | [ %f %f %f ; %f %f %f ; %f %f %f ]   |\\n', h, Erro_Medio_RMILess{h},
MaxErro_Encontrado, SalvaErro_Medio(h), SalvaErro_Xm(h), Erro_Xm_RMILess{h},
SalvaErro_Ym(h), Erro_Ym_RMILess{h}, SalvaErro_Theta(h), Erro_Theta_RMILess{h},
K_RMI_OtimosVertice{h}(1,1), K_RMI_OtimosVertice{h}(1,2), K_RMI_OtimosVertice{h}
(1,3), K_RMI_OtimosVertice{h}(2,1), K_RMI_OtimosVertice{h}(2,2),
K_RMI_OtimosVertice{h}(2,3), K_RMI_OtimosVertice{h}(3,1),
K_RMI_OtimosVertice{h}(3,2), K_RMI_OtimosVertice{h}(3,3));
        fprintf('-----\\n');
    end
end
end
save test.mat
fprintf('=====\\n');
fprintf('|   Vertice   |   Erro Sem RMI   |   Erro com RMI   |   |\\n');
fprintf('-----\\n');
for s = 1:length(ValidaGanho) %Varre as variações
    for v = 1:length(Polytope) %Varre os vértices
        fprintf('%d - %d |   %f   |   %f   |\\n', s, v,
Erro_Medio_RMILess{v}, TMP_ValidaErroVertices{s,v});
    end
end
end

```

### C.3.3 runGanhoOtimoIDR

```
InitModelMIMO
```

```
UsarErroSidney = 0;
```

```

%==== Graficos ====%
GraficoComRMI = 0;
GraficoSemRMI = 0;
GraficoComparativoRMI = 0;
GraficosVerticesRMI = 0;
RespDegrauPorEntrada = 0;
RespPorEntrada = 1;
TrajetoriaReta = 0;
TrajetoriaQuadrada = 0;
TrajetoriaCircular = 0;
TrajetoriaOito = 1;
%=====
%***** Ganhos calculados por Heuristica *****%
% 1) Usando ABS no erro e calculando por estado(2014_04_12_ABS_Estados_01.mat)
%   - KO = [ 0.1 0 0; 0 0.1 0; 0 0 1]
%K_RMI_Otimo = [ 0.100000 0.000000 0.000000 ; 0.500000 0.600000 0.500000 ;
14.800000 14.800000 15.800000 ]
% 2) Usando ABS no erro e calculando por matriz (2014_04_12_ABS_Matriz.mat)
%   - KO = [ 2.8 0 0; 0 2.8 0; 0 0 0.2]
%K_RMI_Otimo = [ 3.400000 0.600000 0.600000 ; 0.600000 3.400000 0.600000 ;
0.600000 0.600000 0.800000 ]
%Usando incremento somente na diagonal e Ko = 0.9*KLMI - Menor erro obtido
%em relação ao debaixo
%K_RMI_Otimo = [ 0.0486   -0.0068   -0.0065;  -0.0068   0.0485   -0.0065;
-0.0454   -0.0455   0.2694];
%Usando incrementos diferenciados para diagonal do resto e Ko = 0.9*KLMI
%Erro maior
%K_RMI_Otimo = [ 0.068610 -0.009840 -0.009480 ; -0.009840 0.068520 -0.009480 ;
-0.048360 -0.048450 0.289380 ];

%***** Ganhos calculados por LMI *****%
% 1) Usando alg 221
K_RMI_Otimo = [0.0429 -0.0076 -0.0072; -0.0076 0.0428 -0.0072; -0.0504 -0.0505 0.2882];
%K = [1.8445 0 0; 0 0.8384 0; 0 0 3.6150];
%
K_RMI = K_RMI_Otimo;

% Monta o Politopo
Polytope = CalculatePolytopeNL_MaxMin(a1_m, a2_m, a3_m, b1_m, b2_m, 0, 1, 1, 1);

simOutputsPolytopeRMI = cell(16);

```

```

simOutputsPolytopeRMILess = cell(16);
for i = 1:length(Polytope)
    a1 = Polytope{i}.a1;
    a2 = Polytope{i}.a2;
    a3 = Polytope{i}.a3;
    b1 = Polytope{i}.b1;
    b2 = Polytope{i}.b2;
    %***** Simulação com RMI Heuristico *****
    %K = [ 7.7551 0 0; 0 6.4975 0; 0 0 0.8755 ];
    %K_RMI = [0.05 -0.008 -0.008;-0.008 0.05 -0.008;-0.05 -0.05 2.6];
    %K_RMI = [ 0.0486    -0.0068    -0.0065;  -0.0068    0.0485    -0.0065;
-0.0454    -0.0455    0.2694];
    %***** Simulação com RMI-LMI Alg221 *****
    %K_RMI = [0.0429 -0.0076 -0.0072; -0.0076 0.0428 -0.0072; -0.0504 -0.0505
0.2882];
    %***** Simulação com RMI Heuristico *****
    %K_RMI = [ 0.0486    -0.0068    -0.0065;  -0.0068    0.0485    -0.0065; -0.0454
-0.0455    0.2694];
    % ***** Ganho Alg. Genetico *****
    K_RMI = [ 0.0472 -0.0084 -0.0079; -0.0084 0.0470 -0.00079; -0.0554 -0.0584
0.3175];
    %*****
    %***** SEM RMI *****
    %K_RMI = zeros(3);
    %*****
    simOutputsPolytopeRMI{i} = sim( 'RefatoracaoMod_MIMO_RMI','SaveOutput', 'on' );
    %*****
    %***** Simulação sem RMI *****
    K_RMI = zeros(3);
    simOutputsPolytopeRMILess{i} = sim( 'RefatoracaoMod_MIMO_RMI','SaveOutput',
'on' );
    %***** Simulação com RMI-LMI Alg221 *****
    %K_RMI = [ 0.068610 -0.009840 -0.009480 ; -0.009840 0.068520 -0.009480 ;
-0.048360 -0.048450 0.289380 ];
    %K = [1.8445 0 0; 0 0.8384 0; 0 0 3.6150];
    %simOutputsPolytopeRMI{i} = sim( 'RefatoracaoMod_MIMO_RMI','SaveOutput',
'on' );
end
%===== Carrega Simulações =====
%*****

%===== Plota as saídas em único gráfico =====

```

```
fig = 1;
%===== Com RMI =====
if(GraficoComRMI)
    for i = 1:length(simOutputsPolytopeRMI)
        ScopeData = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        figure(fig)
        hold on
        grid on
        subplot(311)
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'r');%dXm
        title(ScopeData.signals(1,5).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,2), 'm');%refXm
        subplot(312)
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
        title(ScopeData.signals(1,6).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,2), 'm');%refYm
        subplot(313)
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'k');%dTheta
        title(ScopeData.signals(1,7).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,2), 'm');%refTheta
        title('Com RMI');
    end
end
%===== Sem RMI =====
if(GraficoSemRMI)
    fig = fig+1;
    for i = 1:length(simOutputsPolytopeRMILess)
        ScopeData = simOutputsPolytopeRMILess{i}.get('ScopeDataCompleto');
        figure(fig)
        hold on
        grid on
        subplot(311)
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'r');%dXm
        title(ScopeData.signals(1,5).label);
        hold on
        grid on
```

---

```

    plot(ScopeData.time, ScopeData.signals(1,5).values(:,2), 'm');%refXm
    subplot(312)
    plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
    title(ScopeData.signals(1,6).label);
    hold on
    grid on
    plot(ScopeData.time, ScopeData.signals(1,6).values(:,2), 'm');%refYm
    subplot(313)
    plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'k');%dTheta
    title(ScopeData.signals(1,7).label);
    hold on
    grid on
    plot(ScopeData.time, ScopeData.signals(1,7).values(:,2), 'm');%refTheta
    title('Sem RMI');
end
end
%===== Comparativo de com e sem RMI =====
if(GraficoComparativoRMI)
    fig = fig+1;
    for i = 1:length(simOutputsPolytopeRMILess)
        ScopeDataRMI = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        ScopeData = simOutputsPolytopeRMILess{i}.get('ScopeDataCompleto');
        figure(fig)
        title('Comparativo da saída sem RMI e com RMI');
        hold on
        grid on
        subplot(321)
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'r');%dXm
        title('Resposta  $\dot{X}_m$  sem RMI', 'interpreter', 'latex');
        %title(ScopeData.signals(1,5).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,2), 'm');%refXm
        subplot(323)
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
        title('Resposta  $\dot{Y}_m$  sem RMI', 'interpreter', 'latex');
        %title(ScopeData.signals(1,6).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,2), 'm');%refYm
        subplot(325)
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'k');%dTheta

```

---

```

    title('Resposta $\dot{\theta}$ sem RMI', 'interpreter', 'latex');
    %title(ScopeData.signals(1,7).label);
    hold on
    grid on
    plot(ScopeData.time, ScopeData.signals(1,7).values(:,2), 'm');%refTheta
    hold on
    grid on
    %
    subplot(322)
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,5).values(:,1), 'r');%dXm
    title('Resposta $\dot{X}_m$ com RMI', 'interpreter', 'latex');
    %title(ScopeDataRMI.signals(1,5).label);
    hold on
    grid on
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,5).values(:,2), 'm');%refXm
    subplot(324)
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,6).values(:,1), 'b');%dYm
    title('Resposta $\dot{Y}_m$ com RMI', 'interpreter', 'latex');
    %title(ScopeDataRMI.signals(1,6).label);
    hold on
    grid on
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,6).values(:,2), 'm');%refYm
    subplot(326)
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,7).values(:,1), 'k');%dTheta
    title('Resposta $\dot{\theta}$ com RMI', 'interpreter', 'latex');
    %title(ScopeDataRMI.signals(1,7).label);
    hold on
    grid on
    plot(ScopeDataRMI.time, ScopeDataRMI.signals(1,7).values(:,2), 'm');%refTheta
    %title(ScopeDataRMI.signals(1,7).label);
end
end
%=====
%===== Plota as respostas para cada vértice =====
if(GraficosVerticesRMI)
    fig = fig+1;
    for i = 1:length(simOutputsPolytopeRMI)
        ScopeData = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        figure(fig)
        hold on
        grid on
        subplot(711)

```



---

```

plot(ScopeData.time, ScopeData.signals(1,1).values(:,1), 'r');%ddotXm
title(ScopeData.signals(1,1).label);
hold on
grid on
subplot(711)
plot(ScopeData.time, ScopeData.signals(1,1).values(:,2), 'b');%ddotYm
subplot(711)
plot(ScopeData.time, ScopeData.signals(1,1).values(:,3), 'k');%ddotTheta
subplot(712)
plot(ScopeData.time, ScopeData.signals(1,2).values(:,1), 'r');%u1
title(ScopeData.signals(1,2).label);
hold on
grid on
plot(ScopeData.time, ScopeData.signals(1,2).values(:,2), 'b');%u2
plot(ScopeData.time, ScopeData.signals(1,2).values(:,3), 'k');%u3
subplot(713)
plot(ScopeData.time, ScopeData.signals(1,3).values(:,1), 'r');%Delta_c1
title(ScopeData.signals(1,3).label);
hold on
grid on
plot(ScopeData.time, ScopeData.signals(1,3).values(:,2), 'b');%Delta_c2
plot(ScopeData.time, ScopeData.signals(1,3).values(:,3), 'k');%Delta_c3
subplot(714)
plot(ScopeData.time, ScopeData.signals(1,4).values(:,1), 'r');%v1
title(ScopeData.signals(1,4).label);
hold on
grid on
plot(ScopeData.time, ScopeData.signals(1,4).values(:,2), 'b');%v2
plot(ScopeData.time, ScopeData.signals(1,4).values(:,3), 'k');%v3
subplot(715)
plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'r');%dXm
title(ScopeData.signals(1,5).label);
hold on
grid on
plot(ScopeData.time, ScopeData.signals(1,5).values(:,2), 'm');%refXm
subplot(716)
plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
title(ScopeData.signals(1,6).label);
hold on
grid on
plot(ScopeData.time, ScopeData.signals(1,6).values(:,2), 'm');%refYm
subplot(717)

```

```

        plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'k');%dTheta
        title(ScopeData.signals(1,7).label);
        hold on
        grid on
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,2), 'm');%refTheta
        %
        fig = fig+1;
    end
end
%=====
%***** Plota Resp Entrada *****%
if(RespDegrauPorEntrada)
    for i = 1:length(simOutputsPolytopeRMI)
        ScopeData = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        figure(101)
        hold on
        grid on
        %subplot(131)
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'b');%dXm
        title('Resposta $\dot{X}_m$', 'interpreter', 'latex');
        hold on
        grid on
        handle = plot(ScopeData.time, ScopeData.signals(1,5).values(:,2), 'r-.');%refXm
        set(handle,'LineWidth',[2]);
        ylabel('Amplitude');
        xlabel('Tempo (s)');
        %subplot(132)
        figure(102)
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
        title('Resposta $\dot{Y}_m$', 'interpreter', 'latex');
        hold on
        grid on
        handle = plot(ScopeData.time, ScopeData.signals(1,6).values(:,2), 'r-.');%refYm
        set(handle,'LineWidth',[2]);
        ylabel('Amplitude');
        xlabel('Tempo (s)');
        %subplot(133)
        figure(103)
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'b');%dTheta
        title('Resposta $\dot{\theta}$', 'interpreter', 'latex');
        hold on
        grid on
    end
end

```

---

```

        handle = plot(ScopeData.time, ScopeData.signals(1,7).values(:,2),
        'r-.');%refTheta
        set(handle,'LineWidth',[2]);
        ylabel('Amplitude');
        xlabel('Tempo (s)');
    end
end
%=====
%***** Plota Resp Saídas *****%
if(RespPorEntrada)
    for i = 1:length(simOutputsPolytopeRMI)
        ScopeData = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        figure(101)
        hold on
        grid on
        %subplot(131)
        plot(ScopeData.time, ScopeData.signals(1,5).values(:,1), 'b');%dXm
        title('Resposta $\dot X_m$', 'interpreter', 'latex');
        hold on
        grid on
        handle = plot(ScopeData.time, ScopeData.signals(1,5).values(:,2),
        'r-.');%refXm
        set(handle,'LineWidth',[2]);
        ylabel('$\dot X_m$ [m/s]', 'interpreter', 'latex');
        xlabel('Tempo [s]', 'interpreter', 'latex');
        %subplot(132)
        figure(102)
        plot(ScopeData.time, ScopeData.signals(1,6).values(:,1), 'b');%dYm
        title('Resposta $\dot Y_m$', 'interpreter', 'latex');
        hold on
        grid on
        handle = plot(ScopeData.time, ScopeData.signals(1,6).values(:,2),
        'r-.');%refYm
        set(handle,'LineWidth',[2]);
        ylabel('$\dot Y_m$ [m/s]', 'interpreter', 'latex');
        xlabel('Tempo [s]', 'interpreter', 'latex');
        %subplot(133)
        figure(103)
        plot(ScopeData.time, ScopeData.signals(1,7).values(:,1), 'b');%dTheta
        title('Resposta $\dot \theta$', 'interpreter', 'latex');
        hold on
        grid on
    end
end

```

---

```

        handle = plot(ScopeData.time, ScopeData.signals(1,7).values(:,2),
'r-.');%refTheta
        set(handle,'LineWidth',[2]);
        ylabel('$\dot{\theta}$ [graus]', 'interpreter', 'latex');
        xlabel('Tempo [s]', 'interpreter', 'latex');
    end
end
%=====
%***** Plota Trajetória Reta *****
if(TrajectoriaReta)
    fig = fig+1;
    for i = 1:16
        ScopeData = simOutputsPolytopeRMI{i}.get('ScopeDataCompleto');
        ScopeData_1 = simOutputsPolytopeRMI{i}.get('ScopeData1');
        ScopeData_2 = simOutputsPolytopeRMI{i}.get('ScopeDataPosicoes');
        %
        figure(fig)
        hold on
        grid on
        handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
        set(handle,'LineWidth',[2]);
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
        title('Trajetória Retilínea');
        ylabel('$Y_m$ [m]', 'interpreter', 'latex');
        xlabel('$X_m$ [m]', 'interpreter', 'latex');
        legend('Referência', 'Resposta');
        figure(fig+1)
        hold on
        grid on
        handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
        set(handle,'LineWidth',[2]);
        title('Referencia');
        figure(fig+2)
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');

```

---

```

        title('Resposta Trajetória');
    end
end
%***** Plota Trajetória Quadrada *****
if(TrajectoriaQuadrada)
    fig = fig+1;
    for i = 1:16
        ScopeData_1 = simOutputsPolytopeRMI{i}.get('ScopeData1');
        ScopeData_2 = simOutputsPolytopeRMI{i}.get('ScopeDataPosicoes');
        %
        figure(fig)
        hold on
        grid on
        handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
        set(handle,'LineWidth',[2])
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
        title('Trajetória Quadrada');
        ylabel('$Y_m$ [m]', 'interpreter', 'latex');
        xlabel('$X_m$ [m]', 'interpreter', 'latex');
        legend('Referência','Resposta')
        figure(fig+1)
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r');
        title('Referencia');
        figure(fig+2)
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
        title('Resposta Trajetória');
    end
end
%***** Plota Trajetória Circular *****
if(TrajectoriaCircular)
    fig = fig+1;
    for v = 1:16

```

---

```

ScopeData_1 = simOutputsPolytopeRMI{v}.get('ScopeData1');
ScopeData_2 = simOutputsPolytopeRMI{v}.get('ScopeDataPosicoes');
%
figure(fig)
hold on
grid on
handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
set(handle,'LineWidth',[2]);
hold on
grid on
plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
title('Trajetória Circular');
ylabel('$Y_m$ [m]', 'interpreter', 'latex');
xlabel('$X_m$ [m]', 'interpreter', 'latex');
legend('Referência', 'Resposta');
figure(fig+1)
hold on
grid on
handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
set(handle,'LineWidth',[2]);
title('Referencia');
figure(fig+2)
hold on
grid on
plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
title('Resposta Trajetória');
end
end
%*****
%***** Plota Trajetória em Oito *****
if(TrajectoriaOito)
    fig = fig+1;
    for i = 1:16
        %ScopeData_1 = simOutputsPolytopeRMI{i}.get('ScopeData1');
        ScopeData_2 = simOutputsPolytopeRMI{i}.get('ScopeDataPosicoes');
        %
        figure(fig)
        hold on

```

---

```

        grid on
        handle = plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r-.');
        set(handle,'LineWidth',[2]);
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
        %infinito
        %plot(ScopeData_2.signals(1,2).values(:,1),
ScopeData_2.signals(1,1).values(:,1), 'b');
        title('Trajetória em Oito');
        ylabel('$Y_m$ [m]', 'interpreter', 'latex');
        xlabel('$X_m$ [m]', 'interpreter', 'latex');
        legend('Referência','Resposta')
        %fig = fig+1;
        figure(fig+1)
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,2),
ScopeData_2.signals(1,2).values(:,2), 'r');
        title('Referencia');
        %fig = fig+1;
        figure(fig+2)
        hold on
        grid on
        plot(ScopeData_2.signals(1,1).values(:,1),
ScopeData_2.signals(1,2).values(:,1), 'b');
        title('Resposta Trajetória');
        end
end
%*****
fprintf('=====\\n');
fprintf('| Vértice | Erro Médio S/RMI | Erro Médio C/RMI |
Erro XM (S/RMI-C/RMI) | Erro YM (S/RMI-C/RMI) | Erro Theta
(S/RMI-C/RMI)
|\\n');
fprintf('-----\\n');
for i = 1:length(Polytope)
    Erro_Xm_RMI = mean(abs(simOutputsPolytopeRMI{i}.get('Erro_Xm')));
    Erro_Ym_RMI = mean(abs(simOutputsPolytopeRMI{i}.get('Erro_Ym')));
    Erro_Theta_RMI = mean(abs(simOutputsPolytopeRMI{i}.get('Erro_Theta')));

```

```

    Erro_Medio_RMI = mean(abs(simOutputsPolytopeRMI{i}.get('Erro_Medio')));
    if(UsarErroSidney)
        %Erro_Medio_RMI = sqrt((Erro_Xm_RMI^2) + (Erro_Ym_RMI^2));
    end
    %
    Erro_Xm_RMILess = mean(abs(simOutputsPolytopeRMILess{i}.get('Erro_Xm')));
    Erro_Ym_RMILess = mean(abs(simOutputsPolytopeRMILess{i}.get('Erro_Ym')));
    Erro_Theta_RMILess = mean(abs(simOutputsPolytopeRMILess{i}.get('Erro_Theta')));
    Erro_Medio_RMILess = mean(abs(simOutputsPolytopeRMILess{i}.get('Erro_Medio')));
    if(UsarErroSidney)
        %Erro_Medio_RMILess = sqrt((Erro_Xm_RMILess^2) + (Erro_Ym_RMILess^2));
    end
    fprintf('|      %d      |      %f      |      %f      |      %f-%f      |
%f-%f      |      %f-%f      |', i, Erro_Medio_RMILess, Erro_Medio_RMI,
Erro_Xm_RMILess, Erro_Xm_RMI, Erro_Ym_RMILess, Erro_Ym_RMI, Erro_Theta_RMILess,
Erro_Theta_RMI);
    fprintf('\n-----\n');
end

```

## C.4 Algoritmo Genético de sintonia do ganho $K_{IDR}$

### C.4.1 BuscaIDRGenetica

```

function erro = BuscaRMIGenetica(gain)
global Polytope
global K_RMI
global a1
global a2
global a3
global b1
global b2
global Erro_variacao
K_RMI = [gain(1) gain(4) gain(7); gain(2) gain(5) gain(8); gain(3) gain(6) gain(9)]
for i = 4:4
    for v = 1:length(Polytope)
        a1 = Polytope{v}.a1;
        a2 = Polytope{v}.a2;
        a3 = Polytope{v}.a3;
        b1 = Polytope{v}.b1;
        b2 = Polytope{v}.b2;
        simulation = sim( 'RefatoracaoMod_MIMO_RMI', 'SaveOutput', 'on' );
        Erro_Xm(v) = mean(abs(simulation.get('Erro_Xm')));
        Erro_Ym(v) = mean(abs(simulation.get('Erro_Ym')));
    end
end

```



---

```

        Erro_Theta(v) = mean(abs(simulation.get('Erro_Theta')));
        Erro_vertice(v) = sqrt(power(Erro_Xm(v),2) + power(Erro_Ym(v),2));
    end
    Erro_variacao(i) = mean(Erro_vertice)
end
erro = mean(Erro_variacao)
%     if(erro_s < 0)
%         erro = 100;
%     else
%         erro = erro_s;
%     end
    fprintf('=====\n')
    fprintf('Erro: %f \n', erro);
    fprintf('=====\n')
end

```

#### C.4.2 runBuscaIDRGenetica

```

InitModelMIMO
global K_RMI
global Polytope
K_RMI = double(zeros(3));
global Erro_variacao
Polytope = CalculatePolytopeNL_MaxMin(a1_m, a2_m, a3_m, b1_m, b2_m, 0, 1, 1, 1);

LB = [0 0 0 0 0 0 0 0 0]';
gains = 1.1*[0.0429 -0.0076 -0.0072; -0.0076 0.0428 -0.0072; -0.0504 -0.0505 0.2882];
options = optimset('LargeScale','off', 'MaxIter', 10, 'TolX', 1e-4);
erro = fminsearch('BuscaRMIGenetica', gains)

```

#### C.4.3 CalculatePolytopeNLMaxMin

```

function [Polytope] = CalculatePolytopeNL_MaxMin(a1_nom, a2_nom, a3_nom, b1_nom,
b2_nom, b3_nom, c1_nom, c2_nom, c3_nom)
%
%Cria as matrizes para armazenar os vertices
a1max = 31.001811;
a1min = 1.640326;
a2max = 0.959340;
a2min = 0.674413;
a3max = 1.511401;
a3min = 0.028910;
b1max = b1_nom;

```

```
b1min = b1_nom;
b2max = b2_nom;
b2min = b2_nom;
dotTeta_max = 30;
dotTeta_min = -30;
%
a1 = [a1min a1max];
a2 = [a2min a2max];
a3 = [a3min a3max];
b1 = [b1min b1max];
b2 = [b2min b2max];
dotTeta = [dotTeta_min dotTeta_max];
%
%
cnt = 1;
    for i = 1: 2
        for j = 1: 2
            for l = 1: 2
                for k = 1: 2
                    d1 = a1_nom - a1(j);
                    d2 = a2_nom - a2(l);
                    d3 = a3_nom - a3(k);
                    dTeta = 0 - dotTeta(i);
                    %Va{cnt} = [d1 -d2*dTeta 0; d2*dTeta d1 0; 0 0 d3];
                    Va{cnt} = [(a1_nom - d1)
-(a2_nom - d2)*dTeta 0;
(a2_nom - d2)*dTeta (a1_nom - d1) 0;
0 0 (a3_nom - d3)];
                    Vb{cnt} = [b1(1) 0 0; 0 b1(1) 0; 0 0 b2(1)];
                    Polytope{cnt} = struct('A', Va{cnt}, 'B', Vb{cnt}, 'C',
eye(3), 'a1', a1(j), 'a2', a2(l), 'a3', a3(k), 'dotTeta', dotTeta(i),
'd1', d1, 'd2', d2, 'd3', d3, 'dTeta', dTeta, 'b1', b1(1), 'b2', b2(1) );
                    cnt = cnt + 1;
                end
            end
        end
    end
end
```

## Resultados Das Simulações

### D.1 Tabelas de resultados dos casos de estudos

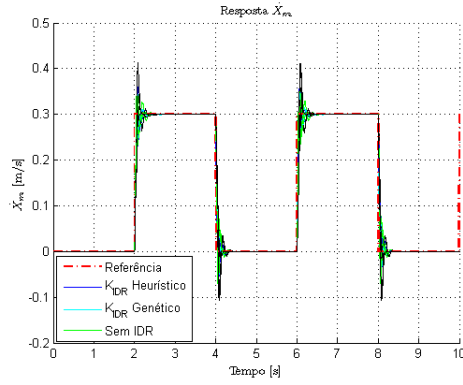
#### D.1.1 Resposta para Degrau Unitário

Tabela D.1: Erros dos vértices para entrada em Degrau

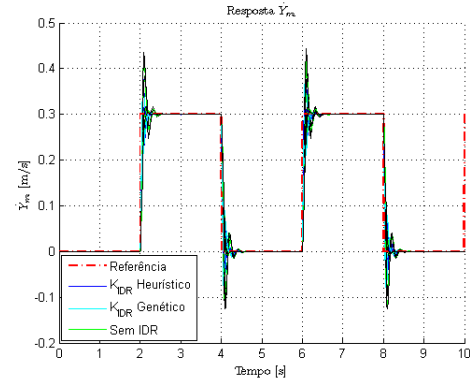
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.139519	<b>0.100438</b>	0.102321
1	$e_{vx}$	0.092197	<b>0.065483</b>	0.066582
1	$e_{vy}$	0.104715	<b>0.076156</b>	0.077694
1	$e_{\dot{\theta}}$	0.040988	<b>0.040242</b>	0.041813
2	$ e_v $	0.140619	<b>0.096722</b>	0.100379
2	$e_{vx}$	0.092783	<b>0.063061</b>	0.065441
2	$e_{vy}$	0.105665	<b>0.073338</b>	0.076114
2	$e_{\dot{\theta}}$	0.067528	<b>0.038325</b>	0.040788
3	$ e_v $	0.139132	<b>0.099992</b>	0.101759
3	$e_{vx}$	0.092333	<b>0.065607</b>	0.066717
3	$e_{vy}$	0.104079	<b>0.075459</b>	0.076836
3	$e_{\dot{\theta}}$	0.066260	<b>0.040231</b>	0.041801
4	$ e_v $	0.140125	<b>0.097677</b>	0.100065
4	$e_{vx}$	0.092924	<b>0.064077</b>	0.065593
4	$e_{vy}$	0.104881	<b>0.073721</b>	0.075569
4	$e_{\dot{\theta}}$	0.067528	<b>0.038967</b>	0.040776
5	$ e_v $	0.134017	<b>0.089005</b>	0.091104
5	$e_{vx}$	0.089445	<b>0.057865</b>	0.059317
5	$e_{vy}$	0.099800	<b>0.067627</b>	0.069148
5	$e_{\dot{\theta}}$	0.068501	<b>0.040958</b>	0.042761
6	$ e_v $	0.135453	<b>0.085926</b>	0.090289
6	$e_{vx}$	0.090403	<b>0.055675</b>	0.058699
6	$e_{vy}$	0.100870	<b>0.065449</b>	0.068605
6	$e_{\dot{\theta}}$	0.069813	<b>0.039201</b>	0.042194
7	$ e_v $	0.134162	<b>0.089115</b>	0.091213
7	$e_{vx}$	0.089450	<b>0.057862</b>	0.059320
7	$e_{vy}$	0.099991	<b>0.067775</b>	0.069288
7	$e_{\dot{\theta}}$	0.068501	<b>0.040965</b>	0.042769
8	$ e_v $	0.135598	<b>0.086048</b>	0.090407
8	$e_{vx}$	0.090413	<b>0.055656</b>	0.058696
8	$e_{vy}$	0.101056	<b>0.065625</b>	0.068761
8	$e_{\dot{\theta}}$	0.069813	<b>0.039213</b>	0.042205

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.139519	<b>0.100438</b>	0.102321
9	$e_{vx}$	0.092197	<b>0.065483</b>	0.066582
9	$e_{vy}$	0.104715	<b>0.076156</b>	0.077694
9	$e_{\dot{\theta}}$	0.066260	<b>0.040242</b>	0.041813
10	$ e_v $	0.140619	<b>0.096722</b>	0.100379
10	$e_{vx}$	0.092783	<b>0.063061</b>	0.065441
10	$e_{vy}$	0.105665	<b>0.073338</b>	0.076114
10	$e_{\dot{\theta}}$	0.067528	<b>0.038325</b>	0.040788
11	$ e_v $	0.139132	<b>0.099992</b>	0.101759
11	$e_{vx}$	0.092333	<b>0.065607</b>	0.066717
11	$e_{vy}$	0.104079	<b>0.075459</b>	0.076836
11	$e_{\dot{\theta}}$	0.066260	<b>0.040231</b>	0.041801
12	$ e_v $	0.140125	<b>0.097677</b>	0.100065
12	$e_{vx}$	0.092924	<b>0.064077</b>	0.065593
12	$e_{vy}$	0.104881	<b>0.073721</b>	0.075569
12	$e_{\dot{\theta}}$	0.067528	<b>0.038967</b>	0.040776
13	$ e_v $	0.134017	<b>0.089005</b>	0.091104
13	$e_{vx}$	0.089445	<b>0.057865</b>	0.059317
13	$e_{vy}$	0.099800	<b>0.067627</b>	0.069148
13	$e_{\dot{\theta}}$	0.068501	<b>0.040958</b>	0.042761
14	$ e_v $	0.135453	<b>0.085926</b>	0.090289
14	$e_{vx}$	0.090403	<b>0.055675</b>	0.058699
14	$e_{vy}$	0.100870	<b>0.065449</b>	0.068605
14	$e_{\dot{\theta}}$	0.069813	<b>0.039201</b>	0.042194
15	$ e_v $	0.134162	<b>0.089115</b>	0.091213
15	$e_{vx}$	0.089450	<b>0.057862</b>	0.059320
15	$e_{vy}$	0.099991	<b>0.067775</b>	0.069288
15	$e_{\dot{\theta}}$	0.068501	<b>0.040965</b>	0.042769
16	$ e_v $	0.135598	<b>0.086048</b>	0.090407
16	$e_{vx}$	0.090413	<b>0.055656</b>	0.058696
16	$e_{vy}$	0.101056	<b>0.065625</b>	0.068761
16	$e_{\dot{\theta}}$	0.069813	<b>0.039213</b>	0.042205

## D.1.2 Trajetória Reta Sem Rotação



(a) Resposta  $\dot{X}_m$  Trajetória Reta Sem Rotação



(b) Resposta  $\dot{Y}_m$  para trajetória reta sem rotação

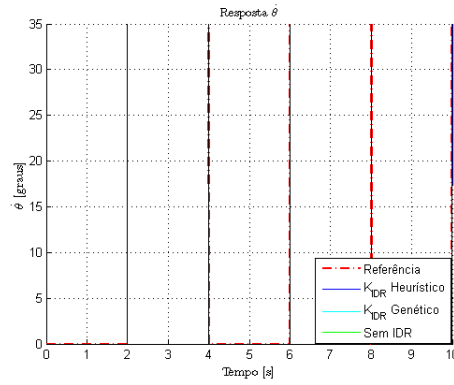


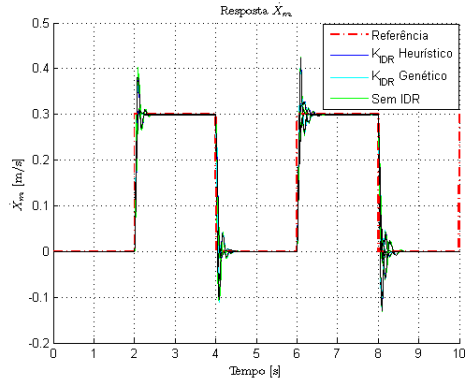
Figura D.1: Resposta  $\dot{\theta}$  para trajetória reta sem rotação

Tabela D.2: Erro por Vértice: Trajetória Reta Sem Rotação

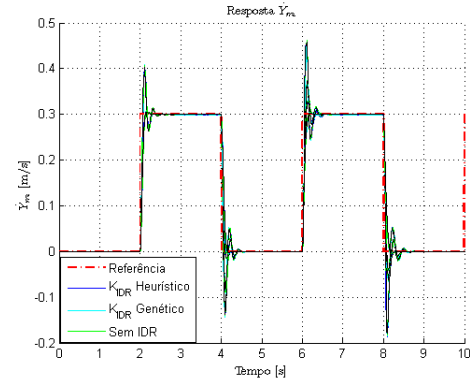
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	2.563638	<b>2.301941</b>	2.403746
1	$e_{vx}$	0.031227	<b>0.019570</b>	0.021199
1	$e_{vy}$	0.038315	<b>0.024382</b>	0.033932
1	$e_{\dot{\theta}}$	2.563162	<b>2.301729</b>	2.403413
2	$ e_v $	2.493875	2.406760	<b>2.278024</b>
2	$e_{vx}$	0.030180	0.020507	<b>0.020375</b>
2	$e_{vy}$	0.037300	<b>0.025608</b>	0.032618
2	$e_{\dot{\theta}}$	2.493413	2.406536	<b>2.277699</b>
3	$ e_v $	2.618399	2.281104	<b>2.244752</b>
3	$e_{vx}$	0.033222	0.022077	<b>0.021606</b>
3	$e_{vy}$	0.036826	<b>0.019903</b>	0.031464
3	$e_{\dot{\theta}}$	2.617929	2.280910	<b>2.244428</b>
4	$ e_v $	2.478167	<b>2.156378</b>	2.327533
4	$e_{vx}$	0.032355	<b>0.021161</b>	0.022290
4	$e_{vy}$	0.035568	<b>0.018917</b>	0.031866
4	$e_{\dot{\theta}}$	2.477700	<b>2.156191</b>	2.327208
5	$ e_v $	2.569292	2.354738	<b>2.299723</b>
5	$e_{vx}$	0.030884	0.026515	<b>0.026188</b>
5	$e_{vy}$	0.031946	<b>0.026486</b>	0.028234
5	$e_{\dot{\theta}}$	2.568907	2.354440	<b>2.299400</b>
6	$ e_v $	2.574156	2.437198	<b>2.356759</b>
6	$e_{vx}$	0.030287	0.027377	<b>0.026359</b>
6	$e_{vy}$	0.031309	<b>0.027347</b>	0.028567
6	$e_{\dot{\theta}}$	2.573788	2.436891	<b>2.356439</b>
7	$ e_v $	2.622147	<b>2.374583</b>	2.390530
7	$e_{vx}$	0.029322	0.024298	<b>0.023974</b>
7	$e_{vy}$	0.034564	<b>0.030635</b>	0.031795
7	$e_{\dot{\theta}}$	2.621755	<b>2.374261</b>	2.390198
8	$ e_v $	<b>2.396545</b>	2.414314	2.463445
8	$e_{vx}$	0.027027	<b>0.024146</b>	0.024440
8	$e_{vy}$	0.032069	<b>0.030413</b>	0.032402
8	$e_{\dot{\theta}}$	<b>2.396178</b>	2.414002	2.463111

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	2.563638	<b>2.301941</b>	2.403746
9	$e_{vx}$	0.031227	<b>0.019570</b>	0.021199
9	$e_{vy}$	0.038315	<b>0.024382</b>	0.033932
9	$e_{\dot{\theta}}$	2.563162	<b>2.301729</b>	2.403413
10	$ e_v $	2.493875	2.406760	<b>2.278024</b>
10	$e_{vx}$	0.030180	0.020507	<b>0.020375</b>
10	$e_{vy}$	0.037300	<b>0.025608</b>	0.032618
10	$e_{\dot{\theta}}$	2.493413	2.406536	<b>2.277699</b>
11	$ e_v $	2.618399	2.281104	<b>2.244752</b>
11	$e_{vx}$	0.033222	0.022077	<b>0.021606</b>
11	$e_{vy}$	0.036826	<b>0.019903</b>	0.031464
11	$e_{\dot{\theta}}$	2.617929	2.280910	<b>2.244428</b>
12	$ e_v $	2.478167	<b>2.156378</b>	2.327533
12	$e_{vx}$	0.032355	<b>0.021161</b>	0.022290
12	$e_{vy}$	0.035568	<b>0.018917</b>	0.031866
12	$e_{\dot{\theta}}$	2.477700	<b>2.156191</b>	2.327208
13	$ e_v $	2.569292	2.354738	<b>2.299723</b>
13	$e_{vx}$	0.030884	0.026515	<b>0.026188</b>
13	$e_{vy}$	0.031946	<b>0.026486</b>	0.028234
13	$e_{\dot{\theta}}$	2.568907	2.354440	<b>2.299400</b>
14	$ e_v $	2.574156	2.437198	<b>2.356759</b>
14	$e_{vx}$	0.030287	0.027377	<b>0.026359</b>
14	$e_{vy}$	0.031309	<b>0.027347</b>	0.028567
14	$e_{\dot{\theta}}$	2.573788	2.436891	<b>2.356439</b>
15	$ e_v $	2.622147	<b>2.374583</b>	2.390530
15	$e_{vx}$	0.029322	0.024298	<b>0.023974</b>
15	$e_{vy}$	0.034564	<b>0.030635</b>	0.031795
15	$e_{\dot{\theta}}$	2.621755	<b>2.374261</b>	2.390198
16	$ e_v $	<b>2.396545</b>	2.414314	2.463445
16	$e_{vx}$	0.027027	<b>0.024146</b>	0.024440
16	$e_{vy}$	0.032069	<b>0.030413</b>	0.032402
16	$e_{\dot{\theta}}$	<b>2.396178</b>	2.414002	2.463111

### D.1.3 Trajetória Reta Com Rotação



(a) Resposta  $\dot{X}_m$  para trajetória reta com rotação



(b) Resposta  $\dot{Y}_m$  para trajetória reta com rotação

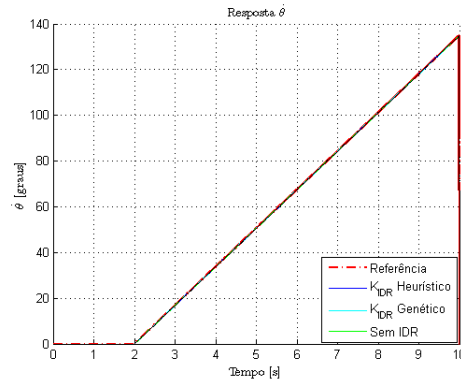


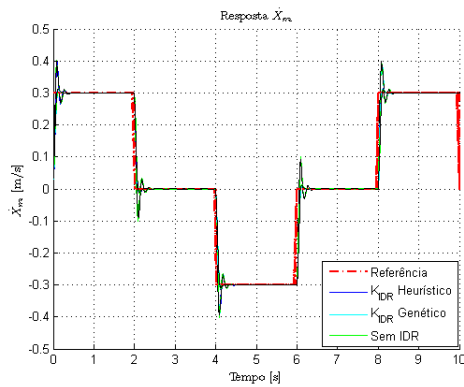
Figura D.2: Resposta  $\dot{\theta}$  para trajetória reta com rotação

Tabela D.3: Erro por Vértice: Trajetória Reta Com Rotação

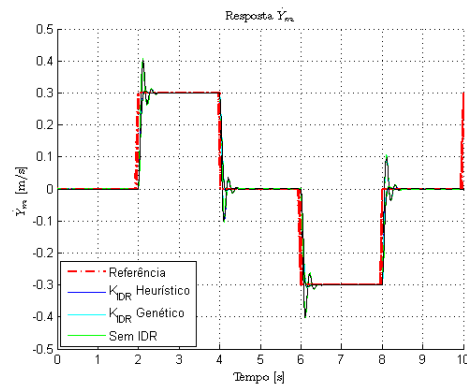
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	<b>1.073441</b>	1.092719	1.092340
1	$e_{vx}$	0.027714	0.026342	<b>0.025749</b>
1	$e_{vy}$	0.034994	0.034245	<b>0.033483</b>
1	$e_{\dot{\theta}}$	<b>1.072513</b>	1.091865	1.091523
2	$ e_v $	<b>1.072421</b>	1.075403	1.094460
2	$e_{vx}$	0.027358	0.027001	<b>0.026030</b>
2	$e_{vy}$	0.034589	0.034600	<b>0.033550</b>
2	$e_{\dot{\theta}}$	<b>1.071514</b>	1.074507	1.093636
3	$ e_v $	<b>1.078205</b>	1.095080	1.108633
3	$e_{vx}$	<b>0.027709</b>	0.027987	0.028377
3	$e_{vy}$	0.029633	<b>0.029618</b>	0.029860
3	$e_{\dot{\theta}}$	<b>1.077441</b>	1.094321	1.107867
4	$ e_v $	<b>1.095093</b>	1.103881	1.110522
4	$e_{vx}$	<b>0.028211</b>	0.029827	0.028420
4	$e_{vy}$	0.030310	0.031849	<b>0.029926</b>
4	$e_{\dot{\theta}}$	<b>1.094310</b>	1.103018	1.109755
5	$ e_v $	<b>1.083175</b>	1.122483	1.137489
5	$e_{vx}$	<b>0.027694</b>	0.027804	0.028921
5	$e_{vy}$	<b>0.028694</b>	0.029117	0.030299
5	$e_{\dot{\theta}}$	1.082440	<b>1.121761</b>	1.136718
6	$ e_v $	<b>1.092306</b>	1.135521	1.132073
6	$e_{vx}$	<b>0.027716</b>	0.028297	0.028076
6	$e_{vy}$	<b>0.028758</b>	0.029511	0.029256
6	$e_{\dot{\theta}}$	<b>1.091575</b>	1.134785	1.131346
7	$ e_v $	<b>1.091170</b>	1.126823	1.145862
7	$e_{vx}$	<b>0.024517</b>	0.024739	0.024768
7	$e_{vy}$	<b>0.028313</b>	0.028379	0.028549
7	$e_{\dot{\theta}}$	<b>1.090527</b>	1.126194	1.145238
8	$ e_v $	<b>1.103414</b>	1.144388	1.143412
8	$e_{vx}$	<b>0.024737</b>	0.025519	0.025166
8	$e_{vy}$	<b>0.028573</b>	0.029455	0.028914
8	$e_{\dot{\theta}}$	<b>1.102766</b>	1.143724	1.142769

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	<b>1.073441</b>	1.092719	1.092340
9	$e_{vx}$	0.027714	0.026342	<b>0.025749</b>
9	$e_{vy}$	0.034994	0.034245	<b>0.033483</b>
9	$e_{\dot{\theta}}$	<b>1.072513</b>	1.091865	1.091523
10	$ e_v $	<b>1.072421</b>	1.075403	1.094460
10	$e_{vx}$	0.027358	0.027001	<b>0.026030</b>
10	$e_{vy}$	0.034589	0.034600	<b>0.033550</b>
10	$e_{\dot{\theta}}$	<b>1.071514</b>	1.074507	1.093636
11	$ e_v $	<b>1.078205</b>	1.095080	1.108633
11	$e_{vx}$	<b>0.027709</b>	0.027987	0.028377
11	$e_{vy}$	0.029633	<b>0.029618</b>	0.029860
11	$e_{\dot{\theta}}$	<b>1.077441</b>	1.094321	1.107867
12	$ e_v $	<b>1.095093</b>	1.103881	1.110522
12	$e_{vx}$	<b>0.028211</b>	0.029827	0.028420
12	$e_{vy}$	0.030310	0.031849	<b>0.029926</b>
12	$e_{\dot{\theta}}$	<b>1.094310</b>	1.103018	1.109755
13	$ e_v $	<b>1.083175</b>	1.122483	1.137489
13	$e_{vx}$	<b>0.027694</b>	0.027804	0.028921
13	$e_{vy}$	<b>0.028694</b>	0.029117	0.030299
13	$e_{\dot{\theta}}$	1.082440	<b>1.121761</b>	1.136718
14	$ e_v $	<b>1.092306</b>	1.135521	1.132073
14	$e_{vx}$	<b>0.027716</b>	0.028297	0.028076
14	$e_{vy}$	<b>0.028758</b>	0.029511	0.029256
14	$e_{\dot{\theta}}$	<b>1.091575</b>	1.134785	1.131346
15	$ e_v $	<b>1.091170</b>	1.126823	1.145862
15	$e_{vx}$	<b>0.024517</b>	0.024739	0.024768
15	$e_{vy}$	<b>0.028313</b>	0.028379	0.028549
15	$e_{\dot{\theta}}$	<b>1.090527</b>	1.126194	1.145238
16	$ e_v $	<b>1.103414</b>	1.144388	1.143412
16	$e_{vx}$	<b>0.024737</b>	0.025519	0.025166
16	$e_{vy}$	<b>0.028573</b>	0.029455	0.028914
16	$e_{\dot{\theta}}$	<b>1.102766</b>	1.143724	1.142769

### D.1.4 Trajetória Quadrada Sem Rotação



(a) Resposta  $\dot{X}_m$  para trajetória quadrada sem rotação



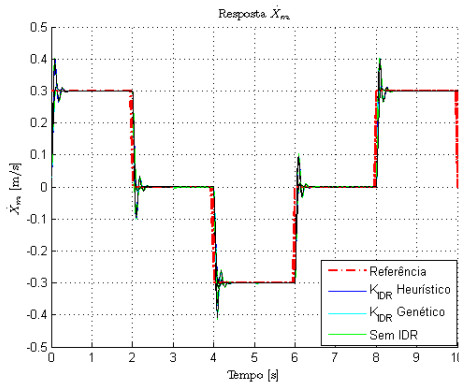
(b) Resposta  $\dot{Y}_m$  para trajetória quadrada sem rotação

Tabela D.4: Erro por Vértice: Trajetória Quadrada Sem Rotação

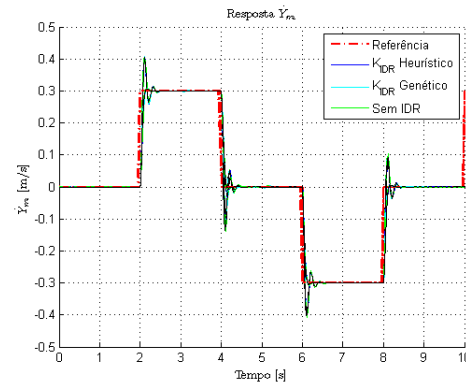
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.038680	0.034212	<b>0.034176</b>
1	$e_{vx}$	0.028493	0.025232	<b>0.025189</b>
1	$e_{vy}$	0.026159	0.023102	<b>0.023096</b>
1	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000238	0.000292
2	$ e_v $	0.038680	0.035410	<b>0.034760</b>
2	$e_{vx}$	0.028493	0.026989	<b>0.025704</b>
2	$e_{vy}$	0.026159	<b>0.022922</b>	0.023398
2	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000234	0.000290
3	$ e_v $	0.038680	<b>0.034290</b>	0.034750
3	$e_{vx}$	0.028493	<b>0.025348</b>	0.025623
3	$e_{vy}$	0.026159	<b>0.023093</b>	0.023472
3	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000238	0.000295
4	$ e_v $	0.038680	0.035658	<b>0.034780</b>
4	$e_{vx}$	0.028493	0.027088	<b>0.025675</b>
4	$e_{vy}$	0.026159	<b>0.023188</b>	0.023460
4	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000241	0.000294
5	$ e_v $	0.036233	0.030103	<b>0.029617</b>
5	$e_{vx}$	0.026432	0.023218	<b>0.022848</b>
5	$e_{vy}$	0.024782	0.019158	<b>0.018844</b>
5	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000173	0.000213
6	$ e_v $	0.036233	<b>0.029290</b>	0.030435
6	$e_{vx}$	0.026432	<b>0.022754</b>	0.023407
6	$e_{vy}$	0.024782	<b>0.018442</b>	0.019451
6	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000169	0.000209
7	$ e_v $	0.036233	<b>0.029685</b>	0.029739
7	$e_{vx}$	0.026432	0.022915	<b>0.022892</b>
7	$e_{vy}$	0.024782	<b>0.018870</b>	0.018981
7	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000168	0.000204
8	$ e_v $	0.036233	<b>0.029740</b>	0.030410
8	$e_{vx}$	0.026432	<b>0.023064</b>	0.023418
8	$e_{vy}$	0.024782	<b>0.018775</b>	0.019398
8	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000172	0.000213

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.038680	0.034212	<b>0.034176</b>
9	$e_{vx}$	0.028493	0.025232	<b>0.025189</b>
9	$e_{vy}$	0.026159	0.023102	<b>0.023096</b>
9	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000238	0.000292
10	$ e_v $	0.038680	0.035410	<b>0.034760</b>
10	$e_{vx}$	0.028493	0.026989	<b>0.025704</b>
10	$e_{vy}$	0.026159	<b>0.022922</b>	0.023398
10	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000234	0.000290
11	$ e_v $	0.038680	<b>0.034290</b>	0.034750
11	$e_{vx}$	0.028493	<b>0.025348</b>	0.025623
11	$e_{vy}$	0.026159	<b>0.023093</b>	0.023472
11	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000238	0.000295
12	$ e_v $	0.038680	0.035658	<b>0.034780</b>
12	$e_{vx}$	0.028493	0.027088	<b>0.025675</b>
12	$e_{vy}$	0.026159	<b>0.023188</b>	0.023460
12	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000241	0.000294
13	$ e_v $	0.036233	0.030103	<b>0.029617</b>
13	$e_{vx}$	0.026432	0.023218	<b>0.022848</b>
13	$e_{vy}$	0.024782	0.019158	<b>0.018844</b>
13	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000173	0.000213
14	$ e_v $	0.036233	<b>0.029290</b>	0.030435
14	$e_{vx}$	0.026432	<b>0.022754</b>	0.023407
14	$e_{vy}$	0.024782	<b>0.018442</b>	0.019451
14	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000169	0.000209
15	$ e_v $	0.036233	<b>0.029685</b>	0.029739
15	$e_{vx}$	0.026432	0.022915	<b>0.022892</b>
15	$e_{vy}$	0.024782	<b>0.018870</b>	0.018981
15	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000168	0.000204
16	$ e_v $	0.036233	<b>0.029740</b>	0.030410
16	$e_{vx}$	0.026432	<b>0.023064</b>	0.023418
16	$e_{vy}$	0.024782	<b>0.018775</b>	0.019398
16	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000172	0.000213

### D.1.5 Trajetória Quadrada Com Rotação



(c) Resposta  $\dot{X}_m$  para trajetória quadrada com rotação



(d) Resposta  $\dot{Y}_m$  para trajetória quadrada com rotação



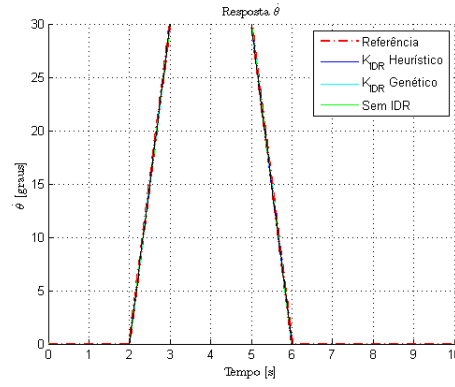


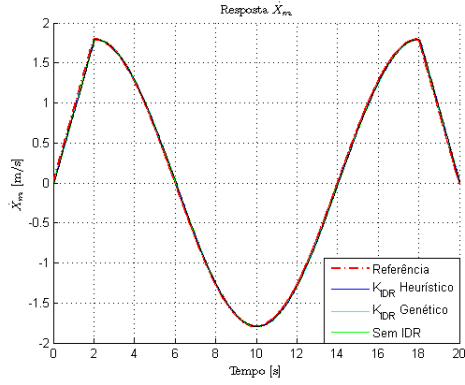
Figura D.3: Resposta  $\dot{\theta}$  para trajetória quadrada com rotação

Tabela D.5: Erro por Vértice: Trajetória Quadrada Com Rotação

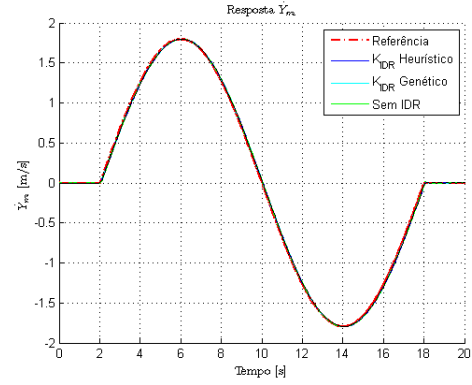
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.174581	<b>0.170217</b>	0.172814
1	$e_{vx}$	0.030335	<b>0.026426</b>	<b>0.026426</b>
1	$e_{vy}$	0.029090	0.025906	<b>0.025736</b>
1	$e_{\dot{\theta}}$	0.169446	<b>0.166145</b>	0.168832
2	$ e_v $	0.179898	0.170704	<b>0.167500</b>
2	$e_{vx}$	0.030781	0.028435	<b>0.026716</b>
2	$e_{vy}$	0.029555	<b>0.026383</b>	0.026390
2	$e_{\dot{\theta}}$	0.174764	0.166239	<b>0.163236</b>
3	$ e_v $	0.176874	<b>0.167893</b>	0.170427
3	$e_{vx}$	0.031969	0.028073	<b>0.027736</b>
3	$e_{vy}$	0.029494	0.026166	<b>0.025757</b>
3	$e_{\dot{\theta}}$	0.171443	<b>0.163448</b>	0.166170
4	$ e_v $	0.180214	0.174607	<b>0.169034</b>
4	$e_{vx}$	0.031516	0.030042	<b>0.028829</b>
4	$e_{vy}$	0.028883	<b>0.026511</b>	0.027051
4	$e_{\dot{\theta}}$	0.175071	0.169948	<b>0.164346</b>
5	$ e_v $	0.178274	0.167480	<b>0.165743</b>
5	$e_{vx}$	0.026985	0.026470	<b>0.026080</b>
5	$e_{vy}$	0.026076	0.022944	<b>0.022536</b>
5	$e_{\dot{\theta}}$	0.174280	0.163776	<b>0.162119</b>
6	$ e_v $	0.183384	0.170565	<b>0.168638</b>
6	$e_{vx}$	0.028001	0.026842	<b>0.026359</b>
6	$e_{vy}$	0.027098	0.023060	<b>0.022754</b>
6	$e_{\dot{\theta}}$	0.179196	0.166853	<b>0.165004</b>
7	$ e_v $	0.183708	0.167733	<b>0.165537</b>
7	$e_{vx}$	0.027786	<b>0.026664</b>	0.027390
7	$e_{vy}$	0.027166	<b>0.023355</b>	0.024134
7	$e_{\dot{\theta}}$	0.179551	0.163944	<b>0.161462</b>
8	$ e_v $	0.183961	0.171558	<b>0.167647</b>
8	$e_{vx}$	0.027353	<b>0.026759</b>	0.027001
8	$e_{vy}$	0.026702	<b>0.023183</b>	0.023691
8	$e_{\dot{\theta}}$	0.179946	0.167865	<b>0.163754</b>

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.174581	<b>0.170217</b>	0.172814
9	$e_{vx}$	0.030335	<b>0.026426</b>	<b>0.026426</b>
9	$e_{vy}$	0.029090	0.025906	<b>0.025736</b>
9	$e_{\dot{\theta}}$	0.169446	<b>0.166145</b>	0.168832
10	$ e_v $	0.179898	0.170704	<b>0.167500</b>
10	$e_{vx}$	0.030781	0.028435	<b>0.026716</b>
10	$e_{vy}$	0.029555	<b>0.026383</b>	0.026390
10	$e_{\dot{\theta}}$	0.174764	<b>0.166239</b>	0.163236
11	$ e_v $	0.176874	<b>0.167893</b>	0.170427
11	$e_{vx}$	0.031969	0.028073	<b>0.027736</b>
11	$e_{vy}$	0.029494	0.026166	<b>0.025757</b>
11	$e_{\dot{\theta}}$	0.171443	<b>0.163448</b>	0.166170
12	$ e_v $	0.180214	0.174607	<b>0.169034</b>
12	$e_{vx}$	0.031516	0.030042	<b>0.028829</b>
12	$e_{vy}$	0.028883	<b>0.026511</b>	0.027051
12	$e_{\dot{\theta}}$	0.175071	0.169948	<b>0.164346</b>
13	$ e_v $	0.178274	0.167480	<b>0.165743</b>
13	$e_{vx}$	0.026985	0.026470	<b>0.026080</b>
13	$e_{vy}$	0.026076	0.022944	<b>0.022536</b>
13	$e_{\dot{\theta}}$	0.174280	0.163776	<b>0.162119</b>
14	$ e_v $	0.183384	0.170565	<b>0.168638</b>
14	$e_{vx}$	0.028001	0.026842	<b>0.026359</b>
14	$e_{vy}$	0.027098	0.023060	<b>0.022754</b>
14	$e_{\dot{\theta}}$	0.179196	0.166853	<b>0.165004</b>
15	$ e_v $	0.183708	0.167733	<b>0.165537</b>
15	$e_{vx}$	0.027786	<b>0.026664</b>	0.027390
15	$e_{vy}$	0.027166	<b>0.023355</b>	0.024134
15	$e_{\dot{\theta}}$	0.179551	0.163944	<b>0.161462</b>
16	$ e_v $	0.183961	0.171558	<b>0.167647</b>
16	$e_{vx}$	0.027353	<b>0.026759</b>	0.027001
16	$e_{vy}$	0.026702	<b>0.023183</b>	0.023691
16	$e_{\dot{\theta}}$	0.179946	0.167865	<b>0.163754</b>

### D.1.6 Trajetória Circular Sem Rotação



(a) Resposta  $\dot{X}_m$  para trajetória circular sem rotação



(b) Resposta  $\dot{Y}_m$  para trajetória circular sem rotação

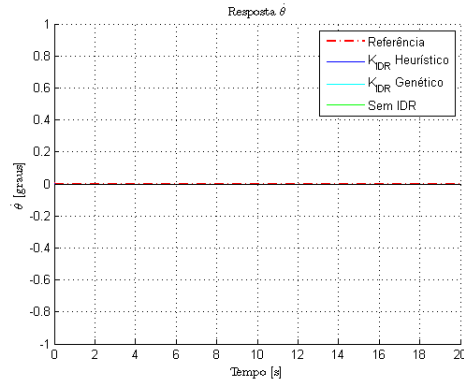


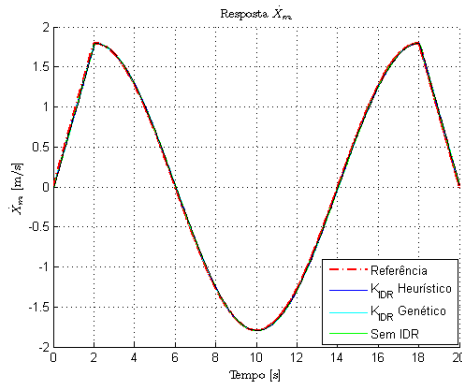
Figura D.4: Resposta  $\dot{\theta}$  para trajetória circular sem rotação

Tabela D.6: Erro por Vértice: Trajetória Circular Sem Rotação

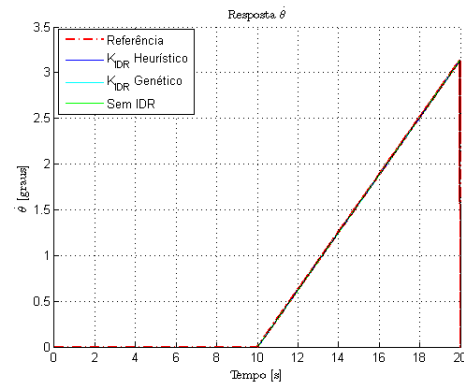
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.012210	<b>0.012136</b>	0.012140
1	$e_{vx}$	0.010072	<b>0.009748</b>	0.009751
1	$e_{vy}$	<b>0.006902</b>	0.007229	0.007231
1	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000072	0.000092
2	$ e_v $	0.012210	0.012121	<b>0.012119</b>
2	$e_{vx}$	0.010072	<b>0.009745</b>	<b>0.009745</b>
2	$e_{vy}$	0.006902	0.007207	<b>0.007204</b>
2	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000078	0.000086
3	$ e_v $	0.012210	<b>0.012130</b>	0.012135
3	$e_{vx}$	0.010072	<b>0.009746</b>	0.009747
3	$e_{vy}$	<b>0.006902</b>	0.007222	0.007228
3	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000074	0.000096
4	$ e_v $	0.012210	0.012146	<b>0.012126</b>
4	$e_{vx}$	0.010072	0.009739	<b>0.009718</b>
4	$e_{vy}$	0.006902	0.007257	<b>0.007252</b>
4	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000076	0.000086
5	$ e_v $	<b>0.026740</b>	0.026876	0.026882
5	$e_{vx}$	0.020745	0.020427	<b>0.020368</b>
5	$e_{vy}$	<b>0.016872</b>	0.017463	0.017541
5	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000291	0.000339
6	$ e_v $	<b>0.026740</b>	0.026849	0.026905
6	$e_{vx}$	0.020745	<b>0.020396</b>	0.020406
6	$e_{vy}$	<b>0.016872</b>	0.017459	0.017532
6	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000289	0.000343
7	$ e_v $	<b>0.026740</b>	0.026878	0.026901
7	$e_{vx}$	0.020745	0.020416	<b>0.020400</b>
7	$e_{vy}$	<b>0.016872</b>	0.017478	0.017532
7	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000295	0.000340
8	$ e_v $	<b>0.026740</b>	0.026875	0.026871
8	$e_{vx}$	0.020745	<b>0.020398</b>	0.020445
8	$e_{vy}$	<b>0.016872</b>	0.017496	0.017434
8	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000292	0.000345

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.012210	<b>0.012136</b>	0.012140
9	$e_{vx}$	0.010072	<b>0.009748</b>	0.009751
9	$e_{vy}$	0.006902	<b>0.007229</b>	0.007231
9	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000072	0.000092
10	$ e_v $	0.012210	0.012121	<b>0.012119</b>
10	$e_{vx}$	0.010072	<b>0.009745</b>	<b>0.009745</b>
10	$e_{vy}$	<b>0.006902</b>	0.007207	0.007204
10	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000078	0.000086
11	$ e_v $	0.012210	<b>0.012130</b>	0.012135
11	$e_{vx}$	0.010072	<b>0.009746</b>	0.009747
11	$e_{vy}$	<b>0.006902</b>	0.007222	0.007228
11	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000074	0.000096
12	$ e_v $	0.012210	0.012146	<b>0.012126</b>
12	$e_{vx}$	0.010072	0.009739	<b>0.009718</b>
12	$e_{vy}$	<b>0.006902</b>	0.007257	0.007252
12	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000076	0.000086
13	$ e_v $	<b>0.026740</b>	0.026876	0.026882
13	$e_{vx}$	0.020745	0.020427	<b>0.020368</b>
13	$e_{vy}$	<b>0.016872</b>	0.017463	0.017541
13	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000291	0.000339
14	$ e_v $	<b>0.026740</b>	0.026849	0.026905
14	$e_{vx}$	0.020745	<b>0.020396</b>	0.020406
14	$e_{vy}$	<b>0.016872</b>	0.017459	0.017532
14	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000289	0.000343
15	$ e_v $	<b>0.026740</b>	0.026878	0.026901
15	$e_{vx}$	0.020745	0.020416	<b>0.020400</b>
15	$e_{vy}$	<b>0.016872</b>	0.017478	0.017532
15	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000295	0.000340
16	$ e_v $	<b>0.026740</b>	0.026875	0.026871
16	$e_{vx}$	0.020745	<b>0.020398</b>	0.020445
16	$e_{vy}$	<b>0.016872</b>	0.017496	0.017434
16	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000292	0.000345

### D.1.7 Trajetória Circular Com Rotação



(a) Resposta  $\dot{X}_m$  para trajetória circular com rotação



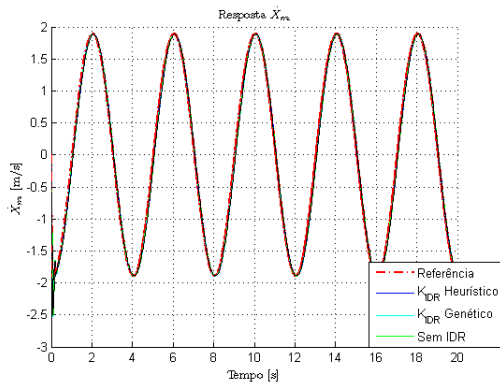
(b) Resposta  $\dot{\theta}$  para trajetória circular com rotação

Tabela D.7: Erro por Vértice: Trajetória Circular Com Rotação

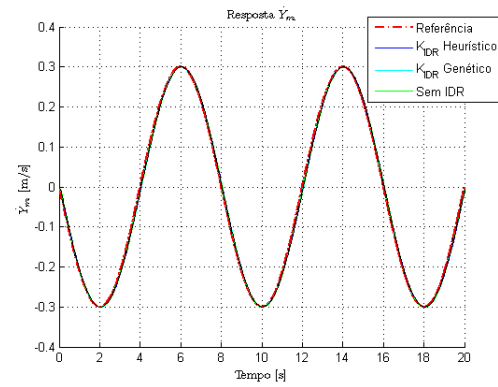
Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.018407	<b>0.017435</b>	0.017528
1	$e_{vx}$	0.009509	<b>0.009694</b>	0.009697
1	$e_{vy}$	0.007343	0.007305	<b>0.007270</b>
1	$e_{\dot{\theta}}$	0.013946	<b>0.012516</b>	0.012663
2	$ e_v $	0.018524	<b>0.017402</b>	0.017514
2	$e_{vx}$	<b>0.009432</b>	0.009712	0.009682
2	$e_{vy}$	0.007427	<b>0.007252</b>	0.007263
2	$e_{\dot{\theta}}$	0.014107	<b>0.012487</b>	0.012659
3	$ e_v $	0.018423	<b>0.017399</b>	0.017539
3	$e_{vx}$	<b>0.009511</b>	0.009802	0.009756
3	$e_{vy}$	0.007385	<b>0.007192</b>	0.007246
3	$e_{\dot{\theta}}$	0.013944	<b>0.012447</b>	0.012646
4	$ e_v $	0.018537	<b>0.017414</b>	0.017509
4	$e_{vx}$	<b>0.009432</b>	0.009708	0.009683
4	$e_{vy}$	0.007471	<b>0.007286</b>	0.007294
4	$e_{\dot{\theta}}$	0.014101	<b>0.012486</b>	0.012633
5	$ e_v $	0.030351	<b>0.029644</b>	0.029753
5	$e_{vx}$	<b>0.020103</b>	0.020482	0.020399
5	$e_{vy}$	0.017660	<b>0.017318</b>	0.017483
5	$e_{\dot{\theta}}$	0.014325	<b>0.012624</b>	0.012786
6	$ e_v $	0.030367	<b>0.029706</b>	0.029748
6	$e_{vx}$	<b>0.020146</b>	0.020378	0.020346
6	$e_{vy}$	0.017592	<b>0.017485</b>	0.017510
6	$e_{\dot{\theta}}$	0.014381	<b>0.012706</b>	0.012822
7	$ e_v $	0.030367	<b>0.029723</b>	0.029801
7	$e_{vx}$	<b>0.020205</b>	0.020440	0.020532
7	$e_{vy}$	0.017604	0.017453	<b>0.017350</b>
7	$e_{\dot{\theta}}$	0.014284	<b>0.012691</b>	0.012866
8	$ e_v $	0.030399	<b>0.029699</b>	0.029793
8	$e_{vx}$	<b>0.020152</b>	0.020351	0.020269
8	$e_{vy}$	0.017638	<b>0.017540</b>	0.017661
8	$e_{\dot{\theta}}$	0.014384	<b>0.012658</b>	0.012839

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.018407	<b>0.017435</b>	0.017528
9	$e_{vx}$	0.009509	<b>0.009694</b>	0.009697
9	$e_{vy}$	0.007343 0.007305	<b>0.007270</b>	
9	$e_{\dot{\theta}}$	0.013946	<b>0.012516</b>	0.012663
10	$ e_v $	0.018524	<b>0.017402</b>	0.017514
10	$e_{vx}$	<b>0.009432</b>	0.009712	0.009682
10	$e_{vy}$	0.007427	<b>0.007252</b>	0.007263
10	$e_{\dot{\theta}}$	0.014107	<b>0.012487</b>	0.012659
11	$ e_v $	0.018423	<b>0.017399</b>	0.017539
11	$e_{vx}$	<b>0.009511</b>	0.009802	0.009756
11	$e_{vy}$	0.007385	<b>0.007192</b>	0.007246
11	$e_{\dot{\theta}}$	0.013944	<b>0.012447</b>	0.012646
12	$ e_v $	0.018537	<b>0.017414</b>	0.017509
12	$e_{vx}$	<b>0.009432</b>	0.009708	0.009683
12	$e_{vy}$	0.007471	<b>0.007286</b>	0.007294
12	$e_{\dot{\theta}}$	0.014101	<b>0.012486</b>	0.012633
13	$ e_v $	0.030351	<b>0.029644</b>	0.029753
13	$e_{vx}$	<b>0.020103</b>	0.020482	0.020399
13	$e_{vy}$	0.017660	<b>0.017318</b>	0.017483
13	$e_{\dot{\theta}}$	0.014325	<b>0.012624</b>	0.012786
14	$ e_v $	0.030367	<b>0.029706</b>	0.029748
14	$e_{vx}$	<b>0.020146</b>	0.020378	0.020346
14	$e_{vy}$	0.017592	<b>0.017485</b>	0.017510
14	$e_{\dot{\theta}}$	0.014381	<b>0.012706</b>	0.012822
15	$ e_v $	0.030367	<b>0.029723</b>	0.029801
15	$e_{vx}$	<b>0.020205</b>	0.020440	0.020532
15	$e_{vy}$	0.017604	0.017453	<b>0.017350</b>
15	$e_{\dot{\theta}}$	0.014284	<b>0.012691</b>	0.012866
16	$ e_v $	0.030399	<b>0.029699</b>	0.029793
16	$e_{vx}$	<b>0.020152</b>	0.020351	0.020269
16	$e_{vy}$	0.017638	<b>0.017540</b>	0.017661
16	$e_{\dot{\theta}}$	0.014384	<b>0.012658</b>	0.012839

### D.1.8 Trajetória em Oito Sem Rotação



(c) Resposta  $\dot{X}_m$  para trajetória em oito sem rotação



(d) Resposta  $\dot{Y}_m$  para trajetória em oito sem rotação

Tabela D.8: Erro por Vértice: Trajetória em Oito Sem Rotação

Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.042096	0.041791	<b>0.041602</b>
1	$e_{vx}$	0.041982	0.041678	<b>0.041487</b>
1	$e_{vy}$	0.003103	<b>0.003076</b>	0.003079
1	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000129	0.000152
2	$ e_v $	0.042096	0.041811	<b>0.041583</b>
2	$e_{vx}$	0.041982	0.041698	<b>0.041469</b>
2	$e_{vy}$	0.003103	<b>0.003077</b>	0.003078
2	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000151
3	$ e_v $	<b>0.042096</b>	0.042674	0.042402
3	$e_{vx}$	<b>0.041982</b>	0.042563	0.042289
3	$e_{vy}$	0.003103	<b>0.003075</b>	0.003080
3	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000153
4	$ e_v $	0.042096	0.042429	<b>0.041504</b>
4	$e_{vx}$	0.041982	0.042318	<b>0.041389</b>
4	$e_{vy}$	0.003103	<b>0.003076</b>	0.003078
4	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000151
5	$ e_v $	0.084829	0.084142	<b>0.084139</b>
5	$e_{vx}$	0.084489	0.083803	<b>0.083801</b>
5	$e_{vy}$	0.007582	0.007540	<b>0.007519</b>
5	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405
6	$ e_v $	0.084829	<b>0.084042</b>	0.084142
6	$e_{vx}$	0.084489	<b>0.083703</b>	0.083804
6	$e_{vy}$	0.007582	0.007534	<b>0.007518</b>
6	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405
7	$ e_v $	0.084829	<b>0.084335</b>	0.084466
7	$e_{vx}$	0.084489	<b>0.083997</b>	0.084130
7	$e_{vy}$	0.007582	0.007536	<b>0.007519</b>
7	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000406
8	$ e_v $	0.084829	<b>0.084208</b>	0.084451
8	$e_{vx}$	0.084489	<b>0.083870</b>	0.084115
8	$e_{vy}$	0.007582	0.007532	<b>0.007514</b>
8	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.042096	0.041791	<b>0.041602</b>
9	$e_{vx}$	0.041982	0.041678	<b>0.041487</b>
9	$e_{vy}$	0.003103	<b>0.003076</b>	0.003079
9	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000129	0.000152
10	$ e_v $	0.042096	0.041811	<b>0.041583</b>
10	$e_{vx}$	0.041982	0.041698	<b>0.041469</b>
10	$e_{vy}$	0.003103	<b>0.003077</b>	0.003078
10	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000151
11	$ e_v $	<b>0.042096</b>	0.042674	0.042402
11	$e_{vx}$	<b>0.041982</b>	0.042563	0.042289
11	$e_{vy}$	0.003103	<b>0.003075</b>	0.003080
11	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000153
12	$ e_v $	0.042096	0.042429	<b>0.041504</b>
12	$e_{vx}$	0.041982	0.042318	<b>0.041389</b>
12	$e_{vy}$	0.003103	<b>0.003076</b>	0.003078
12	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000128	0.000151
13	$ e_v $	0.084829	0.084142	<b>0.084139</b>
13	$e_{vx}$	0.084489	0.083803	<b>0.083801</b>
13	$e_{vy}$	0.007582	0.007540	<b>0.007519</b>
13	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405
14	$ e_v $	0.084829	<b>0.084042</b>	0.084142
14	$e_{vx}$	0.084489	<b>0.083703</b>	0.083804
14	$e_{vy}$	0.007582	0.007534	<b>0.007518</b>
14	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405
15	$ e_v $	0.084829	<b>0.084335</b>	0.084466
15	$e_{vx}$	0.084489	<b>0.083997</b>	0.084130
15	$e_{vy}$	0.007582	0.007536	<b>0.007519</b>
15	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000406
16	$ e_v $	0.084829	<b>0.084208</b>	0.084451
16	$e_{vx}$	0.084489	<b>0.083870</b>	0.084115
16	$e_{vy}$	0.007582	0.007532	<b>0.007514</b>
16	$e_{\dot{\theta}}$	<b>0.000000</b>	0.000332	0.000405

### D.1.9 Trajetória em Oito com Rotação

Tabela D.9: Erro por Vértice: Trajetória em Oito com Rotação

Vértice	Erro	Sem IDR	Heurístico	Genético
1	$ e_v $	0.192354	0.165873	<b>0.164986</b>
1	$e_{vx}$	<b>0.038556</b>	0.040654	0.040444
1	$e_{vy}$	0.007371	0.006749	<b>0.006708</b>
1	$e_{\dot{\theta}}$	0.188306	0.160672	<b>0.159811</b>
2	$ e_v $	0.194574	0.167510	<b>0.166395</b>
2	$e_{vx}$	<b>0.038695</b>	0.040654	0.040519
2	$e_{vy}$	0.007384	0.006791	<b>0.006700</b>
2	$e_{\dot{\theta}}$	0.190544	0.162360	<b>0.161247</b>
3	$ e_v $	0.191653	0.164256	<b>0.162047</b>
3	$e_{vx}$	<b>0.038946</b>	0.041883	0.041530
3	$e_{vy}$	0.003687	0.003613	<b>0.003601</b>
3	$e_{\dot{\theta}}$	0.187617	0.158785	<b>0.156593</b>
4	$ e_v $	0.193668	0.164644	<b>0.161166</b>
4	$e_{vx}$	<b>0.038705</b>	0.041669	0.040626
4	$e_{vy}$	0.003656	0.003661	<b>0.003581</b>
4	$e_{\dot{\theta}}$	0.189726	0.159242	<b>0.155921</b>
5	$ e_v $	0.205334	0.196573	<b>0.195782</b>
5	$e_{vx}$	<b>0.080682</b>	0.082429	0.082446
5	$e_{vy}$	0.010183	0.009971	<b>0.009954</b>
5	$e_{\dot{\theta}}$	0.188544	0.178177	<b>0.177296</b>
6	$ e_v $	0.207565	0.198366	<b>0.196902</b>
6	$e_{vx}$	<b>0.080396</b>	0.082555	0.082351
6	$e_{vy}$	0.010160	0.010047	<b>0.009939</b>
6	$e_{\dot{\theta}}$	0.191092	0.180091	<b>0.178577</b>
7	$ e_v $	0.209409	0.199193	<b>0.197891</b>
7	$e_{vx}$	<b>0.081080</b>	0.083129	0.083104
7	$e_{vy}$	<b>0.007229</b>	0.007344	0.007276
7	$e_{\dot{\theta}}$	0.192940	0.180869	<b>0.179448</b>
8	$ e_v $	0.211460	0.201377	<b>0.199707</b>
8	$e_{vx}$	<b>0.081067</b>	0.083153	0.083131
8	$e_{vy}$	<b>0.007225</b>	0.007444	0.007288
8	$e_{\dot{\theta}}$	0.195170	0.183256	<b>0.181436</b>

Vértice	Erro	Sem IDR	Heurístico	Genético
9	$ e_v $	0.192354	0.165873	<b>0.164986</b>
9	$e_{vx}$	<b>0.038556</b>	0.040654	0.040444
9	$e_{vy}$	0.007371	0.006749	<b>0.006708</b>
9	$e_{\dot{\theta}}$	0.188306	0.160672	<b>0.159811</b>
10	$ e_v $	0.194574	0.167510	<b>0.166395</b>
10	$e_{vx}$	<b>0.038695</b>	0.040654	0.040519
10	$e_{vy}$	0.007384	0.006791	<b>0.006700</b>
10	$e_{\dot{\theta}}$	0.190544	0.162360	<b>0.161247</b>
11	$ e_v $	0.191653	0.164256	<b>0.162047</b>
11	$e_{vx}$	<b>0.038946</b>	0.041883	0.041530
11	$e_{vy}$	0.003687	0.003613	<b>0.003601</b>
11	$e_{\dot{\theta}}$	0.187617	0.158785	<b>0.156593</b>
12	$ e_v $	0.193668	0.164644	<b>0.161166</b>
12	$e_{vx}$	<b>0.038705</b>	0.041669	0.040626
12	$e_{vy}$	0.003656	0.003661	<b>0.003581</b>
12	$e_{\dot{\theta}}$	0.189726	0.159242	<b>0.155921</b>
13	$ e_v $	0.205334	0.196573	<b>0.195782</b>
13	$e_{vx}$	<b>0.080682</b>	0.082429	0.082446
13	$e_{vy}$	0.010183	0.009971	<b>0.009954</b>
13	$e_{\dot{\theta}}$	0.188544	0.178177	<b>0.177296</b>
14	$ e_v $	0.207565	0.198366	<b>0.196902</b>
14	$e_{vx}$	<b>0.080396</b>	0.082555	0.082351
14	$e_{vy}$	0.010160	<b>0.010047</b>	0.009939
14	$e_{\dot{\theta}}$	0.191092	0.180091	<b>0.178577</b>
15	$ e_v $	0.209409	0.199193	<b>0.197891</b>
15	$e_{vx}$	<b>0.081080</b>	0.083129	0.083104
15	$e_{vy}$	<b>0.007229</b>	0.007344	0.007276
15	$e_{\dot{\theta}}$	0.192940	0.180869	<b>0.179448</b>
16	$ e_v $	0.211460	0.201377	<b>0.199707</b>
16	$e_{vx}$	<b>0.081067</b>	0.083153	0.083131
16	$e_{vy}$	<b>0.007225</b>	0.007444	0.007288
16	$e_{\dot{\theta}}$	0.195170	0.183256	<b>0.181436</b>

# Bibliografia

- [Ara07] F. M. U. Araújo. *Sistemas de Controle*. Universidade Federal do Rio Grande do Norte: DCA, 3 edição, 2007.
- [Ass73] Sabi J. R. Asseo. Decoupling of a class of nonlinear systems and its application to an aircraft control problem. *Journal of Aircraft*, 10(12):739–747, 1973.
- [Ast02] K. J. Astrom. *Control System Design*. Department of Mechanical and Environmental Engineering University of California - Santa Barbara, 01 edição, 2002.
- [Car12] V. Carrara. *Análise e Controle de Sistemas Lineares*. Instituto Nacional de Pesquisas Espaciais - INPE, 1 edição, 2012.
- [CG99] M. C. and P. Gahinet. Robust Pole Placement in LMI regions. *IEEE*, 44:2257 – 2270, 1999.
- [Che99] C.-T. Chen. *Linear System Theory and Design*. Oxford University Press, New York, 3 edição, 1999.
- [CLPV03] B. M. Chen, T. H. Lee, K. Peng, and V. Venkataramanan. Composite nonlinear feedback control for linear systems with input saturation: Theory and an application. *IEEE Transactions On Automatic Control*, 48(3), 2003.
- [Fra06] A. L. D. Franco. *Controle Não Linear Robusto: Um Método Baseado em uma Linearização por Realimentação*. Tese de Doutorado, Universidade Federal de Santa Catarina, 2006.
- [GB00] H. Guillard and H. Bourlès. Robust feedback linearization. *In Proceedings of the 14th International Symposium on Mathematical Theory of Networks and Systems (MTNS)*, 14, Junho 2000.
- [Gon06] E. N. Gonçalves. *Análise e Síntese de Controladores e Filtros Robustos para Sistemas com Domínios Politépicos de Incerteza*. Tese de Doutorado, Belo Horizonte, UFMG, 2006.
- [HB92] W. Haddad and D. Berustein. *Controller Design with regional pole constraints*. *IEEE Transactions on Automatic Control*, 37 edição, 1992.

- [HSM83] L. Hunt, R. Su, and G. Meyer. *Global transformations of nonlinear systems*. IEEE Transactions on Automatic Control, 1 edição, 1983.
- [HSM92] J. Hauser, S. Sastry, and G. Meyer. Nonlinear control design for slightly non minimum phase systems: Application to v/stol aircraft. *Automatica*, 28(4):665–679, 1992.
- [Isi85] A. Isidori. *Nonlinear Control Systems: An Introduction*. Springer-Verlag, New York, 1 edição, 1985.
- [Lav05] F. Lavergne. *Méthodologie de synthèse de lois de commandes non-linéaires et robustes : Application au suivi de trajectoire des avions de transport*. Tese de Doutorado, Toulouse, France: LAAS-CNRS, 2005.
- [Leo14] A. P. Leon. Algoritmos genéticos. [www.icmc.usp.br/andre/research/genetic/](http://www.icmc.usp.br/andre/research/genetic/), 2014. Acessado em Março de 2014.
- [Luc02] D. C. Lucas. Algoritmos genéticos: uma introdução. Technical report, 2002.
- [Mar14] L. A. B. S. Martin. Grupos de lie. <http://www.ime.unicamp.br/~smartin/cursos/grupolie-2013/gruplie0.pdf>, 2014. Instituto de Matemática, Estatística e Computação Científica, Universidade Estadual de Campinas - UNICAMP, acessado em Abril de 2014.
- [Mat14] Mathworks. R2014a documentation. <http://www.mathworks.com>, July 2014. Site oficial da ferramenta, acessado em 2014.
- [MC89] B. Marc and J. Chiasson. Application of nonlinear control methods to the positioning of a permanent magnet stepper motor. *IEEE Conference on Decision and Control*, 1(28):531–532, Dec 1989.
- [MPT95] R. Marino, S. Peresada, and P. Tomei. Nonlinear adaptive control of permanent magnet step motors. *Automatica*, 31(11):1595–1604, 1995.
- [Nis01] N. S. Nise. *Engenharia de Sistemas de Controle*. LTC, Rio de Janeiro, 3 edição, 2001.
- [Oga85] K. Ogata. *Engenharia de Controle Moderno*. Prentice Hall, 3 edição, 1985.
- [OL03] G. D. L. Ottoni and W. F. Lages. Navegação de robôs móveis em ambientes desconhecidos utilizando sonares de ultra-som. *SBA: Controle e Automação, Sociedade Brasileira de Automática.*, 14:402:411, 2003.
- [OR95] J. M. O. and S. A. Resende. *Controle Adaptativo*. Universidade do Porto, 1 edição, 1995.
- [Per14] M. N. Pereira. O que é heurística? [cantinhodomanel.blogspot.com.br/heuristica](http://cantinhodomanel.blogspot.com.br/heuristica), 2014. Acessado em Março de 2014.



- 
- [PG09] I. Poulakakis and J. W. Grizzle. The spring loaded inverted pendulum as the hybrid zero dynamics of an asymmetric hopper. *IEEE Transactions On Automatic Control*, 54, 2009.
  - [Pop61] V. M. Popov. Absolute stability of nonlinear systems of automatic control. *Automatika i Telemekhanika*, 22(8):961–979, Aug 1961.
  - [Rob11] Robocup. Site robocup. <http://www.robocup.org>, 2011. Acessado em dezembro de 2012.
  - [Sko01] S. Skogestad. *Multivariable Feedback Control - Analysis and Design*. John Wiley and Sons, New York, 2 edição, 2001.
  - [SL91] J.J.E. Slotine and W.A. LI. *Applied Nonlinear Control*. Prentice Hall, 1991.
  - [SLN14a] F. P. Silva, V. J. S. Leite, and E. G. Nepomuceno. Heurística para sintonia de ganho de compensação de incertezas de modelagem via multi-inversão robusta. *Congresso Brasileiro de Automática - CBA*, 2014.
  - [SLN14b] F. P. Silva, Valter J. S. Leite, and E. G. Nepomuceno. Heuristic to tune the compensation gain of modeling uncertainties through the robust multi inversion. *International Conference on Methods and Models in Automation and Robotics - MMAR - IEEE*, 2014.
  - [Soa11] S. J. P. Soares. Controle de um robô móvel utilizando realimentação linearizante robusta. Dissertação de Mestrado, Centro Federal de Educação Tecnológica de Minas Gerais, 2011.
  - [Suc04] I. R. Sucupira. Métodos heurísticos genéricos: Meta-heurísticas e hiper-heurísticas. Dissertação de Mestrado, Universidade de São Paulo, 2004.
  - [WB13] A. Weitzenfeld and J. Biswas. Robocup small-size league: Past, present and future. *University of South Florida*, 1, 2013.
  - [Wik13] Wikipédia. Teória de controle. <http://pt.wikipedia.org/wiki/Teoriadecontrole>, 2013. Acessado em novembro de 2013.
  - [Wik14a] Wikipédia. Heurística. <http://pt.wikipedia.org/wiki/Heuristica>, 2014. Acessado em Março de 2014.
  - [Wik14b] Wikipédia. Álgebra de lie. <http://pt.wikipedia.org/wiki/ÁlgebradeLie>, 2014. Acessado em Maio de 2014.
  - [ZW98] K. Zhou and J. While. *Essentials of Robust Control*. Prentice Hall, 1 edição, 1998.